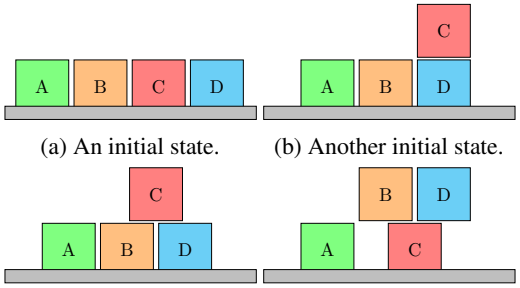
Automatically Uncovering Intended Domain Constraints in Automated Planning

Elliot Gestrin, Johannes Fichte, Jendrik Seipp

Linköping University, Sweden
firstname.lastnameliu.se

Abstract

Planning domains group tasks with shared properties and constraints. However, these constraints are typically not defined explicitly, leaving standard domain formulations incomplete. We formalize the task of automatically uncovering such constraints from positive examples alone and present an abstract algorithmic framework unifying several existing invariant learning methods. Any method fitting this framework inherits the guarantees we establish, primarily probably approximately correct convergence to the optimal estimate within the chosen candidate language, the strongest attainable guarantee in this setting. We identify three traversal strategies (exhaustive, eager and lazy) with identical guarantees but increasing efficiency. Notably, all prior methods are exhaustive or eager. To evaluate our framework, we instantiate it with two novel constraint expression sources and demonstrate efficient learning of interpretable constraints from small training sets.



(a) An initial state. (b) Another initial state.

(c) A third initial state? (d) A fourth initial state?

Figure 1: States (a)–(d) are all *syntactically* correct initial states for Blocksworld. However, only (a)–(b) are commonly accepted as legal. We use examples to automatically learn constraints that differentiate between legal and illegal states.

1 Introduction

Automated planning finds a sequence of actions that transforms an initial state into a goal state (Ghallab, Nau, and Traverso 2004). A *domain* specifies shared properties of problem sets: types, predicates and actions. Each domain imposes constraints on which problems can be encountered. For example, in Blocksworld (Slaney and Thiébaux 2001), no block may be on itself or on two others (Figure 1). Such constraints exist in all domains but are typically only given implicitly by experts carefully selecting valid input problems. Making them explicit enables generating new legal problems (e.g., Grundke, Helmert, and Röger 2025), validating generalized policies (e.g., Srivastava, Immerman, and Zilberstein 2011), improving planner performance (e.g., Fišer et al. 2021) and more. Recently, Shleyfman and Karpas (2018) and Grundke, Röger, and Helmert (2024) manually augmented domains with such explicit constraints.

Here, we go one step further and, for the first time, automatically learn such constraints over initial and goal states from positive examples. We show that many methods for learning lifted invariants in planning (e.g., Rintanen 2000; Lin 2004; Mukherji and Schubert 2005; Helmert 2009) can be adapted to our setting. Once adapted, a common structure emerges: iteratively evaluating and refining candidates from a constraint language against example states, retaining those consistent with all observations. Yet this shared skeleton has not been recognized or formalized in planning, making it hard to compare the properties of these methods.

We formalize this structure as an abstract framework for **u**ncovering **d**omain **c**onstraints from positive examples, UDC. The framework supports arbitrary first order (FO) formulas as candidates and is parameterized by a candidate set $\mathcal{C}$, an implication DAG $\mathcal{H}$ over $\mathcal{C}$ and a reduction procedure. Many existing methods, when adapted for constraint learning, are instantiations of UDC and thereby inherit the guarantees we establish: no misclassifications on the training set and *probably approximately correct* (PAC) (Haussler and Warmuth 1993) convergence to the optimal estimate within $\mathcal{C}$, the strongest possible guarantee under positive-only supervision. We further identify three DAG traversal strategies, *exhaustive*, *eager* and *lazy*, yielding identical results but increasing efficiency. No existing method is lazy, the most efficient variant.

Example 1. *Given examples like States a and b in Figure 1 and absence of c and d, we can learn, among others:*

$$\varphi_1 = \forall_{x,y} \left(on(x,\ y) \rightarrow \neg \exists_{z:\ z \neq y} on(x,\ z) \right)$$

$$\varphi_2 = \forall_{x,y} \left(on(x,\ y) \rightarrow \neg \exists_{z:\ z \neq x} on(z,\ y) \right)$$

φ_1 says that a block is on at most one other, disallowing c. φ_2 says that a block has at most one block on it, disallowing d.

2 Classical Planning

We assume that the reader is familiar with first-order logic (FO) (Priest 2001; Ebbinghaus, Flum, and Thomas 2021) and grounding. Below, we briefly summarize notation and terminology. Planning Domain Definition Language (PDDL) (McDermott et al. 1998) is the predominant specification language for deterministic planning problems and consists of domains, which specify common properties among instances, and problems, which give concrete instances.

A *domain* is a tuple $\mathcal{D} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ s.t. $\mathcal{T}$ is a rooted directed acyclic graph (DAG) $\mathcal{T} = \langle T, E \rangle$, where T is a set of *types* and $E \subseteq T \times T$ represents an order between the types such that types can be used in place of their parents. An arborescence (directed rooted tree) is most common. Let t_1 and $t_2 \in T$, then we say that t_1 is a subtype of t_2, in symbols $t_1 \leq^t t_2$, if there is a path from t_2 to t_1. $\mathcal{P}$ is a finite set of *typed predicate symbols*. A *typed predicate symbol* $p = \langle n, t \rangle$ consists of a unique symbol name n, a number $m \in \mathbb{N}_0$ of parameters (arity), and specified types t on the parameters with $t = \langle t_1, \ldots, t_m \rangle$ and $t_i \in T$. By *signature* $\Sigma_{\mathcal{P}}$, we mean the symbol names of $\mathcal{P}$, i.e., $\Sigma_{\mathcal{P}} \coloneqq \{\, n \mid \langle n, \cdot \rangle \in \mathcal{P} \,\}$. $\mathcal{A}$ is a finite set of *lifted actions* inducing state transitions. We omit further details as they are not relevant for our purposes.

Each problem belongs to a domain and is defined as follows. A *problem* is a tuple $\mathcal{R} = \langle \mathcal{D}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$ s.t. $\mathcal{D} = \langle \mathcal{T}, \mathcal{P}, \cdot \rangle$ is a *domain* with $\mathcal{T} = \langle T, E \rangle$. $\mathcal{O}$ is a finite set of *objects*. An object $o = \langle n, t \rangle$ consists of a unique symbol name n and a type $t \in T$. By abuse of notation we write $\Sigma_{\mathcal{O}}$ for all object names. We define an auxiliary function $\mathrm{type}(o)$, which returns the type of o. $\mathcal{I}$ is the *initial state*, those *atoms* (ground predicates) that hold initially. $\mathcal{G}$ is the *goal*, a typed sentence over $\Sigma_{\mathcal{O} \cup \mathcal{P}}$. By *typed formula*, we mean a formula where predicate symbols take only objects or variables according to the types. A *sentence* is a formula without free variables. The intent is to find an action sequence from $\mathcal{I}$ to a state where $\mathcal{G}$ holds. We omit the type if there is only one.

3 Legal Planning Tasks

In the classical PDDL domain definition, we cannot encode limitations on the problems themselves, though newer PDDL versions (PDDL 3) allow constraints on plan trajectories. However, in practice, only a fraction of the possible problems are intended by the domain designer. Previous works (Shleyfman and Karpas 2018; Grundke, Röger, and Helmert 2024) define constructs to formalize such constraints. Similarly, we introduce FO sentences over states to describe these restrictions. Since this in general might require derived predicates which are not directly included in the domain, we also define these. We focus on initial states before generalizing to goals.

Definition 1 (Derived Predicates). *$\varphi[x_1, \ldots, x_k]$ is a* formula *over symbols Σ with free variables x_1, …, x_k. Let $\mathcal{P}_d \coloneqq \{\, P_i^k \mid ar(P_i^k) = k; k, i \in \mathbb{N} \,\}$ be the set of all* derived predicates*. Each $P_i^k \in \mathcal{P}_d$ is defined by a* derived formula *$\varphi_i^k[x_1, \ldots, x_k]$ over $\Sigma \cup \mathcal{P}_d$ and P_i^k holds in a state iff φ_i^k evaluates to true in that state.*

Example 2 (Derived Predicate). *Consider the Blocksworld domain from Example 1. We might be interested in knowing whether a block x is anywhere above a block y, but no such predicate is provided. We then define the* `above` *predicate:* $\mathrm{above}(x, y) \coloneqq \mathrm{on}(x, y) \vee (\exists_z \mathrm{above}(x, z) \wedge \mathrm{above}(z, y))$.

An *invariant* is a derived predicate that holds in all states reachable from the initial state of a task. Finding these is a well-studied problem in planning. In contrast, we define constraints over arbitrary sets of states.

Definition 2 (State Constraints). *A* state constraint *$\mathcal{C}$ is a set of derived predicates which holds for a set of states.*

We will use state constraints over initial states to define legal problems. For ease of analysis, we present state constraints as sets of formulas, representing a conjunction.

Example 3 (State Constraints). *Consider the Blocksworld domain from Example 1. We design a state constraint $\mathcal{C}$ using the derived predicate* `above` *from Example 2:*

$$\mathcal{C} = \Big\{ \forall_x \neg\, \mathrm{above}(x, x),\ \forall_{x,y} \big(\mathrm{on}(x, y) \rightarrow \neg \exists_{z:\, z \neq y} \mathrm{on}(x, z) \big),\ \forall_x \big(\text{on-table}(x) \vee \exists_y \mathrm{on}(x, y) \big) \Big\}.$$

$\mathcal{C}$ disallows circular towers, guarantees that each block is on at most one other block and enforces that each block is somewhere. Fully describing Blocksworld initial states requires a larger constraint that also guarantees an empty hand, that each block has at most one other block on it and that the `clear` *predicate is consistent with* `on`*.*

Definition 3 (Legal Task). *Let $\mathcal{D} = \langle \cdot, \mathcal{P}, \cdot \rangle$ be a domain, $\mathcal{C}_{\mathcal{D}} \subseteq \mathbb{F}_{\mathcal{P}}$ a state constraint, and $\mathcal{R} = \langle \mathcal{D}, \cdot, \mathcal{I}, \cdot \rangle$ a problem. We say $\mathcal{R}$ is* legal *for $\mathcal{C}_{\mathcal{D}}$ if $\mathcal{I} \models \bigwedge_{c \in \mathcal{C}_{\mathcal{D}}} c$.*

In line with Shleyfman and Karpas (2018) and Grundke, Röger, and Helmert (2024), we assume the following:

Assumption 1 (Intended Constraints). *For domain $\mathcal{D}$, there is a state constraint $\mathcal{C}_{\mathcal{D}}$ intended by the modeler for deciding what problems are legal, called the* intended constraint*.*

When we state that a problem is legal, we mean that it is legal for the intended constraint of its associated domain.

Intended constraints have unique properties compared to invariants. For instance, a generalized policy may rely on structural properties of initial states that are lost in later states, for example they might always lift a block first (which requires an empty hand), meaning that invariants may be insufficient to prove their correctness. Similarly, generating or validating new problem instances requires knowledge of what legal initial states look like, which invariants over reachable states do not capture. If needed, state constraints can be extended to invariants via action analysis, as discussed later.

We are now in position to specify what we mean by automatically uncovering intended constraints.

Problem 1 (Uncover Intended Constraints). *Given a domain $\mathcal{D}$ and a set $E = \{\mathcal{R}_1, \ldots, \mathcal{R}_k\}$ of legal problems from $\mathcal{D}$, find a "good approximation" of the intended constraint.*

The notion of a "good approximation" depends on the intended use of the learned constraint. In Section 4, we present a broadly applicable definition and show that our framework converges to the optimal approximation thereof.

Algorithm 1 Algorithm UDC($\mathcal{C}$,E,$\mathcal{H}$, mode)

Data: State constraint $\mathcal{C}$, set E of problems, DAG $\mathcal{H}$ over state constraint, mode $\in \{\text{eager}, \text{lazy}\}$
Result: State constraint $\hat{\mathcal{C}} \subseteq \mathcal{C}$
$\hat{\mathcal{C}} \leftarrow \text{roots}(\mathcal{H})$
for $\langle \cdot, \mathcal{I}, \cdot \rangle \in E$ **do**
  for $c \in \hat{\mathcal{C}}$ **do**
    if $\neg(\mathcal{I} \models c)$ **then**
      if *mode* $=$ *eager* **then**
        detach c's children from all parents
      remove c from $\mathcal{H}$
      $\hat{\mathcal{C}} \leftarrow \text{roots}(\mathcal{H})$
$\hat{\mathcal{C}} \leftarrow \text{reduce}(\hat{\mathcal{C}})$
return $\hat{\mathcal{C}}$

3.1 Dealing with Goals

While the goals in PDDL tasks can be arbitrary FO sentences, they are often conjunctions of atoms. In this case, we can compile them into the initial state without loss of information and re-use the same learning framework for both goals and initial states. We do this by introducing a new goal variant p_g of each predicate p. Then, for each problem, we add $p_g(\bar{o})$ to the initial state for each goal atom $p(\bar{o})$ in the conjunction $\mathcal{G}$. Similar approaches are used by Shleyfman and Karpas (2018) and Grundke, Röger, and Helmert (2024). Formally:

Definition 4. *A* goal-compiled *domain $\mathcal{D}'$ and problem $\mathcal{R}'$ are constructed from a domain $\mathcal{D}$ and problem $\mathcal{R}$ via:*

$$\mathcal{T}' \coloneqq \mathcal{T}, \quad \mathcal{A}' \coloneqq \mathcal{A}, \quad \mathcal{O}' \coloneqq \mathcal{O}, \quad \mathcal{G}' \coloneqq \mathcal{G}$$
$$\mathcal{P}' \coloneqq \mathcal{P} \cup \{p_g \mid p \in \mathcal{P}\},$$
$$\mathcal{I}' \coloneqq \mathcal{I} \cup \{p_g(\bar{o}) \mid \exists_{\varphi \in \mathbb{F}_{\mathcal{P}}} \text{ s.t. } p(\bar{o}) \wedge \varphi \equiv \mathcal{G}\}.$$

For a non-conjunctive goal, this compilation remains sound but becomes incomplete. Every compiled atom $p_g(\bar{o})$ still encodes a necessary condition of the goal, but further such conditions (for example disjunctions $p(\bar{o}) \vee p(\bar{o}')$) may not be captured. Below, we assume that all tasks are goal-compiled.

4 The UDC Framework

Verifying whether an arbitrary formula is an invariant is PSPACE-hard, since it can be used to determine unsolvability (Rintanen 2000). In contrast, verifying whether a state constraint holds for a set of example states is polynomial, one simply evaluates the formula on each state. However, the space of possible constraints is infinite and so we restrict the search to a candidate constraint set (or language) which defines the formulas to be considered. Such restrictions are common, for example, in inductive logic programming (Cropper and Dumančić 2022) and constraint acquisition (O'Sullivan 2010). All existing methods discussed in Section 5.1 similarly fix a candidate language. This language can be provided as a fully enumerated set or defined implicitly through an update function that generates new candidates during search by *weakening* a failed candidate into implied variants.

The UDC framework (Algorithm 1) uses such candidate sets to formalize the shared structure underlying several existing invariant learning methods. Given a set of candidate constraints $\mathcal{C}$, a set of example problems E, an implication DAG $\mathcal{H}$ over $\mathcal{C}$ and a choice of traversal mode, UDC evaluates candidates against the example initial states and retains only those consistent with all observations. Since each evaluation checks a formula against a finite set of states, the procedure is polynomial in $|\mathcal{C}|$ and $|E|$. Notably, UDC is entirely independent of the action formalism: it operates only on states, making it applicable regardless of how transitions are specified. In Section 5.1, we show how existing methods can be cast as instantiations of this framework, thereby inheriting the powerful guarantees established below.

The implication DAG $\mathcal{H}$ over $\mathcal{C}$ controls the search behavior: an edge from c to c' indicates that c implies c'. As such, when c holds for an example, its descendants can be skipped as they are guaranteed to hold. When c fails, its children may be promoted to the active set $\hat{\mathcal{C}}$, depending on the traversal mode as discussed below. At the end, a *sound* reduction step prunes the candidates, improving compactness and interpretability without affecting correctness. A reduction is sound if it only removes candidates actually implied by the remaining ones, and any such reduction is acceptable. For example, candidates c and c' together might imply a third candidate c'' which cannot be captured in $\mathcal{H}$, since $\mathcal{H}$ implications must be between individual constraint pairs. The reduction could then remove c'' for a more compact result.

The framework supports two primary traversal modes, *eager* and *lazy*, which differ in how children of a failed candidate are handled. In eager mode, when a candidate c fails, its children are detached from all parents in $\mathcal{H}$ before c is removed, immediately making them roots and actively considered candidates. Eager mode has the advantage of supporting a fully implicit candidate representation via a weakening function, without requiring prior enumeration. In lazy mode, c is simply removed, and a child becomes active only once all its parents have been removed. A third mode, *exhaustive*, is achievable by specifying a DAG without edges. In essence, exhaustive mode always considers all candidates, eager considers candidates as soon as *one* parent fails and lazy considers candidates only when *all* parents fail. As we show below, all modes produce identical outputs, but at different degrees of efficiency. Existing methods either operate in exhaustive or eager mode, the two less efficient ones.

Next, we present a series of properties of UDC, proven in the appendix. First, we show that while the selection of $\mathcal{H}$ and traversal mode can significantly affect the efficiency of UDC, they do not affect the final result.

Proposition 1. *Let $\mathcal{H}$ and $\mathcal{H}'$ be two DAGs over $\mathcal{C}_{cand}$ where children are implied by parents and let m, m' be traversal modes. Then, for any set of problems E and candidates $\mathcal{C}_{cand}$, $\text{UDC}(\mathcal{C}_{cand}, E, \mathcal{H}, m)$ and $\text{UDC}(\mathcal{C}_{cand}, E, \mathcal{H}', m')$ yield equivalent results.*

While all modes yield identical results, they can differ massively in runtime. Let $\text{eval}(\mathcal{C}_{\text{cand}}, E, \mathcal{H}, m)$ denote the number of candidate evaluations, $(\mathcal{I} \models c)$, by $\text{UDC}(\mathcal{C}_{\text{cand}}, E, \mathcal{H}, m)$.

Proposition 2. *For any candidate set $\mathcal{C}_{cand}$, examples E and implication DAG $\mathcal{H}$, we have*

$$1 \leq \frac{\mathit{eval}(\mathcal{C}_{cand}, E, \mathcal{H}, \mathit{eager})}{\mathit{eval}(\mathcal{C}_{cand}, E, \mathcal{H}, \mathit{lazy})} < |\mathcal{C}_{cand}| - 1,$$

where the lower bound is attainable and $|\mathcal{C}_{cand}| - 1$ is a limit that can be approached arbitrarily closely.

Remark 1. *Exhaustive mode is never more efficient than either eager or lazy, as it always evaluates all candidates.*

Remark 2. *Since $\mathcal{H}$ and traversal mode only affect efficiency (Proposition 2), not results (Proposition 1), we use $\mathrm{UDC}(\mathcal{C}_{cand}, E)$ as notation in further analysis for simplicity.*

Next, if given sufficient examples $\mathrm{UDC}(\mathcal{C}_{\text{cand}}, E)$ will be a subset $\mathcal{C}^{\text{best}}_{\text{cand}} \subseteq \mathcal{C}_{\text{cand}}$ estimating $\mathcal{C}_{\mathcal{D}}$ best, meaning that $\mathcal{C}^{\text{best}}_{\text{cand}}$ is a subset most likely to classify illegal problems as illegal, while identifying all legal problems E as legal. Formally:

Definition 5. *Let $\mathcal{C}_{cand}$ be a set of candidate constraints and $\mathcal{C}_{\mathcal{D}}$ be the intended constraint. Then, we define the* best approximation *$\mathcal{C}^{best}_{cand}$ as the subset of $\mathcal{C}_{cand}$ such that:*

(1) Every problem legal for $\mathcal{C}_{\mathcal{D}}$ is also legal for $\mathcal{C}^{best}_{cand}$.
(2) Moreover, for any other subset $\mathcal{C}' \subseteq \mathcal{C}_{cand}$ satisfying the above, every problem illegal for $\mathcal{C}'$ is also illegal for $\mathcal{C}^{best}_{cand}$, i.e., $\mathcal{C}^{best}_{cand}$ accepts no more intended illegal instances than any other $\mathcal{C}' \subseteq \mathcal{C}_{cand}$ with Property 1.

Since $\emptyset$ satisfies Property 1 and $\mathcal{C}_{\text{cand}}$ is finite, a subset satisfying Property 1 that is maximal under Property 2 is guaranteed to exist and $\mathcal{C}^{\text{best}}_{\text{cand}}$ is well-defined. More precisely, $\mathcal{C}^{\text{best}}_{\text{cand}}$ is the largest subset of $\mathcal{C}_{\text{cand}}$ consistent with all legal instances. To see this, recall that $\mathcal{C}^{\text{best}}_{\text{cand}}$'s discriminatory power is the conjunction of its members. Property 1 thus rules out precisely the candidates that reject some legal instance, and Property 2 is maximized by retaining every candidate that does not. Note that $\mathcal{C}^{\text{best}}_{\text{cand}}$ might not be unique, but all sets are equally expressive, which we abbreviate by equalities. Next, we show that $\mathrm{UDC}(\mathcal{C}_{\text{cand}}, E)$ converges to $\mathcal{C}^{\text{best}}_{\text{cand}}$ as E grows.

Lemma 1. *Let $\mathcal{C}' \subseteq \mathcal{C}$ be state constraints. Then, the set of problems legal in $\mathcal{C}$ is a subset of those legal in $\mathcal{C}'$.*

Theorem 1. *Let E consist of unique legal problems randomly sampled from $\mathcal{D}$ and let $\mathcal{C}_{cand}$ be a candidate constraint set. Then, $\lim_{|E|\to\infty} \mathrm{UDC}(\mathcal{C}_{cand}, E) = \mathcal{C}^{best}_{cand}$.*

Given sufficient examples, UDC will yield the best possible approximation of the intended constraint from the candidates. In practice, we may have $\mathcal{C}^{\text{best}}_{\text{cand}} \subseteq \mathrm{UDC}(\mathcal{C}_{\text{cand}}, E)$ but we show that UDC is *probably approximately correct* (PAC) (Haussler and Warmuth 1993) in relation to $\mathcal{C}^{\text{best}}_{\text{cand}}$. This means that the probability of $\mathrm{UDC}(\mathcal{C}_{\text{cand}}, E)$ disagreeing with $\mathcal{C}^{\text{best}}_{\text{cand}}$ for a new randomly sampled problem is small.

Definition 6. *Let $\mathbb{D}$ be a distribution over legal problems in domain $\mathcal{D}$. Let E be a set of problems independently sampled from $\mathbb{D}$. Then, an algorithm yielding a constraint $\mathcal{C}$ based on E is* probably approximately correct *(PAC) with respect to $\mathcal{C}^{best}_{cand}$ if for any error rate ϵ and confidence level δ in $(0, 1)$, there exists a sample size $|E|$ such that: with probability at least $1 - \delta$, the probability that $\mathcal{C}$ and $\mathcal{C}^{best}_{cand}$ disagree on a randomly sampled problem from $\mathbb{D}$ is at most ϵ.*

Theorem 2. *For any distribution $\mathbb{D}$ over legal problems, $\mathrm{UDC}(\mathcal{C}_{cand}, E)$ is PAC with respect to $\mathcal{C}^{best}_{cand}$.*

Note that exhaustive UDC is a version of the CNF-learning algorithm from the original PAC work of Valiant (1984) by viewing $\mathcal{C}_{\text{cand}}$ as a Boolean space and legal problems as assignments. Both this and Shleyfman and Karpas (2018) show that, under positive-only supervision, PAC guarantees with respect to $\mathcal{C}^{\text{best}}_{\text{cand}}$ are the strongest attainable in our setting.

Algorithm 2 GreedyReduce($\mathcal{C}$)

Data: State constraint $\mathcal{C}$
Result: State constraint $\mathcal{C}' \subseteq \mathcal{C}$ s.t. $\wedge_{\mathcal{C}'} \leftrightarrow \wedge_{\mathcal{C}}$
$\mathcal{C}' \leftarrow \mathcal{C}$
for $c \in \mathcal{C}'$ **do**
  if UNSAT$(\{\neg c\} \cup \mathcal{C}' \setminus \{c\})$ **then** $\mathcal{C}' \leftarrow \mathcal{C}' \setminus \{c\}$
return $\mathcal{C}'$

Additionally, any sound reduction preserves all guarantees and analysis above. We use a greedy procedure (Algorithm 2) that iteratively removes a candidate whenever it is implied by the others (i.e., its negation is impossible), which is simple to implement and supports any candidate language. A reduction is not necessary for correctness, but it substantially improves interpretability and compactness. As empirically shown, remaining candidates can be reduced by orders of magnitude, which helps even for machine-based downstream tasks.

Since pruning nothing is sound, the reduction can also be omitted. Doing so makes UDC additive: results from one candidate set can directly inform choices for another, and examples can be processed incrementally without recomputation. We use this in Section 5 to compose instantiations. Any sound reduction can safely be applied afterwards.

Proposition 3. *Let $\mathcal{C}$ and $\mathcal{C}'$ be constraints, and E and E' be sets of problems. If we omit UDC's reduction, then:*

$$\mathrm{UDC}(\mathcal{C} \cup \mathcal{C}', E) = \mathrm{UDC}(\mathcal{C}, E) \cup \mathrm{UDC}(\mathcal{C}', E)$$
$$\mathrm{UDC}(\mathcal{C}, E \cup E') = \mathrm{UDC}(\mathrm{UDC}(\mathcal{C}, E), E')$$

5 Instantiating UDC

To apply UDC, one must choose a candidate constraint set $\mathcal{C}_{\text{cand}}$, an implication DAG $\mathcal{H}$ over $\mathcal{C}_{\text{cand}}$ and a traversal mode. We first show how existing invariant learning methods can be viewed as making these choices when adapted for initial state learning, inheriting the theoretical guarantees from Section 4. We then introduce two novel candidate sets.

5.1 Existing Methods as UDC Instances

Many methods have been proposed for learning invariants in planning. Most focus on learning ground invariants specific to a single problem instance unable to generalize across problems. Since we are interested in learning constraints that hold for entire domains, we focus instead on lifted invariants. Below we describe some diverse lifted methods, showing how each maps onto the UDC framework as an instantiation.

Rintanen (2000). Rintanen's algorithm learns lifted invariants for a single task in a two-phase process. Initialized with the set of base predicates, Phase 1 iteratively weakens candidates until all remaining candidates hold in the initial state. Weakening is done by adding a new predicate, adding an equality or identifying a variable pair. Phase 2 then keeps weakening candidates which can be violated by any of the

actions. The algorithm is restricted to STRIPS (Fikes and Nilsson 1971) and invariants are predicate disjunctions with argument inequalities. Later works (e.g., Rintanen 2017) extend supported formalisms and improve efficiency without affecting the connection to our framework. Phase 1 maps directly onto UDC in eager mode with an implicit DAG induced by the weakening procedure. Phase 2, extending state constraints to invariants via action analysis, is orthogonal and could serve as post-processing by extending the UDC loop, as explored by Mukherji and Schubert (2005).

Lin (2004). Lin proposes four categories of invariants: functional dependency, mutual exclusion, type information and domain closure. To learn the invariants, they start with a set of candidates and iteratively prove which are invariants across all example problems and actions, while assuming those that have already been proven. The four invariant categories form finite explicit candidate sets with edgeless DAGs, i.e. exhaustive mode. The iterative proof structure, is not needed in UDC: if candidates hold, they will be directly verified over the examples. Their correctness results (that small "representative" problem examples suffice for verifying correctness across all sizes) present another post-processing alternative to convert learned UDC constraints into invariants.

Mukherji and Schubert (2005). Mukherji and Schubert estimate the set of invariants for a task by sampling states and checking which candidates hold across these. Like UDC, this has the advantage of being applicable to any action formalism. To find invariants, they iteratively transform poor-performing candidates into better ones by forming a disjunction with another, adding variable (or object) inequalities or quantifying variables. To guide the search, they estimate how complementary two candidates are by storing for which assignments they fail and computing vector operations on these. Since the method relies on sampling, it does not come with guarantees for invariance, but they empirically show that it performs well. At first glance, this approach is the closest to UDC since it evaluates candidates against provided states. However, their method uses vector operations on failed invariants to guide the weakening procedure which is not well represented by a fixed, or even eagerly, induced candidate set. To map this to UDC, the additivity property (Proposition 3) can be used to let their heuristic guide candidate selection: run UDC on a fixed batch, generate all children of failed candidates, score them and select the next batch, then repeat. Each iteration can use an edgeless DAG. The UDC PAC guarantee then holds with respect to the candidates actually considered, though their heuristics primarily prune candidates guaranteed to fail.

Helmert (2009). Helmert's algorithm learns so-called *monotonicity invariants* directly from the domain definition without requiring example problems. These invariants constrain the number of true atoms within certain sets to be non-increasing across transitions. In particular, if only one is true in an initial state, this means that the atoms are mutually exclusive, called *mutexes*. The algorithm iteratively refines candidates by adding predicates until they are all provably monotone, defining an implicit UDC DAG for eager mode. It restricts the candidate space to invariants where all elements either have no arguments or share at least one argument with another. Since UDC operates on states, monotonicity cannot be checked directly. However, the practical goal of deriving mutexes is a state property and can be captured by UDC.

5.2 Novel Instantiations of UDC

Beyond viewing existing methods as UDC instances, we introduce two novel candidate sets that can instantiate the framework for any domain: $\mathcal{C}_{\text{TI}}$, based on typed implications, and the LLM-generated $\mathcal{C}_{\text{LLM}}$. Each has examples in Figure 4.

Domain-Adapted Typed Implications ($\mathcal{C}_{\text{TI}}$)
$\mathcal{C}_{\text{TI}}$ is derived from domain types and predicates. It is expressive enough to capture many useful initial-state constraints while remaining enumerable in our domains. We give expanded reasoning and the full formal definition in Appendix B and summarize the key design choices here.

Candidate Language. $\mathcal{C}_{\text{TI}}$ consists of typed single-predicate implications (with $\top$ allowed as antecedent), including negated antecedents/consequents and quantifier restrictions tied to antecedent arguments. This compact language captures common patterns such as always-true atoms, mutexes, and conditional existence constraints. We also permit type-check formulas $(\text{type}(o) = t)$ and $(\text{type}(o) \leq^t t)$, which helps represent constraints over type hierarchies.

Expressivity vs. Size. Allowing arbitrary conjunctions/disjunctions would make enumeration intractable, so we keep the base language single-predicate. We include one high-impact extension: transitive closures of arity-2 predicates (e.g., `above`), which often encode planning-relevant structure at modest additional cost.

Canonicalization and DAG. After generation, we canonicalize constraints by removing tautologies/unsatisfiable formulas and keeping one representative per equivalence class. We then build an implication DAG by one-step weakenings (antecedent generalization and consequent weakening), with root selection based on maximal quantifier domains. This enables direct comparison of exhaustive, eager, and lazy traversal in Section 6. Table 1 reports resulting $\mathcal{C}_{\text{TI}}$ sizes.

LLM-Generated Candidates ($\mathcal{C}_{\text{LLM}}$)
While $\mathcal{C}_{\text{TI}}$ is systematic, it is inherently limited to single-predicate implications. Constraints requiring conjunctive bodies, counting, or domain-specific *semantic* knowledge fall outside its scope. To address this, and to demonstrate the generality of UDC as a framework, we introduce an LLM-generated candidate set $\mathcal{C}_{\text{LLM}}$. LLMs can produce arbitrary FO constraints without manual template design and while using semantic knowledge. Recently, they have shown success in generating planning heuristics (e.g., Corrêa, Pereira, and Seipp 2025) and policies (e.g., Stein et al. 2026), suggesting they can similarly produce meaningful state constraints. Our prompts consist of

1. an expertise description to prime the LLM;
2. a description of constraints and their syntax;
3. the PDDL domain;
4. the smallest and largest example problem PDDL files; and
5. an instruction to generate up to 10 derived predicates and exactly 100 constraint candidates.

					$\mathcal{C}_{\text{TI}}$			$\mathcal{C}_{\text{LLM}}$		
Domain	$\lvert T\rvert$	$\lvert\mathcal{P}\rvert$	Arities	$\lvert E\rvert$	Candidates	Valid	Reduced	Candidates	Valid	Reduced
Blocksworld	1	5	0 / 1.0 / 2	35	8681	39.1%	36	100	84%	26
Logistics	9	3	2 / 2.0 / 2	32	19618	55.5%	36	100	70%	22
Transport	6	5	2 / 2.0 / 2	30	21358	53.0%	42	100	98%	32
Floortile	3	10	1 / 1.7 / 2	20	106358	49.5%	364	100	85%	35
Grid	1	12	0 / 1.2 / 2	5	111976	49.2%	174	100	93%	43

Table 1: Results of UDC using $\mathcal{C}_{\text{TI}}$ and $\mathcal{C}_{\text{LLM}}$ on IPC domains. $|T|$, $|\mathcal{P}|$, and $|E|$ denote the number of types, predicates, and examples, respectively. Arities are given as min/avg/max. A constraint is *valid* if it holds for all examples, i.e., remains until the UDC reduction step. Valid constraints are reported as a proportion of total candidates. The *reduced* set contains those constraints remaining after reduction, which together imply all valid constraints. As the reduced set was found using a solver with a timeout, the reduced set size is an upper bound on the minimal set. Both candidate sets yield high proportions of valid constraints with compact reduced sets.

We include the full prompt in the appendix. Showing all example problems to an LLM would be infeasible due to context length limits, and so we choose the smallest and largest ones, which was effective in prior work (e.g., Corrêa, Pereira, and Seipp 2025). We then query the LLM with structured output settings to encourage valid constraints. As $\mathcal{C}_{\text{LLM}}$ is small, exhaustive UDC mode is sufficient.

While the LLM might make mistakes, the guarantees from UDC still hold with respect to the candidates it produces and any invalid candidates will be filtered out, ensuring empirical validity and PAC properties of the final output constraint. We primarily view $\mathcal{C}_{\text{LLM}}$ as an exploratory proof of concept. Fully evaluating LLMs' candidate-generation capabilities would require a paper of its own.

6 Experiments

In this section, we present the results of evaluating UDC using $\mathcal{C}_{\text{TI}}$ and $\mathcal{C}_{\text{LLM}}$ on five domains from the International Planning Competition (IPC). With $\mathcal{C}_{\text{TI}}$, we additionally evaluate all UDC evaluation modes.

Experiment Design. We consider all domains from Grundke, Röger, and Helmert (2024), together with Logistics, which features more complex typing and was studied in Shleyfman and Karpas (2018). For each domain, we use the first typed IPC variant along with all example problems from that year's satisficing track. For $\mathcal{C}_{\text{LLM}}$, we use GPT-5-2025-08-07. Further LLM details, reduction solver configurations, hardware and wall-clock timings are provided in Appendix D.

Measures and Restrictions. For each domain and candidate set, we evaluate each constraint across all examples E. This enables us to simulate 1000 (or all) random permutations of E to evaluate sample efficiency across all traversal modes. Since positive examples alone are insufficient to compute metrics such as precision or recall, we perform a qualitative analysis of the learned constraints. Developing principled protocols for generating or sourcing representative negative examples is an interesting direction for future work.

6.1 Results

Table 1 summarizes our results, Figures 2–3 show sample efficiency and Figure 4 highlights some learned constraints. All reduced constraints are available in the appendix, along with the entirety of $\mathcal{C}_{\text{LLM}}$ for each domain.

Valid Constraints. While the number of $\mathcal{C}_{\text{TI}}$ candidates varies significantly between domains, the proportion of valid constraints is consistently high (between 39.1% and 55.5%), indicating that $\mathcal{C}_{\text{TI}}$ contains constraints applicable to a wide range of domains. This is particularly notable since roughly half of the candidates are mutually exclusively negations of the other half, though we can go over 50% valid constraints if the antecedent never holds. 7.3% (Blocksworld) to 21.1% (Logistics) of the candidate constraints fall into this category, leaving 31.7% to 41.5% of candidates valid with an antecedent that was true at some point (theoretical limit 50%). Most constraints with inapplicable antecedents have exact type matches for abstract types as their antecedents. $\mathcal{C}_{\text{LLM}}$ similarly has a high rate of valid constraints (between 70% and 98%), indicating that LLMs are capable of generating relevant candidate constraints.

Reduced Constraints. Despite the large number of valid constraints, especially for $\mathcal{C}_{\text{TI}}$, the number of reduced constraints is comparatively low for both sets (between 22 and 364). This shows that UDC is able to identify compact representations of the intended constraint. Additionally, it shows that the LLM generates redundancy in $\mathcal{C}_{\text{LLM}}$, which is suitable since this lets UDC find the most general constraints.

LLM Tendencies. Across domains, GPT-5 in $\mathcal{C}_{\text{LLM}}$ favors implication-style constraints. Many are useful, but two failure modes recur: tautologies under the model's own derived predicates (e.g., $\text{B}_{\text{LLM}}10$), and over-general constraints that admit states outside benchmark intent, such as Blocksworld with an initially held block. UDC filters inconsistent candidates and retains only those empirically supported by the examples.

Sample Efficiency. As can be seen in Figure 2, UDC demonstrates high sample efficiency with both candidate sets. The largest reduction occurs after the first example, with candidates for some domains being valid for all examples or none and reduction therefore happening *only* for the first example (e.g., $\mathcal{C}_{\text{LLM}}$ Transport). However, we also see some tasks being uniquely informative, the only ones to invalidate certain candidates (e.g., $\mathcal{C}_{\text{LLM}}$ Logistics at 31 examples).

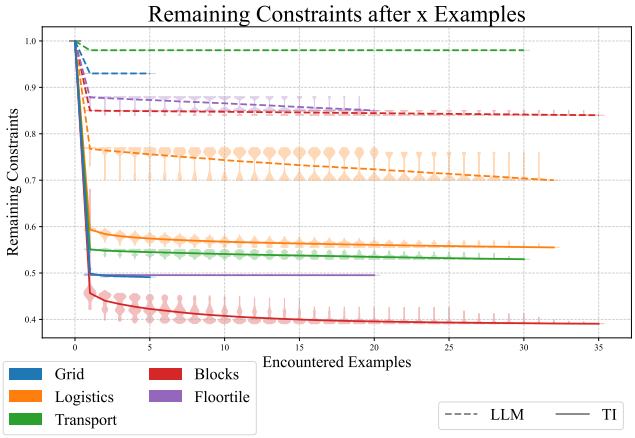


Figure 2: The proportion of candidate constraints remaining as a function of seen examples. Lines show average across 1000 permutations and areas show the distribution. Each candidate set converges quickly, though some tasks uniquely falsify certain candidates.

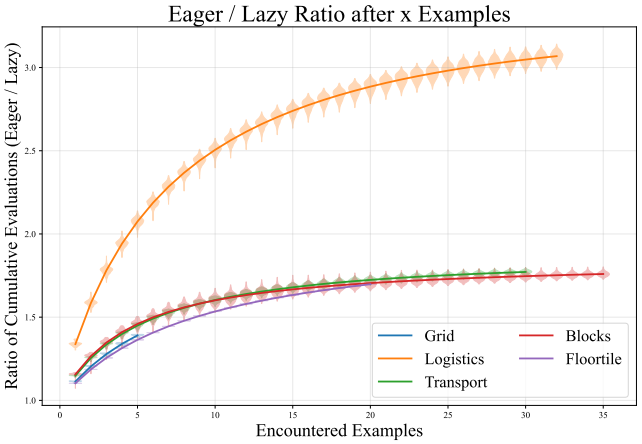


Figure 3: The cumulative constraint evaluations ratio, $\frac{(\mathcal{I} \models c) \text{ for Eager}}{(\mathcal{I} \models c) \text{ for Lazy}}$. Lines show average across 1000 permutations and areas show the distribution. For most domains, eager evaluates approximately 1.92 times more in the limit. In Logistics, this is higher at about 3.47.

Traversal Performance. As shown in Figure 3 and per Proposition 2, eager traversal evaluates more candidates than lazy traversal. Across most domains, eager converges to roughly 1.92 times the lazy evaluation count. Logistics is higher at about 3.47, likely due to its richer type structure. Exhaustive traversal is another 2.74–4.33 times slower than eager in the limit. Overall, these results reinforce lazy traversal as a major efficiency opportunity.

Blocksworld. Using $\mathcal{C}_{\text{TI}}$, we fully capture the initial-state restrictions from Grundke, Röger, and Helmert (2024), including that each block has exactly one position, when blocks are clear and that no tower is circular. Goal constraints are captured only approximately since partial states are less informative, so the single-bottom property is missed, and since $\mathcal{C}_{\text{TI}}$ excludes transitive closures for goals, meaning that acyclicity is only approximated. We also overfit some initial-state constraints. For example, $\text{B}_{\text{TI}}30$ disallows states whose only stack has height exactly two, a pattern incidentally true in all IPC examples. Under $\mathcal{C}_{\text{LLM}}$, GPT-5 instead introduces a more general representation that also permits the robot to initially hold a block or goals to have multiple stacks. Aside from these generalizations and spurious tautological constraints like $\text{B}_{\text{LLM}}10$, $\mathcal{C}_{\text{LLM}}$ likewise captures the intended domain. For $\mathcal{C}_{\text{TI}}$ we also reduce the constraints manually, finding that 22 constraints suffice to imply the 36 from our greedy reduction, showing that stronger reductions are possible.

Logistics. $\mathcal{C}_{\text{TI}}$ captures the intended domain behavior nearly perfectly. For example, $\text{L}_{\text{TI}}2$ recovers mandatory initial truck placement, while $\text{L}_{\text{TI}}27$ shows that airplanes only need a starting location when more than one exists, in contrast to trucks. This asymmetry is due to a single IPC problem containing an airplane without a starting location, likely an oversight by the original modelers. Such findings illustrate how UDC together with the $\mathcal{C}_{\text{TI}}$ structure can help debug domains and surface possible flaws when a learned constraint differs from expectation. $\mathcal{C}_{\text{LLM}}$, on the other hand, proposes the stronger assumption that every vehicle must start somewhere, which UDC then filters when the examples contradict it.

Transport. Constraint $\text{T}_{\text{TI}}11$ reveals that the *Locatable* type is never directly instantiated and functions only as an abstract supertype, while $\text{T}_{\text{TI}}12$ shows the stronger property that no *Target* subtype is ever instantiated, leaving the *Target* hierarchy entirely unused. As with Logistics, learned constraints thus expose latent domain-encoding artifacts in addition to state-level exclusions, providing a diagnostic benefit unavailable from hand-crafted constraints alone. $\mathcal{C}_{\text{LLM}}$ does not capture these abstractness properties and would permit objects of type *Locatable* to be created directly.

Floortile. Floortile is the most challenging domain for both candidate sets. The formulation of Grundke, Röger, and Helmert (2024) requires the grid to be square and the goal to be a checkered pattern, both of which involve interactions too complex for $\mathcal{C}_{\text{TI}}$ and rely on domain knowledge beyond $\mathcal{C}_{\text{LLM}}$. $\text{F}_{\text{LLM}}7$ illustrates that even GPT-5 can overfit to the provided examples: its accompanying description explicitly qualifies the constraint as holding "in the provided instances".

Grid. Although Grid is untyped in PDDL, the domain encodes type information by grounding exactly one "type predicate" per object, such as `key`. UDC recovers this latent type system for both candidate sets, identifying the types from predicate usage and inferring that every object has exactly one type, such as $\text{G}_{\text{LLM}}5$ for the `at` / `key` relationship.

7 Related Work

We discussed related work focusing on invariant learning in Section 5.1. Here, we discuss other related research.

PDDL Constraints. PDDL 3.0 (Gerevini and Long 2005) allows for constraints over plans in temporal logic, however this is largely unsupported and seldom used. Some planners

$$\forall_{a,b}\ \mathrm{on}_{tc}(a,b) \rightarrow \exists_{c:\, c\neq b}\ \exists_{d}\ \mathrm{on}_{tc}(c,d) \qquad (\mathrm{B_{TI}}30)$$

$$\forall_{x}(\ \forall_{y}(\ (\mathrm{above}(x,y) \rightarrow \mathrm{supported}(x)))) \qquad (\mathrm{B_{LLM}}10)$$

$$\top \rightarrow \forall_{a:\ (\mathrm{type}(a)\leq^{t}\mathrm{Truck})}\ \exists_{b:\ (\mathrm{type}(b)\leq^{t}\mathrm{Place})}\ \mathrm{at}(a,b) \qquad (\mathrm{L_{TI}}2)$$

$$\forall_{a:\ (\mathrm{type}(a)\leq^{t}\mathrm{Airplane})}\ (\mathrm{type}(a) \leq^{t} \mathrm{Airplane}) \rightarrow \forall_{b:\ (\mathrm{type}(b)\leq^{t}\mathrm{Airplane}) \wedge\ b\neq a}\ \exists_{c:\ (\mathrm{type}(c)\leq^{t}\mathrm{Airport})}\ \mathrm{at}(b,c) \qquad (\mathrm{L_{TI}}27)$$

$$\top \rightarrow \neg \exists_{a:\ (\mathrm{type}(a)\leq^{t}\mathrm{Locatable})}\ (\mathrm{type}(a) = \mathrm{Locatable}) \qquad (\mathrm{T_{TI}}11)$$

$$\top \rightarrow \neg \exists_{a:\ (\mathrm{type}(a)\leq^{t}\mathrm{Target})}\ (\mathrm{type}(a) \leq^{t} \mathrm{Target}) \qquad (\mathrm{T_{TI}}12)$$

$$\forall_{r:\ robot}(\ \forall_{x:\ tile}(\ \forall_{y:\ tile}((\mathrm{robot\text{-}at}(r,x) \wedge \mathrm{up}(y,x) \rightarrow \mathrm{clear}(y))))) \qquad (\mathrm{F_{LLM}}7)$$

$$\forall_{k,p}\ (\mathrm{at}(k,p) \rightarrow \mathrm{key}(k)) \qquad (\mathrm{G_{LLM}}5)$$

Figure 4: The selection of learned constraints discussed in the text. Subscript indicates the candidate set of origin and letter indicates the domain (B: Blocksworld, L: Logistics, T: Transport, F: Floortile, G: Grid). All learned constraints are available in the appendix.

also support user-provided or learned expressions to speed up search (e.g., Bacchus and Kabanza 2000; Kvarnström and Doherty 2000; Lequen 2024). However, none of these approaches can restrict initial and goal states, and as such they are orthogonal to our work. Shleyfman and Karpas (2018) suggest extending PDDL 3.0 constraints to allow such restrictions and evaluate how well mutexes from Helmert (2009) hold across initial states, finding that they usually either hold across all or none. Grundke, Röger, and Helmert (2024) instead introduce a formal framework using PDDL axioms, which are more expressive and broadly supported. They show that stratified Datalog$^{\neg}$ with a successor relation can fully capture the semantics of any PDDL domain in which the underlying constraints can be verified in polynomial time. Our work follows this approach, but focuses on learning constraints from examples rather than hand-crafting them.

LLMs for Constraints. To our knowledge, LLMs have not been used for constraint learning in planning. Nevertheless, they have shown promise in generating constraints for software validation. Pirzada et al. (2024) use LLMs to generate candidate loop invariants, which are then validated by a logic solver and used to replace the original loop with assertions. This uses the fact that the entire state space of the loop is available to guarantee correctness, which is not the case in our setting. Sun et al. (2025) instead use LLMs to generate class invariants, which are validated by LLM-written test cases. In our setting, a test case is a legal problem instance, and LLM-generated ones cannot be verified without knowing the intended constraints we aim to learn. Our results exemplify the associated risks, for example GPT-5 incorrectly accepts Blocksworld states where the robot starts holding a block. This illustrates both the difficulties and the value of our filtering method with PAC guarantees when using LLMs.

Learning Logic. Many other fields similarly learn logical expressions from examples. For example, inductive logic programming (ILP) (Cropper and Dumančić 2022) learns potentially complex expressions which might include auxiliary predicates. However, most ILP methods would result in expressions that only classify a single initial state per task size as legal, making them inapplicable to our setting. Concept learning (Lehmann and Hitzler 2010) aims to learn logical expressions that classify objects as belonging to a concept or not. This is similar to our setting, with the key difference that we learn constraints that classify entire states as legal or not, rather than individual objects. While most concept learning, ILP and similar approaches require both positive and negative examples, the field of one-class classification (OCC) (Perera, Oza, and Patel 2021) learns without negative examples, and our UDC falls into this.

8 Applications of Intended Constraints

Intended constraints have several direct downstream uses in planning pipelines.

Task generation. They can enable sampling of legal tasks and generation of training data, as shown by Grundke, Helmert, and Röger (2025).

Invariant and mutex synthesis. They can be expanded into domain-wide invariants or mutexes, as described above. Additionally, they can be used in existing methods such as Fišer (2020). Said work learns a form of lifted mutexes and includes a check that the initial weight is 1 for the current task. In principle, it is simple to generalize this into using intended constraints instead and, as such, generate mutexes that hold for all legal tasks.

Policy validation. They can lift policy evaluation beyond the usual fixed benchmark checks to instead validate across the full space of legal tasks via e.g. searching for falsifying (but legal) counter examples.

9 Conclusions & Future Work

We tackled the challenge of automatically making automated planning domain formulations more complete. To this end, we formalized the problem of learning intended constraints from positive examples and introduced UDC as a unifying framework, showing that suitable adaptations of existing invariant learning methods are special cases. Furthermore, our identification of lazy traversal as a strictly more efficient, yet previously unexploited, strategy opens a concrete avenue for future work on memory and evaluation-efficient instantiations. Our two new instantiations, $\mathcal{C}_{\mathrm{TI}}$ and $\mathcal{C}_{\mathrm{LLM}}$, demonstrate that the framework is both systematic and flexible: $\mathcal{C}_{\mathrm{TI}}$ provides principled, domain-adaptive coverage, while $\mathcal{C}_{\mathrm{LLM}}$ demonstrates the viability of LLM-generated candidates within a formally grounded learning pipeline. The resulting constraints are compact, interpretable, and directly applicable to downstream tasks such as problem generation, invariant synthesis and policy validation.

In the future, we will quantitatively evaluate the learned constraints, transfer existing works and show how to interpolate between eager and lazy evaluation modes.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

References

Bacchus, F. 2001. The AIPS'00 Planning Competition. *AI Magazine*, 22(3): 47–56.

Bacchus, F.; and Kabanza, F. 2000. Using Temporal Logics to Express Search Control Knowledge for Planning. *AIJ*, 116(1–2): 123–191.

Coles, A.; Coles, A.; García Olaya, A.; Jiménez, S.; Linares López, C.; Sanner, S.; and Yoon, S. 2012. A Survey of the Seventh International Planning Competition. *AI Magazine*, 33(1): 83–88.

Corrêa, A. B.; Pereira, A. G.; and Seipp, J. 2025. Classical Planning with LLM-Generated Heuristics: Challenging the State of the Art with Python Code. arXiv:2503.18809.

Cropper, A.; and Dumančić, S. 2022. Inductive logic programming at 30: a new introduction. *JAIR*, 74: 765–850.

De Moura, L.; and Bjørner, N. 2008. Z3: An efficient SMT solver. In *Proc. TACAS 2008*, 337–340. Springer.

Ebbinghaus, H.-D.; Flum, J.; and Thomas, W. 2021. *Mathematical Logic*. Springer.

Fikes, R. E.; and Nilsson, N. J. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *AIJ*, 2: 189–208.

Fišer, D. 2020. Lifted Fact-Alternating Mutex Groups and Pruned Grounding of Classical Planning Problems. In *Proc. AAAI 2020*, 9835–9842.

Fišer, D.; Gnad, D.; Katz, M.; and Hoffmann, J. 2021. Custom-Design of FDR Encodings: The Case of Red-Black Planning. In *Proc. IJCAI 2021*, 4054–4061.

Gerevini, A. E.; and Long, D. 2005. Plan Constraints and Preferences in PDDL3. Technical Report R. T. 2005-08-47, University of Brescia, Department of Electronics for Automation.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning: Theory and Practice*. Morgan Kaufmann.

Grundke, C.; Helmert, M.; and Röger, G. 2025. Domain-Independent Instance Generation for Classical Planning. In *Proc. CP 2025*, 805–809.

Grundke, C.; Röger, G.; and Helmert, M. 2024. Formal Representations of Classical Planning Domains. In *Proc. ICAPS 2024*, 239–248.

Haussler, D.; and Warmuth, M. 1993. *The Probably Approximately Correct (PAC) and Other Learning Models*, 291–312. Boston, MA: Springer US. ISBN 978-0-585-27366-2.

Helmert, M. 2009. Concise Finite-Domain Representations for PDDL Planning Tasks. *AIJ*, 173: 503–535.

Kvarnström, J.; and Doherty, P. 2000. TALplanner: A temporal logic based forward chaining planner. *Annals of Mathematics and Artificial Intelligence*, 30: 119–169.

Lehmann, J.; and Hitzler, P. 2010. Concept learning in description logics using refinement operators. *Machine Learning*, 78(1): 203–250.

Lequen, A. 2024. Learning Interpretable Classifiers for PDDL Planning. In *Proc. ECAI 2024*, 4140–4147.

Lin, F. 2004. Discovering State Invariants. In *Proc. KR 2004*, 536–544.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL – The Planning Domain Definition Language – Version 1.2. Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, Yale University.

McDermott, D. M. 2000. The 1998 AI Planning Systems Competition. *AI Magazine*, 21(2): 35.

Mukherji, P.; and Schubert, L. K. 2005. Discovering Planning Invariants as Anomalies in State Descriptions. In *ICAPS*, 223–230.

O'Sullivan, B. 2010. Automated Modelling and Solving in Constraint Programming. In *Proc. AAAI 2010*, volume 24, 1493–1497.

Perera, P.; Oza, P.; and Patel, V. M. 2021. One-Class Classification: A Survey. arXiv:2101.03064.

Pirzada, M. A. A.; Reger, G.; Bhayat, A.; and Cordeiro, L. C. 2024. LLM-Generated Invariants for Bounded Model Checking Without Loop Unrolling. In *Proc. ASE 2024*, 1395–1407.

Priest, G. 2001. *Logic: A Very Short Introduction*. Oxford University Press. ISBN 0192893203.

Rintanen, J. 2000. An Iterative Algorithm for Synthesizing Invariants. In *Proc. AAAI 2000*, 806–811.

Rintanen, J. 2017. Schematic Invariants by Reduction to Ground Invariants. In *Proc. AAAI 2017*, 3644–3650.

Shleyfman, A.; and Karpas, E. 2018. Position Paper: Reasoning About Domains with PDDL. In *Technical Report SS-18*, The 2018 AAAI Spring Symposium Series, 582–587. ISBN 978-1-57735-795-7.

Slaney, J.; and Thiébaux, S. 2001. Blocks World revisited. *AIJ*, 125(1–2): 119–153.

Srivastava, S.; Immerman, N.; and Zilberstein, S. 2011. A new representation and associated algorithms for generalized planning. *AIJ*, 175(2): 393–401.

Stein, K.; Hodel, N.; Fišer, D.; Hoffmann, J.; Katz, M.; and Koller, A. 2026. Improved Generalized Planning with LLMs through Strategy Refinement and Reflection. In *Proceedings of the 36th International Conference on Automated Planning and Scheduling (ICAPS 2026)*.

Sun, C.; Agashe, V.; Chakraborty, S.; Taneja, J.; Barrett, C.; Dill, D.; Qiu, X.; and Lahiri, S. K. 2025. ClassInvGen: Class invariant synthesis using large language models. In *International Symposium on AI Verification*, 64–96. Springer.

Valiant, L. G. 1984. A theory of the learnable. *Commun. ACM*, 27(11): 1134–1142.

Automatically Uncovering Intended Domain Constraints in Automated Planning

Appendix

A Proofs

Proposition 1.
Let $\mathcal{H}$ and $\mathcal{H}'$ be two DAGs over $\mathcal{C}_{\text{cand}}$ where children are implied by parents and let m, m' be traversal modes. Then, for any set of problems E and candidates $\mathcal{C}_{\text{cand}}$, $\text{UDC}(\mathcal{C}_{\text{cand}}, E, \mathcal{H}, m)$ and $\text{UDC}(\mathcal{C}_{\text{cand}}, E, \mathcal{H}', m')$ yield equivalent results.

Proof. Let $\tilde{\mathcal{C}}$ and $\tilde{\mathcal{C}}'$ be the sets before reduction when using $\mathcal{H}$ with m and $\mathcal{H}'$ with m' respectively. For every $c \in \tilde{\mathcal{C}} \setminus \tilde{\mathcal{C}}'$, some $c' \in \tilde{\mathcal{C}}'$ must imply c. This follows since c held on all examples (by its inclusion in $\tilde{\mathcal{C}}$) and as such could never have left $\tilde{\mathcal{C}}'$. Thus, an ancestor c' of c was considered under $\mathcal{H}'$ with m' instead and remained in $\tilde{\mathcal{C}}'$. However, conversely, since c' held, either it or an $\mathcal{H}$-ancestor thereof must appear in $\tilde{\mathcal{C}}$. This can be traced upward until finding a $\tilde{c}$ appearing in both sets that implies both c and c', or is one of them, such that $\tilde{c}$ is implied by no other candidate in either set. Regardless of sound reduction used, $\tilde{c}$ (or a candidate subset implying $\tilde{c}$) will be kept in both sets. Irrespective of which of these options remains in each set, it will imply both c and c' and as such neither will contribute to the discriminative power of their set and any potential reduction-dependent inclusion or exclusion thereof is as such inconsequential. Repeating this for all candidates in $\tilde{\mathcal{C}} \setminus \tilde{\mathcal{C}}'$ and $\tilde{\mathcal{C}}' \setminus \tilde{\mathcal{C}}$ yields equivalent resulting sets. □

Proposition 2.
For any candidate set $\mathcal{C}_{\text{cand}}$, examples E and implication DAG $\mathcal{H}$:

$$1 \;\leq\; \frac{\text{eval}_{\text{eager}}(\mathcal{C}_{\text{cand}}, E, \mathcal{H})}{\text{eval}_{\text{lazy}}(\mathcal{C}_{\text{cand}}, E, \mathcal{H})} \;<\; |\mathcal{C}_{\text{cand}}| - 1$$

where the lower bound is attainable and $|\mathcal{C}_{\text{cand}}| - 1$ is a limit that can be approached arbitrarily closely.

Proof. We prove the lower and upper bounds separately.

Lower bound. In lazy mode, a candidate c is evaluated if and only if it is a root of $\mathcal{H}$ initially or all its parents have been removed. In eager mode, a candidate c is instead evaluated if and only if it is a root initially or at least *one* parent has been removed. As such, if c is a root, then it is evaluated in both modes. If c is not a root and it is evaluated in lazy mode, then all its parents have been removed, and as such at least one parent has been removed, so c is also evaluated in eager mode. Therefore every candidate evaluated in lazy mode is also evaluated in eager mode, giving $\text{eval}_{\text{eager}}(\mathcal{C}_{\text{cand}}, E, \mathcal{H}) \geq \text{eval}_{\text{lazy}}(\mathcal{C}_{\text{cand}}, E, \mathcal{H})$ and a ratio of at least 1. Equality holds, e.g., for an edgeless DAG.

Upper bound. Let α and $\beta_1, \ldots, \beta_N$ be formulas such that α holds on all initial states and $\beta_1, \ldots, \beta_N$ hold on none. Let $\beta = \beta_1 \wedge \cdots \wedge \beta_N$ and $\varphi_i = \alpha \vee \beta_i$ for each i. Define $\mathcal{C}_{\text{cand}} = \{\alpha,\ \beta\} \cup \{\varphi_i : 1 \leq i \leq N\}$. Let $\mathcal{H}$ have α and β as roots, with each φ_i as a child of both.

In lazy mode: α holds, so it is never removed, while the conjunction β fails on the first example and is removed immediately. However, each φ_i remains unconsidered since a parent holds. For each subsequent example, only α is evaluated. Total: $1 + |E|$ evaluations.

In eager mode: As in lazy mode, α holds and β is removed on the first example. However, since a parent is removed, each φ_i is added to the considered set and are also evaluated on the first example, holding since their α component holds. From then on, every example evaluates α and all N children. Total: $1 + |E|(N + 1)$ evaluations.

The ratio is $\frac{1+|E|(N+1)}{1+|E|}$, which approaches $N+1 = |\mathcal{C}_{\text{cand}}|-1$ as $|E| \to \infty$. This is the supremum: any difference between eager and lazy mode requires at least one candidate to fail, and the above construction maximizes the resulting difference in evaluated candidates after that first failure. □

Lemma 1.
Let $\mathcal{C}' \subseteq \mathcal{C}$ be state constraints. Then, the set of problems legal in $\mathcal{C}$ is a subset of those legal in $\mathcal{C}'$.

Proof. Since $\mathcal{C}' \subseteq \mathcal{C}$, we have that $\bigwedge_{C \in \mathcal{C}} C \rightarrow \bigwedge_{C' \in \mathcal{C}'} C'$. Therefore, any initial state satisfying all constraints in $\mathcal{C}$ must also satisfy all constraints in $\mathcal{C}'$. □

Theorem 1.
Let E consist of unique legal problems randomly sampled from $\mathcal{C}_{\mathcal{D}}$ and let $\mathcal{C}_{\text{cand}}$ be any candidate constraint. Then, $\lim_{|E| \to \infty} \text{UDC}(\mathcal{C}_{\text{cand}}, E) = \mathcal{C}_{\text{cand}}^{\text{best}}$.

Proof. By construction, $\text{UDC}(\mathcal{C}_{\text{cand}}, E)$ has no false negatives. By Lemma 1, removing any $c \in \mathcal{C}_{\text{cand}}$ either preserves or decreases illegal problems identified as such. Since we removed the minimum set of candidates to eliminate false negatives, no other subset identifies more illegal problems without introducing false negatives. □

Theorem 2.
For any distribution $\mathbb{D}$ over legal problems, $\text{UDC}(\mathcal{C}_{\text{cand}}, E)$ is PAC with respect to $\mathcal{C}_{\text{cand}}^{\text{best}}$.

Proof. Let E_i denote the first i examples and let $\mathbb{P}_i$ be the probability under $\mathbb{D}$ that $\text{UDC}(\mathcal{C}_{\text{cand}}, E_i)$ and $\mathcal{C}_{\text{cand}}^{\text{best}}$ disagree on a randomly sampled legal problem. Since $\text{UDC}(\mathcal{C}_{\text{cand}}, E_{i+1}) \subseteq \text{UDC}(\mathcal{C}_{\text{cand}}, E_i)$ (candidates are only removed), Lemma 1 gives $\mathbb{P}_{i+1} \leq \mathbb{P}_i$, so $\mathbb{P}_i$ is non-increasing in i.

Let $B(p, c, S)$ denote the smallest number of independent Bernoulli trials with success probability at least p needed to push the probability of fewer than S successes below c; $B(p, c, S)$ is finite for all $0 < p, c < 1$ and $S \in \mathbb{N}$.

The $(i{+}1)$-th example has conditional probability exactly $\mathbb{P}_i$ (given E_i) of being a legal problem falsely rejected by $\text{UDC}(\mathcal{C}_{\text{cand}}, E_i)$. If this happens, at least one candidate is removed. Call this a *hit*.

After $|E|$ examples, one of two cases holds:

(1) $\mathbb{P}_j \leq \epsilon$ for some $j \leq |E|$. Since $\mathbb{P}_i$ is non-increasing, $\mathbb{P}_{|E|} \leq \epsilon$, satisfying the PAC condition.
(2) $\mathbb{P}_i > \epsilon$ for all $i \leq |E|$, meaning fewer than $\Delta C = |\mathcal{C}_{\text{cand}} \setminus \mathcal{C}_{\text{cand}}^{\text{best}}|$ hits occurred: each hit removes at least one candidate, so after ΔC hits $\text{UDC}(\mathcal{C}_{\text{cand}}, E_i)$ has reached $\mathcal{C}_{\text{cand}}^{\text{best}}$ and $\mathbb{P}_i = 0 < \epsilon$.

The probability of case (2) is at most the probability of fewer than ΔC successes in $|E|$ i.i.d. Bernoulli(ϵ) trials: each conditional hit probability is at least ϵ regardless of prior outcomes (since $\mathbb{P}_i > \epsilon$ in case (2)), and the comparison to i.i.d. trials holds by stochastic dominance. By definition of B, choosing $|E| = B(\epsilon, \delta, \Delta C)$ makes this probability at most δ.

Since ΔC is unknown but bounded above by $|\mathcal{C}_{\text{cand}}|$, we may instead choose $|E| = B(\epsilon, \delta, |\mathcal{C}_{\text{cand}}|) \geq B(\epsilon, \delta, \Delta C)$, which only decreases the probability of case (2). Hence with probability at least $1 - \delta$ we are in case (1) and $\mathbb{P}_{|E|} \leq \epsilon$. This proof closely follows Valiant's (1984) proof of PAC learnability of bounded CNF expressions, with each problem inducing a Boolean vector over $\mathcal{C}_{\text{cand}}$. □

Proposition 3.
Let $\mathcal{C}$ and $\mathcal{C}'$ be constraints, and E and E' be sets of problems. If we omit UDC's reduction, then:

$$\text{UDC}(\mathcal{C} \cup \mathcal{C}', E) = \text{UDC}(\mathcal{C}, E) \cup \text{UDC}(\mathcal{C}', E)$$
$$\text{UDC}(\mathcal{C}, E \cup E') = \text{UDC}(\text{UDC}(\mathcal{C}, E), E')$$

Proof.

$$\text{UDC}(\mathcal{C} \cup \mathcal{C}', E) = \left\{ c \in \mathcal{C} \cup \mathcal{C}' : \underset{\langle \cdot, \mathcal{I}, \cdot \rangle \in E}{\forall} \mathcal{I} \models c \right\}$$
$$= \left\{ c \in \mathcal{C} : \underset{\langle \cdot, \mathcal{I}, \cdot \rangle \in E}{\forall} \mathcal{I} \models c \right\} \cup \left\{ c \in \mathcal{C}' : \underset{\langle \cdot, \mathcal{I}, \cdot \rangle \in E}{\forall} \mathcal{I} \models c \right\}$$
$$= \text{UDC}(\mathcal{C}, E) \cup \text{UDC}(\mathcal{C}', E)$$
$$\text{UDC}(\mathcal{C}, E \cup E') = \left\{ c \in \mathcal{C} : \underset{\langle \cdot, \mathcal{I}, \cdot \rangle \in E \cup E'}{\forall} \mathcal{I} \models c \right\}$$
$$= \left\{ c \in \left\{ c \in \mathcal{C} : \underset{\langle \cdot, \mathcal{I}, \cdot \rangle \in E}{\forall} \mathcal{I} \models c \right\} : \underset{\langle \cdot, \mathcal{I}', \cdot \rangle \in E'}{\forall} \mathcal{I}' \models c \right\}$$
$$= \text{UDC}(\text{UDC}(\mathcal{C}, E), E')$$ □

Automatically Uncovering Intended Domain Constraints in Automated Planning
Appendix

B Informal Reasoning and Formal Definition of $\mathcal{C}_{\text{TI}}$

The main paper (Section 5.2) gave a compact overview of $\mathcal{C}_{\text{TI}}$. Here we expand the informal rationale behind each design choice and then provide the full formal definition. $\mathcal{C}_{\text{TI}}$ is derived for a domain from its types and predicates, and is designed as a middle ground between expressivity and tractability: expressive enough to capture many interesting constraints across diverse domains, while still being enumerable so that eager, lazy, and exhaustive traversal modes can be compared directly.

Informal Reasoning

Simple implications. Implications can express many useful constraints, especially if we allow $\top$ as an antecedent. They form the basis of our **T**yped **I**mplications ($\mathcal{C}_{\text{TI}}$) candidate set. In their base form, $\mathcal{C}_{\text{TI}}$ constraints are closed implications whose antecedent and consequent each consist of a single predicate under typed quantifiers. Typical patterns they capture include:

- "A certain predicate is always true." E.g. that the hand starts empty in Blocksworld: $\top \rightarrow \text{hand-empty}()$.
- "A predicate disallows another, possibly for some of the same arguments." E.g. if x is on the table, it cannot be on any y: $\forall_x \text{ontable}(x) \rightarrow \neg\exists_y \text{on}(x, y)$.
- "A predicate enforces another, possibly for some of the same arguments." E.g. if x is not clear, some other block y must be on it: $\forall_x \neg \text{clear}(x) \rightarrow \exists_{y:\ y \neq x} \text{on}(y, x)$.
- "Each object of a certain type has a certain property." E.g. no block is held: $\forall_x(\text{type}(x) = \text{block}) \rightarrow \neg \text{holding}(x)$.

Additionally, consequent quantifiers can restrict themselves to (or exclude) the antecedent arguments, which is what enables the $y \neq x$ condition in the third pattern above. We also enforce that goal-compiled predicates may appear as antecedents only if the consequent is also a goal-compiled predicate, since mixing is semantically meaningless: it would relate the initial and goal state, whereas both must hold independently.

Negations. Many interesting constraints involve negations in the antecedent or consequent, so $\mathcal{C}_{\text{TI}}$ admits negated antecedents and consequents. Consequently, about half of the resulting constraints are mutually exclusive with the other half (being their negations). This redundancy is unavoidable: any domain can be equivalently described using either the positive or negative form of each predicate, and ahead of time we cannot know which convention a given PDDL author used. For example, in Blocksworld, using `hand-full` instead of `hand-empty` results in the same semantics and constraints, only with this predicate's polarity inverted. Including both polarities is what keeps $\mathcal{C}_{\text{TI}}$ encoding-agnostic.

Types. Domains often contain "abstract" types that exist only to organize other types, without being directly instantiated. In a Logistics-style domain, for instance, there might be a Vehicle type with Truck and Airplane as subtypes. To support reasoning over such hierarchies, we allow $\mathcal{C}_{\text{TI}}$ antecedents and consequents to be of the form $(\text{type}(o) = t)$ or $(\text{type}(o) \leq^t t)$, where o is a parameter and t is any type in the domain. This lets us express constraints such as $\top \rightarrow \neg\exists_x(\text{type}(x) = \text{Vehicle})$, forbidding the direct instantiation of abstract types. More generally, type checks let us represent: (i) that some types must be instantiated, (ii) consequents that hold for all objects of a certain type, and (iii) the same with object reuse. The last case warrants elaboration. By our definition of $\mathcal{C}_{\text{TI}}$, consequent quantifiers cannot be restricted based on one another, only by antecedent arguments. A type-check antecedent introduces a single quantified variable that the consequent can then refer to repeatedly. For example, to forbid blocks from being on themselves, we write $\forall_x(\text{type}(x) = \text{block}) \rightarrow (\forall_{y:\ (y=x)} \forall_{z:\ (z=x)} \neg \text{on}(y, z))$, which is equivalent to the more readable $\forall_x(\text{type}(x) = \text{block}) \rightarrow \neg \text{on}(x, x)$. Throughout, we use the simplified form for clarity.

Interactions. Restricting $\mathcal{C}_{\text{TI}}$ to single-predicate antecedents and consequents keeps the resulting set meaningful yet small enough for enumeration. Extending to conjunctive antecedents or disjunctive consequents would more than square the number of possible antecedents or consequents, leading to combinatorial blow-up. There is, however, one common PDDL interaction we do accommodate: transitive closure. The transitive closure of a binary predicate p is $p_{tc}(x, y) := p(x, y) \vee \exists_z(p(x, z) \wedge p_{tc}(z, y))$. A natural example is the `above` predicate in Blocksworld, which is the transitive closure of `on`. Including transitive closures only doubles the number of arity-2 predicates under consideration, while capturing useful planning-relevant structure (e.g. acyclicity of stacks). We therefore allow transitive closures of arity-2 predicates to appear in both antecedents and consequents of $\mathcal{C}_{\text{TI}}$.

Canonical forms. Constraints in $\mathcal{C}_{\text{TI}}$ are not syntactically unique. For example, $\top \rightarrow \forall_x \forall_y p(x, y)$ and $\top \rightarrow \forall_y \forall_x p(x, y)$ are equivalent, and some constraints are tautological or unsatisfiable. To keep $\mathcal{C}_{\text{TI}}$ small, we choose a single representative from each equivalence class and remove tautologies and unsatisfiable constraints. The resulting set is small enough to fully enumerate, which is what enables direct comparison of eager, lazy, and exhaustive traversal modes in our experiments. We note that eager traversal with dynamically generated candidates, without pre-enumeration, is also possible.

Implication DAG. To support efficient traversal modes, we induce a weakening DAG $\mathcal{H}$ over $\mathcal{C}_{\text{TI}}$: the children of a candidate c are all constraints obtained by applying exactly one of the following transformations to c:

1. Replace $\top$ in the antecedent with any other antecedent, with its quantifiers made maximally inclusive and each existential restricted to the antecedent arguments.
2. Convert one universal quantifier in the consequent into an existential quantifier.
3. Restrict a universal consequent quantifier by replacing its type with a direct subtype.
4. Restrict a universal consequent quantifier by removing one object from its domain (via adding or removing a reference to an antecedent argument), or the inverse for an existential quantifier.

The asymmetric handling of existential quantifiers in steps 1 and 2 is needed to avoid vacuous-truth pathologies. The roots of the DAG are:

- candidates with $\top$ as antecedent and maximally inclusive consequent quantification; and
- candidates with a non-$\top$ antecedent and at least one existential consequent quantifier with exclusion, where each quantifier is maximally inclusive.

If the consequent is negated, the roles of universal and existential quantifiers above are swapped. Moving down the DAG corresponds to progressively weaker (more permissive) constraints, which matches the eager/lazy traversal interpretation used in the experiments.

Formal Definition

We now give the precise definition of $\mathcal{C}_{\text{TI}}$. Let $\mathcal{D} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a domain and let $\text{par}(P) = \langle t_1, \ldots, t_m \rangle$ denote the parameter types of predicate or type-check P.

Definition 7 (Typed Implications)**.** *Let X and Y be predicates or type-checks with typed parameters. The set of* typed implications $\text{TI}(X, Y)$ *consists of all constraints of the form:*

$$\underset{x_1\colon \text{type}(x_1) \leq^t \text{type}(\mathit{par}(X)_1)}{\forall} \cdots \underset{x_N\colon \text{type}(x_N) \leq^t \text{type}(\mathit{par}(X)_N)}{\forall} X'(x_1, \ldots, x_N) \rightarrow \underset{\tilde{y}_1\colon \text{type}(\tilde{y}_1) \leq^t t_1 \,\wedge\, (\tilde{y}_1 \circ_1 \tilde{S}_1)}{Q_1} \cdots \underset{\tilde{y}_M\colon \text{type}(\tilde{y}_M) \leq^t t_M \,\wedge\, (\tilde{y}_M \circ_M \tilde{S}_M)}{Q_M} Y'(y_1, \ldots, y_M)$$

where the following conditions hold:

$$\begin{aligned}
&X' \in \{X, \neg X\}\,,\\
&Y' \in \{Y, \neg Y\}\,,\\
&\{\tilde{y}_1, \ldots, \tilde{y}_M\} = \{y_1, \ldots, y_M\}\,,\\
&t_i = \text{type}(\mathit{par}(Y)_j) \textit{ s.t. } y_j = \tilde{y}_i \textit{ for all } i \in [M]\,,\\
&Q_i \in \{\forall, \exists\} \textit{ for all } i \in [M]\,,\\
&\circ_i \in \{\in, \notin\} \textit{ for all } i \in [M]\,,\\
&\tilde{S}_i \subseteq \{x_1, \ldots, x_N\} \textit{ for all } i \in [M]\,.
\end{aligned}$$

To interpret this definition, note:

- X' and Y' control the negation of the antecedent and consequent, respectively.
- $\tilde{y}_i$ is a reordering of the arguments of Y', ensuring that every quantification order is considered.
- t_i is the type that $\tilde{y}_i$ must belong to for it to be a valid argument for Y'.
- Q_i decides whether the quantification for $\tilde{y}_i$ is universal or existential.
- $\circ_i$ decides whether said quantification restricts to some ($\in$) or excludes some ($\notin$) of the arguments of X'.
- $\tilde{S}_i$ is any subset of arguments to X', choosing which arguments $\circ_i$ restricts over.

Note that $|\tilde{S}_i| = 0$ (with $\circ_i$ as $\notin$) incurs no restriction, while $|\tilde{S}_i| = 1$ (with $\circ_i$ as $\in$) forces the i-th argument of Y to equal the chosen argument of X, resulting in assignment-like constraints.

Next, $\mathcal{C}_{\text{TI}}$ is constructed by applying TI across relevant pairs of predicates and type-checks. Let $\mathcal{P}_g = \{p_g \mid p \in \mathcal{P}\}$ denote the goal-compiled predicates (Definition 3) and let $\mathcal{P}_{tc} = \{p_{tc} \mid p \in \mathcal{P},\ \text{arity}(p) = 2\}$ denote the transitive closures of all arity-2 predicates, defined as $p_{tc}(x, y) \coloneqq p(x, y) \vee \exists_z (p(x, z) \wedge p_{tc}(z, y))$.

Definition 8 (Typed Implication Candidate Set)**.** *For a domain $\mathcal{D} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ with type set T, let $\mathcal{P}^+ = \mathcal{P} \cup \mathcal{P}_g \cup \mathcal{P}_{tc}$ be the extended predicate set. Define the helper sets:*

$$\begin{aligned} \mathit{ant}_p &= \mathcal{P}^+ \cup \{\top\} \cup \mathit{con}_t \\ \mathit{con}_p &= \mathcal{P}^+ \\ \mathit{ant}_t &= \{\top\} \\ \mathit{con}_t &= \{\lambda o.\, (\mathrm{type}(o) = t) : t \in T\} \cup \\ &\qquad \{\lambda o.\, (\mathrm{type}(o) \leq^t t) : t \in T\} \end{aligned}$$

Then $\mathcal{C}_{TI}$ is defined as:

$$\mathcal{C}_{TI}(\mathcal{D}) = \bigcup_{\substack{x \in \mathit{ant}_p,\, y \in \mathit{con}_p: \\ (x \in \mathcal{P}_g) \to (y \in \mathcal{P}_g)}} \mathrm{TI}(x, y) \cup \bigcup_{x \in \mathit{ant}_t,\, y \in \mathit{con}_t} \mathrm{TI}(x, y)$$

where $(x \in \mathcal{P}_g) \to (y \in \mathcal{P}_g)$, ensures that goal predicates may appear as antecedent only if the consequent also is a goal predicate.

After construction, $\mathcal{C}_{TI}$ is reduced to canonical forms by choosing one representative from each equivalence class of logically equivalent constraints and removing tautologies and unsatisfiable constraints.

Automatically Uncovering Intended Domain Constraints in Automated Planning

Appendix

C LLM Prompt for $\mathcal{C}_{\text{LLM}}$

```
You are a highly-skilled professor in AI planning and the PDDL
(Planning Domain Definition Language) language. Your task is to
analyze a planning domain and generate state constraints that hold
for valid problems.

While constraints do need to hold across all valid problems in a
domain, yours will be passed through a tool to verify this. As such,
you can take risks and propose constraints that may not hold in all
cases, though you should aim for high accuracy and generality.

The generated constraints should consider both the initial state and
the goal conditions. However, they do not need to hold throughout the
execution of a plan. Also, note that the goal is only a partial
assignment of predicates and that goal predicates are only those
directly involved in a goal conjunction.

# State Constraints

State constraints are arbitrary first-order logic formulas that
restrict which problems are valid in a planning domain. Unlike
traditional planning invariants that must hold throughout execution,
these constraints only need to be satisfied by the initial problem
formulation.

Constraints can be any logical formula combining predicates,
quantifiers, and logical operators.

Constraints should be expressed such that they are true for all valid
problems, i.e. if P or Q should hold for all objects in all valid
problems, you can express this as forall x: T -> (P(x) or Q(x)).

## Syntax

- Predicates: P1(x, y), P2(x), etc.
- Logical operators:
  - and: conjunction
  - or: disjunction
  - not: negation of a formula
  - ->: implication
  - !=: inequality between VARIABLES ONLY (x != y)
- Quantifiers:
  - forall x: T - universal quantification over type T
  - exists x: T - existential quantification over type T

## Examples

1. Generic mutual exclusion (using negation):
   forall x: T1 -> not(P1(x) and P2(x))

2. Generic implication with quantifiers:
   forall x: T1 -> (P1(x) -> exists y: T2 -> P2(x, y))

3. Generic disjunction constraint:
   forall x: T1 -> (P1(x) or P2(x))

4. Generic constraint without free variables:
   (exists x: T1 -> P1(x)) -> (exists y: T2 -> P2(y))
```

5. Generic uniqueness constraint (using variable inequality):
   forall x: T1 -> forall y: T2 -> forall z: T2 ->
   ((P1(x, y) and P1(x, z)) -> y != z)

Derived Predicates

You may also define derived predicates, which are predicates computed from base predicates using logical formulas. These are useful for expressing transitive closures, reachability, or complex conditions that simplify constraint expression.

Derived Predicate Syntax

Derived predicates are defined with a name, typed arguments, and a formula:
- Name: D1, D2, etc.
- Arguments: (x: T1, y: T2, ...)
- Formula: any first-order logic formula using the same syntax as constraints

Derived Predicate Examples

1. Generic transitive closure:
   D1(x: T1, y: T1) := P1(x, y) or
   (exists z: T1 -> (P1(x, z) and D1(z, y)))

2. Generic shared property:
   D2(x: T1, y: T1) := exists z: T2 -> (P2(x, z) and P2(y, z))

3. Generic reachability:
   D3(x: T1) := exists y: T2 -> P3(x, y)

Task

Given the domain PDDL below, along with the smallest and largest example problems, generate:

1. Up to 10 derived predicates that capture useful relationships in the domain (transitive closures, shared properties, reachability, etc.). These are for your own use in writing constraints.
2. Exactly 100 state constraint candidates that are likely to hold for valid problems in this domain. Focus on constraints that are semantically meaningful and likely to hold in valid problems initially (not necessarily throughout execution).

Consider:
- Physical constraints (objects can’t be in multiple places)
- Causal relationships (certain predicates imply others)
- Mutex relationships (predicates that cannot hold simultaneously)
- Resource constraints (limited capacities, unique assignments)
- Transitive properties (connectivity, containment hierarchies)
- Existential requirements (every object must satisfy some property)
- Universal properties (all objects of a type share a characteristic)

Domain PDDL

{domain_pddl}

Smallest Problem PDDL

{smallest_problem_pddl}

Largest Problem PDDL

{largest_problem_pddl}

Instructions

Generate derived predicates and constraint candidates. Focus on constraints that:
1. Are semantically meaningful and capture domain-specific knowledge
2. Are likely to hold in valid problems initially
3. Express important properties such as:
   - Mutual exclusions: forall x -> not(P1(x) and P2(x))
   - Implications: forall x -> (P1(x) -> P2(x))
   - Disjunctions: forall x -> (P1(x) or P2(x))
   - Existential requirements: forall x -> exists y -> P1(x, y)
   - Universal properties: exists x -> forall y -> P1(x, y)
   - Uniqueness with variable inequality: forall x -> forall y -> forall z -> ((P(x,y) and P(x,z)) -> y != z)
4. Use derived predicates where they simplify constraint expression
5. Use 'not' for formula negation (notP(x)) and 'different_objs' ONLY for variable inequality (x != y)

Ensure your constraints follow the correct syntax and use only predicates and types defined in the domain or derived predicates you create.

Available Predicates

The following predicates are available for use in constraints:
{predicates_list}

Note: Goal-compiled predicates (ending in '_g') represent atoms from the problem goals and can be used to express constraints about goal relations. Recall that these are only a partial assignment, those from the `:goal` section of the problem PDDL.

Automatically Uncovering Intended Domain Constraints in Automated Planning

Appendix

D Experimental Setup and Runtime Details

Domains. The domains are: Grid (IPC 1998) (McDermott 2000), Blocksworld (IPC 2000) (Bacchus 2001), Logistics (IPC 2000) (Bacchus 2001), Transport (IPC 2008) (Helmert 2009) and Floortile (IPC 2011) (Coles et al. 2012). Grid is untyped, as it was never typed in the IPC, and allows us to evaluate $\mathcal{C}_{\text{TI}}$ on a domain without types.

LLM Configuration. For $\mathcal{C}_{\text{LLM}}$, we use GPT-5-2025-08-07 with temperature 1.0, the only option. We generate $\mathcal{C}_{\text{LLM}}$ for each domain only once, each using on average 9.8k input tokens and 55.7k output tokens ($0.57 USD).

Reduction and Hardware. For the reduction algorithm (Algorithm 2), we use Z3 (De Moura and Bjørner 2008) with a timeout of 30 seconds per constraint. This timeout and the algorithm's greedy nature cause our set of reduced constraints to be a superset of the minimal one. We run experiments on a Ryzen 7 7800X3D processor with 32 GiB RAM.

Runtimes. Since we do not focus on runtime optimization in this paper, we report single-run wall-clock timings for transparency. Table 2 reports per-stage times (in minutes) for the $\mathcal{C}_{\text{TI}}$ pipeline on the hardware above. The largest contributors are evaluating all constraint-task pairs and the final reduction step.

Domain	Generate	Evaluate	Sim. Exhaustive	Sim. Eager	Sim. Lazy	Reduce	Total
Blocksworld	2.86	0.67	0.15	0.20	0.16	22.57	26.61
Logistics	1.91	2.08	0.37	0.56	0.28	30.55	35.75
Transport	4.72	63.67	0.36	0.42	0.32	32.25	101.74
Floortile	29.57	332.85	1.19	3.94	2.62	262.84	633.01
Grid	63.68	803.25	0.04	0.31	0.29	189.35	1056.92

Table 2: Runtime decomposition for $\mathcal{C}_{\text{TI}}$ in minutes, measured from one full run per domain. "Sim."columns denote simulated traversal runs over the same evaluated constraint-task outcomes.

Automatically Uncovering Intended Domain Constraints in Automated Planning

Appendix

E Learned Domain Constraints — $\mathcal{C}_{\text{TI}}$

Blocksworld

$$\text{on}_{tc}(x\colon block,\ y\colon block) \coloneqq \text{on}(x,y) \vee \underset{z\colon block}{\exists}(\text{on}(x,z) \wedge \text{on}_{tc}(z,y))$$

$$\top \rightarrow \ \text{handempty}() \tag{B1}$$
$$\top \rightarrow \neg\,\text{handempty}_{\text{g}}() \tag{B2}$$
$$\top \rightarrow \neg\underset{a}{\exists}\,\text{clear}_{\text{g}}(a) \tag{B3}$$
$$\top \rightarrow \neg\underset{a}{\exists}\,\text{holding}(a) \tag{B4}$$
$$\top \rightarrow \neg\underset{a}{\exists}\,\text{holding}_{\text{g}}(a) \tag{B5}$$
$$\top \rightarrow \neg\underset{a}{\exists}\,\text{ontable}_{\text{g}}(a) \tag{B6}$$
$$\top \rightarrow \ \underset{a}{\exists}\,\underset{b}{\exists}\,\text{on}_{\text{g}}(a,b) \tag{B7}$$
$$\top \rightarrow \neg\underset{b}{\forall}\,\underset{a}{\exists}\,\text{on}_{\text{g}}(a,b) \tag{B8}$$
$$\top \rightarrow \neg\underset{a}{\forall}\,\underset{b}{\exists}\,\text{on}_{\text{g}}(a,b) \tag{B9}$$
$$\top \rightarrow \neg\underset{b}{\forall}\,\underset{a}{\exists}\,\text{on}_{tc}(a,b) \tag{B10}$$
$$\top \rightarrow \neg\underset{a}{\forall}\,\underset{b}{\exists}\,\text{on}_{tc}(a,b) \tag{B11}$$
$$\underset{a}{\forall}\neg\,\text{clear}(a) \rightarrow \ \underset{b}{\exists}\,\text{on}_{tc}(b,a) \tag{B12}$$
$$\underset{a}{\forall}\neg\,\text{ontable}(a) \rightarrow \ \underset{c}{\exists}\,\text{on}_{tc}(a,c) \tag{B13}$$
$$\underset{a,b}{\forall}\ \text{on}(a,b) \rightarrow \neg\underset{d\colon d\neq b}{\exists}\,\text{on}(a,d) \tag{B14}$$
$$\underset{a,b}{\forall}\ \text{on}(a,b) \rightarrow \neg\underset{c\colon c\neq a}{\exists}\,\text{on}(c,b) \tag{B15}$$
$$\underset{a,b}{\forall}\ \text{on}(a,b) \rightarrow \ \underset{c\colon c\in\{a,b\}}{\exists}\,\underset{d}{\exists}\,\text{on}_{\text{g}}(c,d) \tag{B16}$$
$$\underset{a,b}{\forall}\ \text{on}(a,b) \rightarrow \ \underset{d\colon d\in\{a,b\}}{\exists}\,\underset{c}{\exists}\,\text{on}_{\text{g}}(c,d) \tag{B17}$$
$$\underset{a,b}{\forall}\ \text{on}(a,b) \rightarrow \neg\underset{d\colon d\in\{a,b\}}{\forall}\,\underset{c\colon c\in\{a,b\}}{\exists}\,\text{on}_{\text{g}}(c,d) \tag{B18}$$
$$\underset{a,b}{\forall}\ \text{on}_{\text{g}}(a,b) \rightarrow \neg\underset{d\colon d\neq b}{\exists}\,\text{on}_{\text{g}}(a,d) \tag{B19}$$
$$\underset{a,b}{\forall}\ \text{on}_{\text{g}}(a,b) \rightarrow \neg\underset{c\colon c\neq a}{\exists}\,\text{on}_{\text{g}}(c,b) \tag{B20}$$
$$\underset{a,b}{\forall}\ \text{on}_{\text{g}}(a,b) \rightarrow \neg\underset{d\colon d\in\{a,b\}}{\exists}\,\text{on}_{\text{g}}(b,d) \tag{B21}$$
$$\underset{a,b}{\forall}\ \text{on}_{tc}(a,b) \rightarrow \neg\,\text{clear}(b) \tag{B22}$$
$$\underset{a,b}{\forall}\ \text{on}_{tc}(a,b) \rightarrow \neg\,\text{ontable}(a) \tag{B23}$$
$$\underset{a,b}{\forall}\ \text{on}_{tc}(a,b) \rightarrow \ \underset{d}{\exists}\,\text{on}(a,d) \tag{B24}$$
$$\underset{a,b}{\forall}\ \text{on}_{tc}(a,b) \rightarrow \ \underset{c}{\exists}\,\text{on}(c,b) \tag{B25}$$

$$\underset{a,b}{\forall} \; \mathrm{on}_{tc}(a,b) \rightarrow \underset{c:\, c\in\{a,b\}}{\exists} \underset{d}{\exists} \mathrm{on}_{\mathrm{g}}(c,d) \tag{B26}$$

$$\underset{a,b}{\forall} \; \mathrm{on}_{tc}(a,b) \rightarrow \underset{d:\, d\in\{a,b\}}{\exists} \underset{c}{\exists} \mathrm{on}_{\mathrm{g}}(c,d) \tag{B27}$$

$$\underset{a,b}{\forall} \; \mathrm{on}_{tc}(a,b) \rightarrow \neg \underset{d:\, d\in\{a,b\}}{\forall} \underset{c:\, c\in\{a,b\}}{\exists} \mathrm{on}_{\mathrm{g}}(c,d) \tag{B28}$$

$$\underset{a,b}{\forall} \; \mathrm{on}_{tc}(a,b) \rightarrow \neg \underset{c:\, c\in\{a,b\}}{\forall} \underset{d:\, d\in\{a,b\}}{\exists} \mathrm{on}_{\mathrm{g}}(c,d) \tag{B29}$$

$$\underset{a,b}{\forall} \; \mathrm{on}_{tc}(a,b) \rightarrow \underset{c:\, c\neq b}{\exists} \underset{d}{\exists} \mathrm{on}_{tc}(c,d) \tag{B30}$$

$$\underset{a,b}{\forall} \; \mathrm{on}_{tc}(a,b) \rightarrow \neg \underset{d:\, d\in\{a,b\}}{\exists} \mathrm{on}_{tc}(b,d) \tag{B31}$$

$$\underset{a,b}{\forall} \; \mathrm{on}_{tc}(a,b) \rightarrow \neg \underset{d:\, d\in\{a,b\}}{\forall} \underset{c:\, c\in\{a,b\}}{\exists} \mathrm{on}_{tc}(c,d) \tag{B32}$$

$$\underset{a,b}{\forall} \neg \mathrm{on}_{tc}(a,b) \rightarrow \neg \underset{d:\, d\in\{a,b\}}{\forall} \underset{c:\, c\in\{a,b\}}{\exists} \mathrm{on}_{\mathrm{g}}(c,d) \tag{B33}$$

$$\underset{a,b}{\forall} \neg \mathrm{on}_{tc}(a,b) \rightarrow \neg \underset{c:\, c\in\{a,b\}}{\forall} \underset{d:\, d\in\{a,b\}}{\exists} \mathrm{on}_{\mathrm{g}}(c,d) \tag{B34}$$

$$\underset{a,b}{\forall} \neg \mathrm{on}_{tc}(a,b) \rightarrow \neg \underset{d:\, d\in\{a,b\}}{\exists} \mathrm{on}_{tc}(a,d) \tag{B35}$$

$$\underset{a,b}{\forall} \neg \mathrm{on}_{tc}(a,b) \rightarrow \neg \underset{d:\, d\in\{a,b\}}{\forall} \underset{c:\, c\in\{a,b\}}{\exists} \mathrm{on}_{tc}(c,d) \tag{B36}$$

Valid Blocksworld constraints after automatic pruning.

Manual interpretation of the learned constraints:

- B1: The hand starts empty.
- B2: The goal is never to have the hand empty.
- B3: The goal is never to have a clear block.
- B4: No block is being held at the start.
- B5: The goal is never to be holding a block.
- B6: The goal is never to have a block on the table.
- B7: Some block is on a block in the goal state.
- B8: Some block does not have a block on it in the goal state.
- B9: Some block does not have a block below it in the goal state.
- B10: Some block does not have a block above it. Implied by B31, since a cycle is required to have all blocks above one another.
- B11: Some block is not above any block. Implied by B31.
- B12: If a block is not clear, some block is above it.
- B13: If a block is not on the table, it is above some block.
- B14: If a is on b, a is not on any other block.
- B15: If a is on b, no other block is on b.
- B16: If a is on b, then either a or b is on some block d in the goal state. Implied by B26.
- B17: If a is on b, then either a or b has some block c on it in the goal state. Implied by B27.
- B18: If a is on b, either a or b have neither a or b on them in the goal state. Implied by B28.
- B19: If a is on b in the goal state, a is not on any other block in the goal state.
- B20: If a is on b in the goal state, no other block is on b in the goal state.
- B21: If a is on b in the goal state,b is not on either a nor b. This prevents cycles of height 2 in the goal state, the best possible estimate without transitive closure of goal predicates.
- B22: If a is above b, b is not clear.
- B23: If a is above b, a is not on the table.
- B24: If a is above b, a is on some block d. A totality not caught by our totality check.
- B25: If a is above b, some block c is on b. A totality not caught by our totality check.

- B26: If a is above b, then either a or b is on some block d in the goal state. Means that at most one block per stack could form the bottom of a goal stack. An estimate of the fact that there is exactly one goal stack, the best possible without transitive closure of goal predicates.
- B27: If a is above b, then either a or b has some block c on it in the goal state. Means that at most one block per stack could form the top of a stack. An estimate of the fact that there is exactly one goal stack, the best possible without transitive closure of goal predicates.
- B28: If a is above b, either a or b is on neither a nor b in the goal states. Prevents goal cycles of height 2. Implied by B21.
- B29: If a is above b, either a or b is under neither a nor b in the goal state. Prevents goal cycles of height 2. Implied by B21.
- B30: If a is above b, some block other than b is above some block. Dissallows a state with a single stack of height 2 and all other blocks on the table.
- B31: If a is above b, b is not above either a or b. Prevents cycles.
- B32: If a is above b, a is not above either a or b ($d = a$ is irrelevant since $c = a$ always satisfies it). Implied by B31.
- B33: If a is not above b, either a or b is on neither a nor b in the goal states. Prevents goal cycles of height 2. Implied by B21.
- B34: If a is not above b, either a or b is under neither a nor b in the goal state. Prevents goal cycles of height 2. Implied by B21.
- B35: If a is not above b, a is not above itself. Implied by B31.
- B36: If a is not above b, either a is not above itself or b is not above a nor itself. Implied by B31.

By manually pruning the above constraints, we end up with the following set of 22 dominant constraints that imply all others:

1. B1: The hand starts empty.
2. B2: The goal is never to have the hand empty.
3. B3: The goal is never to have a clear block.
4. B4: No block is being held at the start.
5. B5: The goal is never to be holding a block.
6. B6: The goal is never to have a block on the table.
7. B7: Some block is on a block in the goal state.
8. B8: Some block does not have a block on it in the goal state.
9. B9: Some block does not have a block below it in the goal state.
10. B12: If a block is not clear, some block is above it.
11. B13: If a block is not on the table, it is above some block.
12. B14: If a is on b, a is not on any other block.
13. B15: If a is on b, no other block is on b.
14. B19: If a is on b in the goal state, a is not on any other block in the goal state.
15. B20: If a is on b in the goal state, no other block is on b in the goal state.
16. B21: If a is on b in the goal state, b is not on either a nor b. This prevents cycles of height 2 in the goal state, the best possible estimate without transitive closure of goal predicates.
17. B22: If a is above b, b is not clear.
18. B23: If a is above b, a is not on the table.
19. B26: If a is above b, then either a or b is on some block d in the goal state. Means that at most one block per stack could form the bottom of a goal stack. An estimate of the fact that there is exactly one goal stack, the best possible without transitive closure of goal predicates.
20. B27: If a is above b, then either a or b has some block c on it in the goal state. Means that at most one block per stack could form the top of a stack. An estimate of the fact that there is exactly one goal stack, the best possible without transitive closure of goal predicates.
21. B30: If a is above b, some block other than b is above some block. Dissallows a state with a single stack of height 2 and all other blocks on the table.
22. B31: If a is above b, b is not above either a or b. Prevents cycles.

Logistics

$$\top \rightarrow \underset{a\colon (\mathrm{type}(a) \leq^t \text{Package})}{\forall} \underset{b\colon (\mathrm{type}(b) \leq^t \text{Place})}{\exists} \mathrm{at}(a,b) \tag{L1}$$

$$\top \rightarrow \underset{a\colon (\mathrm{type}(a) \leq^t \text{Truck})}{\forall} \underset{b\colon (\mathrm{type}(b) \leq^t \text{Place})}{\exists} \mathrm{at}(a,b) \tag{L2}$$

$$\top \rightarrow \underset{a\colon (\mathrm{type}(a) \leq^t \text{Vehicle})}{\exists} \underset{b\colon (\mathrm{type}(b) \leq^t \text{Location})}{\exists} \mathrm{at}(a,b) \tag{L3}$$

$$\top \rightarrow \neg \underset{b\colon (\mathrm{type}(b) \leq^t \text{Place})}{\exists} \underset{a\colon (\mathrm{type}(a) \leq^t \text{Package})}{\forall} \mathrm{at}(a,b) \tag{L4}$$

$$\top \rightarrow \neg \underset{b\colon (\mathrm{type}(b) \leq^t \text{Place})}{\exists} \underset{a\colon (\mathrm{type}(a) \leq^t \text{Truck})}{\forall} \mathrm{at}(a,b) \tag{L5}$$

$$\top \rightarrow \neg \underset{b\colon (\mathrm{type}(b) \leq^t \text{Airport})}{\forall} \underset{a\colon (\mathrm{type}(a) \leq^t \text{Physobj})}{\exists} \mathrm{at}(a,b) \tag{L6}$$

$$\top \rightarrow \neg \underset{a\colon (\mathrm{type}(a) \leq^t \text{Package})}{\forall} \underset{b\colon (\mathrm{type}(b) \leq^t \text{Airport})}{\exists} \mathrm{at}(a,b) \tag{L7}$$

$$\top \rightarrow \neg \underset{a\colon (\mathrm{type}(a) \leq^t \text{Truck})}{\forall} \underset{b\colon (\mathrm{type}(b) \leq^t \text{Airport})}{\exists} \mathrm{at}(a,b) \tag{L8}$$

$$\top \rightarrow \neg \underset{a\colon (\mathrm{type}(a) \leq^t \text{Vehicle})}{\forall} \underset{b\colon (\mathrm{type}(b) \leq^t \text{Location})}{\exists} \mathrm{at}(a,b) \tag{L9}$$

$$\top \rightarrow \underset{a\colon (\mathrm{type}(a) \leq^t \text{Physobj})}{\exists} \underset{b\colon (\mathrm{type}(b) \leq^t \text{Airport})}{\exists} \mathrm{at_g}(a,b) \tag{L10}$$

$$\top \rightarrow \underset{a\colon (\mathrm{type}(a) \leq^t \text{Physobj})}{\exists} \underset{b\colon (\mathrm{type}(b) \leq^t \text{Location})}{\exists} \mathrm{at_g}(a,b) \tag{L11}$$

$$\top \rightarrow \neg \underset{a\colon (\mathrm{type}(a) \leq^t \text{Physobj})}{\forall} \underset{b\colon (\mathrm{type}(b) \leq^t \text{Place})}{\exists} \mathrm{at_g}(a,b) \tag{L12}$$

$$\top \rightarrow \underset{b\colon (\mathrm{type}(b) \leq^t \text{City})}{\forall} \underset{a\colon (\mathrm{type}(a) \leq^t \text{Airport})}{\exists} \text{in-city}(a,b) \tag{L13}$$

$$\top \rightarrow \underset{b\colon (\mathrm{type}(b) \leq^t \text{City})}{\forall} \underset{a\colon (\mathrm{type}(a) \leq^t \text{Location})}{\exists} \text{in-city}(a,b) \tag{L14}$$

$$\top \rightarrow \underset{a\colon (\mathrm{type}(a) \leq^t \text{Place})}{\forall} \underset{b\colon (\mathrm{type}(b) \leq^t \text{City})}{\exists} \text{in-city}(a,b) \tag{L15}$$

$$\top \rightarrow \neg \underset{a\colon (\mathrm{type}(a) \leq^t \text{Place})}{\exists} \underset{b\colon (\mathrm{type}(b) \leq^t \text{City})}{\forall} \text{in-city}(a,b) \tag{L16}$$

$$\top \rightarrow \neg \underset{a\colon (\mathrm{type}(a) \leq^t \text{Physobj})}{\exists} (\mathrm{type}(a) = \text{Physobj}) \tag{L17}$$

$$\top \rightarrow \underset{a\colon (\mathrm{type}(a) \leq^t \text{Airplane})}{\exists} (\mathrm{type}(a) \leq^t \text{Airplane}) \tag{L18}$$

$$\top \rightarrow \neg \underset{a\colon (\mathrm{type}(a) \leq^t \text{Place})}{\exists} (\mathrm{type}(a) = \text{Place}) \tag{L19}$$

$$\top \rightarrow \neg \underset{a\colon (\mathrm{type}(a) \leq^t \text{Vehicle})}{\exists} (\mathrm{type}(a) = \text{Vehicle}) \tag{L20}$$

$$\underset{\substack{a,b\colon \\ (\mathrm{type}(a) \leq^t \text{Physobj}) \wedge (\mathrm{type}(b) \leq^t \text{Place})}}{\forall} \mathrm{at}(a,b) \rightarrow \neg \underset{\substack{d\colon (\mathrm{type}(d) \leq^t \text{Place}) \wedge \\ d \neq b}}{\exists} \mathrm{at}(a,d) \tag{L21}$$

$$\underset{\substack{a,b\colon \\ (\mathrm{type}(a) \leq^t \text{Physobj}) \wedge (\mathrm{type}(b) \leq^t \text{Place})}}{\forall} \mathrm{at_g}(a,b) \rightarrow \neg \underset{\substack{d\colon (\mathrm{type}(d) \leq^t \text{Place}) \wedge \\ d \neq b}}{\exists} \mathrm{at_g}(a,d) \tag{L22}$$

$$\underset{\substack{a,b\colon \\ (\mathrm{type}(a) \leq^t \text{Physobj}) \wedge (\mathrm{type}(b) \leq^t \text{Place})}}{\forall} \mathrm{at_g}(a,b) \rightarrow \neg \underset{c\colon (\mathrm{type}(c) \leq^t \text{Package})}{\forall} \underset{\substack{d\colon (\mathrm{type}(d) \leq^t \text{Place}) \wedge \\ d \neq b}}{\exists} \mathrm{at_g}(c,d) \tag{L23}$$

$$\underset{\substack{a,b\colon \\ (\mathrm{type}(a) \leq^t \text{Place}) \wedge (\mathrm{type}(b) \leq^t \text{City})}}{\forall} \text{in-city}(a,b) \rightarrow \neg \underset{\substack{d\colon (\mathrm{type}(d) \leq^t \text{City}) \wedge \\ d \neq b}}{\exists} \text{in-city}(a,d) \tag{L24}$$

$$\underset{\substack{a,b\colon \\ (\mathrm{type}(a) \leq^t \text{Package}) \wedge (\mathrm{type}(b) \leq^t \text{Vehicle})}}{\forall} \neg \mathrm{in_g}(a,b) \rightarrow \neg \underset{d\colon (\mathrm{type}(d) \leq^t \text{Airport})}{\forall} \underset{\substack{c\colon (\mathrm{type}(c) \leq^t \text{Physobj}) \wedge \\ c \in \{a,b\}}}{\exists} \mathrm{at_g}(c,d) \tag{L25}$$

$$\underset{\substack{a,b\colon \\ (\mathrm{type}(a) \leq^t \text{Package}) \wedge (\mathrm{type}(b) \leq^t \text{Vehicle})}}{\forall} \neg \mathrm{in_g}(a,b) \rightarrow \neg \underset{d\colon (\mathrm{type}(d) \leq^t \text{Location})}{\forall} \underset{\substack{c\colon (\mathrm{type}(c) \leq^t \text{Physobj}) \wedge \\ c \in \{a,b\}}}{\exists} \mathrm{at_g}(c,d) \tag{L26}$$

$$\underset{a:\ (\text{type}(a)\leq^t \text{Airplane})}{\forall} (\text{type}(a) \leq^t \text{Airplane}) \rightarrow \underset{\substack{b:\ (\text{type}(b)\leq^t \text{Airplane})\wedge \\ b\neq a}}{\forall} \underset{c:\ (\text{type}(c)\leq^t \text{Airport})}{\exists} \text{at}(b,c) \tag{L27}$$

$$\underset{a:\ (\text{type}(a)\leq^t \text{Airplane})}{\forall} (\text{type}(a) \leq^t \text{Airplane}) \rightarrow \underset{\substack{b:\ (\text{type}(b)\leq^t \text{Physobj})\wedge \\ b\neq a}}{\forall} \underset{c:\ (\text{type}(c)\leq^t \text{Place})}{\exists} \text{at}(b,c) \tag{L28}$$

$$\underset{a:\ (\text{type}(a)\leq^t \text{Airport})}{\forall} (\text{type}(a) \leq^t \text{Airport}) \rightarrow \neg \underset{c:\ (\text{type}(c)\leq^t \text{City})}{\forall} \underset{\substack{b:\ (\text{type}(b)\leq^t \text{Airport})\wedge \\ b\neq a}}{\exists} \text{in-city}(b,c) \tag{L29}$$

$$\underset{a:\ (\text{type}(a)\leq^t \text{City})}{\forall} (\text{type}(a) \leq^t \text{City}) \rightarrow \neg \underset{b:\ (\text{type}(b)\leq^t \text{Place})}{\exists} \text{in-city}_{\text{g}}(b,a) \tag{L30}$$

$$\underset{a:\ (\text{type}(a)\leq^t \text{Place})}{\forall} (\text{type}(a) \leq^t \text{Place}) \rightarrow \neg \underset{b:\ (\text{type}(b)\leq^t \text{Vehicle})}{\exists} \text{at}_{\text{g}}(b,a) \tag{L31}$$

$$\underset{a:\ (\text{type}(a)\leq^t \text{Location})}{\forall} (\text{type}(a) \leq^t \text{Location}) \rightarrow \neg \underset{b:\ (\text{type}(b)\leq^t \text{Airplane})}{\exists} \text{at}(b,a) \tag{L32}$$

$$\underset{a:\ (\text{type}(a)\leq^t \text{Location})}{\forall} (\text{type}(a) \leq^t \text{Location}) \rightarrow \neg \underset{b:\ (\text{type}(b)\leq^t \text{Package})}{\forall} \underset{\substack{c:\ (\text{type}(c)\leq^t \text{Location})\wedge \\ c\neq a}}{\exists} \text{at}(b,c) \tag{L33}$$

$$\underset{a:\ (\text{type}(a)\leq^t \text{Vehicle})}{\forall} (\text{type}(a) \leq^t \text{Vehicle}) \rightarrow \neg \underset{b:\ (\text{type}(b)\leq^t \text{Package})}{\exists} \text{in}(b,a) \tag{L34}$$

$$\underset{a:\ (\text{type}(a)\leq^t \text{Vehicle})}{\forall} (\text{type}(a) \leq^t \text{Vehicle}) \rightarrow \neg \underset{b:\ (\text{type}(b)\leq^t \text{Package})}{\exists} \text{in}_{\text{g}}(b,a) \tag{L35}$$

$$\underset{a:\ (\text{type}(a)\leq^t \text{Truck})}{\forall} (\text{type}(a) \leq^t \text{Truck}) \rightarrow \neg \underset{c:\ (\text{type}(c)\leq^t \text{Location})}{\forall} \underset{\substack{b:\ (\text{type}(b)\leq^t \text{Vehicle})\wedge \\ b\neq a}}{\exists} \text{at}(b,c) \tag{L36}$$

All reduced Logistics constraints. Note that airports are not locations, however both airports and locations are places.

Transport

$$\text{cap-pre}_{tc}(x\colon Cap-Num,\ y\colon Cap-Num) \coloneqq \text{cap-pre}(x,y) \vee \exists_{z\colon Cap-Num}(\text{cap-pre}(x,z) \wedge \text{cap-pre}_{tc}(z,y))$$

$$\text{road}_{tc}(x\colon location,\ y\colon location) \coloneqq \text{road}(x,y) \vee \exists_{z\colon location}(\text{road}(x,z) \wedge \text{road}_{tc}(z,y))$$

$$\top \rightarrow \forall_{a\colon (\text{type}(a)\leq^t \text{Locatable})}\ \exists_{b\colon (\text{type}(b)\leq^t \text{Location})} \text{at}(a,b) \tag{T1}$$

$$\top \rightarrow \neg \forall_{b\colon (\text{type}(b)\leq^t \text{Location})}\ \exists_{a\colon (\text{type}(a)\leq^t \text{Locatable})} \text{at}(a,b) \tag{T2}$$

$$\top \rightarrow \forall_{a\colon (\text{type}(a)\leq^t \text{Package})}\ \exists_{b\colon (\text{type}(b)\leq^t \text{Location})} \text{at}_{\text{g}}(a,b) \tag{T3}$$

$$\top \rightarrow \neg \forall_{a\colon (\text{type}(a)\leq^t \text{Locatable})}\ \exists_{b\colon (\text{type}(b)\leq^t \text{Location})} \text{at}_{\text{g}}(a,b) \tag{T4}$$

$$\top \rightarrow \neg \forall_{b\colon (\text{type}(b)\leq^t \text{Location})}\ \exists_{a\colon (\text{type}(a)\leq^t \text{Locatable})} \text{at}_{\text{g}}(a,b) \tag{T5}$$

$$\top \rightarrow \forall_{a\colon (\text{type}(a)\leq^t \text{Vehicle})}\ \exists_{b\colon (\text{type}(b)\leq^t \text{Cap-Num})} \text{capacity}(a,b) \tag{T6}$$

$$\top \rightarrow \exists_{a\colon (\text{type}(a)\leq^t \text{Cap-Num})}\ \exists_{b\colon (\text{type}(b)\leq^t \text{Cap-Num})} \text{cap-pre}_{tc}(a,b) \tag{T7}$$

$$\top \rightarrow \neg \forall_{b\colon (\text{type}(b)\leq^t \text{Cap-Num})}\ \exists_{a\colon (\text{type}(a)\leq^t \text{Cap-Num})} \text{cap-pre}_{tc}(a,b) \tag{T8}$$

$$\top \rightarrow \neg \forall_{a\colon (\text{type}(a)\leq^t \text{Cap-Num})}\ \exists_{b\colon (\text{type}(b)\leq^t \text{Cap-Num})} \text{cap-pre}_{tc}(a,b) \tag{T9}$$

$$\top \rightarrow \forall_{a\colon (\text{type}(a)\leq^t \text{Location})}\ \exists_{b\colon (\text{type}(b)\leq^t \text{Location})} \text{road}(a,b) \tag{T10}$$

$$\top \rightarrow \neg \exists_{a\colon (\text{type}(a)\leq^t \text{Locatable})} (\text{type}(a) = \text{Locatable}) \tag{T11}$$

$$\top \rightarrow \neg \exists_{a\colon (\text{type}(a)\leq^t \text{Target})} (\text{type}(a) \leq^t \text{Target}) \tag{T12}$$

$$\forall_{\substack{a,b\colon \\ (\text{type}(a)\leq^t \text{Locatable})\wedge(\text{type}(b)\leq^t \text{Location})}} \text{at}(a,b) \rightarrow \neg \exists_{\substack{d\colon (\text{type}(d)\leq^t \text{Location})\wedge \\ d\neq b}} \text{at}(a,d) \tag{T13}$$

$$\forall_{\substack{a,b\colon \\ (\text{type}(a)\leq^t \text{Locatable})\wedge(\text{type}(b)\leq^t \text{Location})}} \text{at}(a,b) \rightarrow \neg\, \text{at}_{\text{g}}(a,b) \tag{T14}$$

$$\forall_{\substack{a,b\colon \\ (\text{type}(a)\leq^t \text{Locatable})\wedge(\text{type}(b)\leq^t \text{Location})}} \text{at}(a,b) \rightarrow \neg \forall_{c\colon (\text{type}(c)\leq^t \text{Package})} \text{at}_{\text{g}}(c,b) \tag{T15}$$

$$\forall_{\substack{a,b\colon \\ (\text{type}(a)\leq^t \text{Locatable})\wedge(\text{type}(b)\leq^t \text{Location})}} \text{at}_{\text{g}}(a,b) \rightarrow \neg \exists_{\substack{d\colon (\text{type}(d)\leq^t \text{Location})\wedge \\ d\neq b}} \text{at}_{\text{g}}(a,d) \tag{T16}$$

$$\forall_{\substack{a,b\colon \\ (\text{type}(a)\leq^t \text{Locatable})\wedge(\text{type}(b)\leq^t \text{Location})}} \text{at}_{\text{g}}(a,b) \rightarrow \neg \forall_{c\colon (\text{type}(c)\leq^t \text{Package})}\ \exists_{\substack{d\colon (\text{type}(d)\leq^t \text{Location})\wedge \\ d\neq b}} \text{at}_{\text{g}}(c,d) \tag{T17}$$

$$\forall_{\substack{a,b\colon \\ (\text{type}(a)\leq^t \text{Vehicle})\wedge(\text{type}(b)\leq^t \text{Cap-Num})}} \text{capacity}(a,b) \rightarrow \neg \exists_{\substack{d\colon (\text{type}(d)\leq^t \text{Cap-Num})\wedge \\ d\neq b}} \text{capacity}(a,d) \tag{T18}$$

$$\forall_{\substack{a,b\colon \\ (\text{type}(a)\leq^t \text{Vehicle})\wedge(\text{type}(b)\leq^t \text{Cap-Num})}} \text{capacity}(a,b) \rightarrow \exists_{c\colon (\text{type}(c)\leq^t \text{Cap-Num})} \text{cap-pre}_{tc}(c,b) \tag{T19}$$

$$\forall_{\substack{a,b\colon \\ (\text{type}(a)\leq^t \text{Cap-Num})\wedge(\text{type}(b)\leq^t \text{Cap-Num})}} \text{cap-pre}(a,b) \rightarrow \neg \forall_{\substack{d\colon (\text{type}(d)\leq^t \text{Cap-Num})\wedge \\ d\in\{a,b\}}}\ \forall_{c\colon (\text{type}(c)\leq^t \text{Vehicle})} \text{capacity}(c,d) \tag{T20}$$

$$\forall_{\substack{a,b\colon \\ (\text{type}(a)\leq^t \text{Cap-Num})\wedge(\text{type}(b)\leq^t \text{Cap-Num})}} \text{cap-pre}(a,b) \rightarrow \neg \exists_{\substack{d\colon (\text{type}(d)\leq^t \text{Cap-Num})\wedge \\ d\neq b}} \text{cap-pre}(a,d) \tag{T21}$$

$$\forall_{\substack{a,b\colon \\ (\text{type}(a)\leq^t \text{Cap-Num})\wedge(\text{type}(b)\leq^t \text{Cap-Num})}} \text{cap-pre}(a,b) \rightarrow \neg \exists_{\substack{c\colon (\text{type}(c)\leq^t \text{Cap-Num})\wedge \\ c\neq a}} \text{cap-pre}(c,b) \tag{T22}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Cap-Num})\wedge(\text{type}(b)\leq^t\text{Cap-Num})}} \text{cap-pre}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Cap-Num})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Cap-Num})\wedge\\ c\in\{a,b\}}} \text{cap-pre}_{tc}(c,d) \quad \text{(T23)}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Cap-Num})\wedge(\text{type}(b)\leq^t\text{Cap-Num})}} \neg\,\text{cap-pre}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Cap-Num})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Cap-Num})\wedge\\ c\in\{a,b\}}} \text{cap-pre}_{tc}(c,d) \quad \text{(T24)}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Cap-Num})\wedge(\text{type}(b)\leq^t\text{Cap-Num})}} \text{cap-pre}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Cap-Num})\wedge\\ d\in\{a,b\}}} \forall_{c:\ (\text{type}(c)\leq^t\text{Vehicle})} \text{capacity}(c,d) \quad \text{(T25)}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Cap-Num})\wedge(\text{type}(b)\leq^t\text{Cap-Num})}} \text{cap-pre}_{tc}(a,b) \rightarrow \exists_{d:\ (\text{type}(d)\leq^t\text{Cap-Num})} \text{cap-pre}(a,d) \quad \text{(T26)}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Cap-Num})\wedge(\text{type}(b)\leq^t\text{Cap-Num})}} \text{cap-pre}_{tc}(a,b) \rightarrow \exists_{c:\ (\text{type}(c)\leq^t\text{Cap-Num})} \text{cap-pre}(c,b) \quad \text{(T27)}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Cap-Num})\wedge(\text{type}(b)\leq^t\text{Cap-Num})}} \text{cap-pre}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Cap-Num})\wedge\\ d\in\{a,b\}}} \text{cap-pre}_{tc}(b,d) \quad \text{(T28)}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Cap-Num})\wedge(\text{type}(b)\leq^t\text{Cap-Num})}} \neg\,\text{cap-pre}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Cap-Num})\wedge\\ d\in\{a,b\}}} \text{cap-pre}_{tc}(a,d) \quad \text{(T29)}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Location})\wedge(\text{type}(b)\leq^t\text{Location})}} \text{road}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Location})\wedge\\ d\in\{a,b\}}} \forall_{c:\ (\text{type}(c)\leq^t\text{Package})} \text{at}_{\text{g}}(c,d) \quad \text{(T30)}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Location})\wedge(\text{type}(b)\leq^t\text{Location})}} \text{road}(a,b) \rightarrow \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Location})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Location})\wedge\\ c\in\{a,b\}}} \text{road}(c,d) \quad \text{(T31)}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Location})\wedge(\text{type}(b)\leq^t\text{Location})}} \text{road}(a,b) \rightarrow \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Location})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Location})\wedge\\ d\in\{a,b\}}} \text{road}(c,d) \quad \text{(T32)}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Location})\wedge(\text{type}(b)\leq^t\text{Location})}} \neg\,\text{road}(a,b) \rightarrow \neg \forall_{c:\ (\text{type}(c)\leq^t\text{Locatable})} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Location})\wedge\\ d\in\{a,b\}}} \text{at}(c,d) \quad \text{(T33)}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Location})\wedge(\text{type}(b)\leq^t\text{Location})}} \neg\,\text{road}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Location})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Location})\wedge\\ d\in\{a,b\}}} \text{road}(c,d) \quad \text{(T34)}$$

$$\forall_{a:\ (\text{type}(a)\leq^t\text{Cap-Num})} (\text{type}(a) \leq^t \text{Cap-Num}) \rightarrow \neg \exists_{b:\ (\text{type}(b)\leq^t\text{Cap-Num})} \text{cap-pre}_{\text{g}}(b,a) \quad \text{(T35)}$$

$$\forall_{a:\ (\text{type}(a)\leq^t\text{Cap-Num})} (\text{type}(a) \leq^t \text{Cap-Num}) \rightarrow \neg \exists_{b:\ (\text{type}(b)\leq^t\text{Vehicle})} \text{capacity}_{\text{g}}(b,a) \quad \text{(T36)}$$

$$\forall_{a:\ (\text{type}(a)\leq^t\text{Location})} (\text{type}(a) \leq^t \text{Location}) \rightarrow \neg \exists_{b:\ (\text{type}(b)\leq^t\text{Vehicle})} \text{at}_{\text{g}}(b,a) \quad \text{(T37)}$$

$$\forall_{a:\ (\text{type}(a)\leq^t\text{Location})} (\text{type}(a) \leq^t \text{Location}) \rightarrow \neg\,\text{road}(a,a) \quad \text{(T38)}$$

$$\forall_{a:\ (\text{type}(a)\leq^t\text{Location})} (\text{type}(a) \leq^t \text{Location}) \rightarrow \neg \exists_{b:\ (\text{type}(b)\leq^t\text{Location})} \text{road}_{\text{g}}(b,a) \quad \text{(T39)}$$

$$\forall_{a:\ (\text{type}(a)\leq^t\text{Location})} (\text{type}(a) \leq^t \text{Location}) \rightarrow \forall_{b:\ (\text{type}(b)\leq^t\text{Location})} \text{road}_{tc}(b,a) \quad \text{(T40)}$$

$$\forall_{a:\ (\text{type}(a)\leq^t\text{Vehicle})} (\text{type}(a) \leq^t \text{Vehicle}) \rightarrow \neg \exists_{b:\ (\text{type}(b)\leq^t\text{Package})} \text{in}(b,a) \quad \text{(T41)}$$

$$\forall_{a:\ (\text{type}(a)\leq^t\text{Vehicle})} (\text{type}(a) \leq^t \text{Vehicle}) \rightarrow \neg \exists_{b:\ (\text{type}(b)\leq^t\text{Package})} \text{in}_{\text{g}}(b,a) \quad \text{(T42)}$$

All reduced Transport constraints. For brevity, `capacity-predecessor` is shortened to `cap-pre` and Capacity-Number to Cap-Num. The numerical fluent `road-length` was discarded before learning.

Floortile

$$\mathrm{down}_{tc}(x\colon tile,\ y\colon tile) \coloneqq \mathrm{down}(x,y) \vee \exists_{z\colon tile}(\mathrm{down}(x,z) \wedge \mathrm{down}_{tc}(z,y))$$
$$\mathrm{left}_{tc}(x\colon tile,\ y\colon tile) \coloneqq \mathrm{left}(x,y) \vee \exists_{z\colon tile}(\mathrm{left}(x,z) \wedge \mathrm{left}_{tc}(z,y))$$
$$\mathrm{right}_{tc}(x\colon tile,\ y\colon tile) \coloneqq \mathrm{right}(x,y) \vee \exists_{z\colon tile}(\mathrm{right}(x,z) \wedge \mathrm{right}_{tc}(z,y))$$
$$\mathrm{up}_{tc}(x\colon tile,\ y\colon tile) \coloneqq \mathrm{up}(x,y) \vee \exists_{z\colon tile}(\mathrm{up}(x,z) \wedge \mathrm{up}_{tc}(z,y))$$

$$\top \rightarrow \neg \exists_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Tile})} \forall_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Color})} \mathrm{painted}_{\mathrm{g}}(a,b) \tag{F1}$$
$$\top \rightarrow \neg \forall_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Tile})} \exists_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Color})} \mathrm{painted}_{\mathrm{g}}(a,b) \tag{F2}$$
$$\top \rightarrow \exists_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Tile})} \exists_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Tile})} \mathrm{right}_{tc}(a,b) \tag{F3}$$
$$\top \rightarrow \neg \forall_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Tile})} \exists_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Tile})} \mathrm{right}_{tc}(a,b) \tag{F4}$$
$$\top \rightarrow \neg \forall_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Tile})} \exists_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Tile})} \mathrm{right}_{tc}(a,b) \tag{F5}$$
$$\top \rightarrow \forall_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Robot})} \exists_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Tile})} \text{robot-at}(a,b) \tag{F6}$$
$$\top \rightarrow \neg \exists_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Tile})} \forall_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Robot})} \text{robot-at}(a,b) \tag{F7}$$
$$\top \rightarrow \neg \forall_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Tile})} \exists_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Robot})} \text{robot-at}(a,b) \tag{F8}$$
$$\top \rightarrow \forall_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Robot})} \exists_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Color})} \text{robot-has}(a,b) \tag{F9}$$
$$\top \rightarrow \forall_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Color})} \exists_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Robot})} \text{robot-has}(a,b) \tag{F10}$$
$$\top \rightarrow \neg \exists_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Color})} \forall_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Robot})} \text{robot-has}(a,b) \tag{F11}$$
$$\top \rightarrow \neg \exists_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Robot})} \forall_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Color})} \text{robot-has}(a,b) \tag{F12}$$
$$\top \rightarrow \exists_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Tile})} \exists_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Tile})} \mathrm{up}_{tc}(a,b) \tag{F13}$$
$$\top \rightarrow \neg \forall_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Tile})} \exists_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Tile})} \mathrm{up}_{tc}(a,b) \tag{F14}$$
$$\top \rightarrow \neg \forall_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Tile})} \exists_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Tile})} \mathrm{up}_{tc}(a,b) \tag{F15}$$
$$\forall_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Tile})} \neg \mathrm{clear}(a) \rightarrow \exists_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Tile})} \mathrm{right}(b,a) \tag{F16}$$
$$\forall_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Tile})} \neg \mathrm{clear}(a) \rightarrow \exists_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Robot})} \text{robot-at}(b,a) \tag{F17}$$
$$\forall_{a\colon (\mathrm{type}(a)\leq^t \mathrm{Tile})} \neg \mathrm{clear}(a) \rightarrow \neg \forall_{b\colon (\mathrm{type}(b)\leq^t \mathrm{Robot})} \exists_{c\colon (\mathrm{type}(c)\leq^t \mathrm{Tile}) \wedge c\neq a} \text{robot-at}(b,c) \tag{F18}$$
$$\forall_{a,b\colon (\mathrm{type}(a)\leq^t \mathrm{Tile}) \wedge (\mathrm{type}(b)\leq^t \mathrm{Tile})} \mathrm{down}(a,b) \rightarrow \neg \exists_{d\colon (\mathrm{type}(d)\leq^t \mathrm{Tile}) \wedge d\in\{a,b\}} \mathrm{down}_{tc}(b,d) \tag{F19}$$
$$\forall_{a,b\colon (\mathrm{type}(a)\leq^t \mathrm{Tile}) \wedge (\mathrm{type}(b)\leq^t \mathrm{Tile})} \mathrm{down}(a,b) \rightarrow \neg \exists_{c\colon (\mathrm{type}(c)\leq^t \mathrm{Tile}) \wedge c\in\{a,b\}} \exists_{d\colon (\mathrm{type}(d)\leq^t \mathrm{Tile}) \wedge d\in\{a,b\}} \mathrm{left}(c,d) \tag{F20}$$
$$\forall_{a,b\colon (\mathrm{type}(a)\leq^t \mathrm{Tile}) \wedge (\mathrm{type}(b)\leq^t \mathrm{Tile})} \mathrm{down}(a,b) \rightarrow \neg \exists_{d\colon (\mathrm{type}(d)\leq^t \mathrm{Tile})} \forall_{c\colon (\mathrm{type}(c)\leq^t \mathrm{Tile}) \wedge c\in\{a,b\}} \mathrm{left}(c,d) \tag{F21}$$
$$\forall_{a,b\colon (\mathrm{type}(a)\leq^t \mathrm{Tile}) \wedge (\mathrm{type}(b)\leq^t \mathrm{Tile})} \mathrm{down}(a,b) \rightarrow \neg \exists_{c\colon (\mathrm{type}(c)\leq^t \mathrm{Tile})} \forall_{d\colon (\mathrm{type}(d)\leq^t \mathrm{Tile}) \wedge d\in\{a,b\}} \mathrm{left}(c,d) \tag{F22}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}(c,d) \tag{F23}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F24}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F25}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F26}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F27}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \exists_{d:\ (\text{type}(d)\leq^t\text{Color})} \text{painted}_{\text{g}}(b,d) \tag{F28}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Color})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{painted}_{\text{g}}(c,d) \tag{F29}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}(c,d) \tag{F30}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F31}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}(c,d) \tag{F32}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F33}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F34}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F35}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F36}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F37}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \forall_{c:\ (\text{type}(c)\leq^t\text{Robot})} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{robot-at}(c,d) \tag{F38}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{c:\ (\text{type}(c)\leq^t\text{Robot})} \text{robot-at}(c,d) \tag{F39}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,a) \tag{F40}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F41}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F42}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(a,d) \tag{F43}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}(a,d) \tag{F44}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F45}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F46}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F47}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}(c,d) \tag{F48}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}(c,d) \tag{F49}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F50}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F51}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F52}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}(c,d) \tag{F53}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F54}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F55}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(b,d) \tag{F56}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F57}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{down}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F58}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F59}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}(c,d) \tag{F60}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F61}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}_{tc}(b,d) \tag{F62}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}(c,d) \tag{F63}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}(c,d) \tag{F64}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})}} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}(c,d) \tag{F65}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}(c,d) \tag{F66}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F67}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})}} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F68}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})}} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F69}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F70}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Color})}} \text{painted}_{\text{g}}(b,d) \tag{F71}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}(c,d) \tag{F72}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})}} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F73}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})}} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}(c,d) \tag{F74}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F75}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F76}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})}} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F77}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})}} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F78}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F79}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Robot})}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{robot-at}(c,d) \tag{F80}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Robot})}} \text{robot-at}(c,d) \tag{F81}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})}} \text{up}(b,d) \tag{F82}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})}} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F83}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})}} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F84}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{down}_{tc}(a,b) \rightarrow \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}_{tc}(c,a) \tag{F85}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \mathrm{down}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \mathrm{up}_{tc}(a,d) \tag{F86}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \mathrm{down}(c,d) \tag{F87}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \mathrm{down}_{tc}(a,d) \tag{F88}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \mathrm{down}_{tc}(c,d) \tag{F89}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \mathrm{left}(c,d) \tag{F90}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \forall_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \exists_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \mathrm{left}(c,d) \tag{F91}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \mathrm{left}_{tc}(c,d) \tag{F92}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \forall_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \exists_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \mathrm{left}_{tc}(c,d) \tag{F93}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \mathrm{right}(c,d) \tag{F94}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \forall_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \exists_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \mathrm{right}(c,d) \tag{F95}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \mathrm{right}_{tc}(c,d) \tag{F96}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \forall_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \exists_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \mathrm{right}_{tc}(c,d) \tag{F97}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \neg\,\mathrm{down}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \mathrm{up}_{tc}(b,d) \tag{F98}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \mathrm{left}(a,b) \rightarrow \neg \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \exists_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \mathrm{down}(c,d) \tag{F99}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \mathrm{left}(a,b) \rightarrow \neg \exists_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})} \forall_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \mathrm{down}(c,d) \tag{F100}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \mathrm{left}(a,b) \rightarrow \neg \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})} \forall_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \mathrm{down}(c,d) \tag{F101}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \mathrm{left}(a,b) \rightarrow \neg \forall_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \mathrm{down}(c,d) \tag{F102}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \mathrm{left}(a,b) \rightarrow \neg \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \exists_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \mathrm{down}_{tc}(c,d) \tag{F103}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \mathrm{left}(a,b) \rightarrow \neg \exists_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})} \forall_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \mathrm{down}_{tc}(c,d) \tag{F104}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \mathrm{left}(a,b) \rightarrow \neg \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})} \forall_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \mathrm{down}_{tc}(c,d) \tag{F105}$$

$$\forall_{a,b:\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})} \quad \mathrm{left}(a,b) \rightarrow \neg \forall_{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge c\in\{a,b\}} \mathrm{down}_{tc}(c,d) \tag{F106}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}_{tc}(b,d) \tag{F107}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Color})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{painted}_{\text{g}}(c,d) \tag{F108}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,a) \tag{F109}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F110}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}(c,d) \tag{F111}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(a,d) \tag{F112}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Robot})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{robot-at}(c,d) \tag{F113}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \forall_{c:\ (\text{type}(c)\leq^t\text{Robot})} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{robot-at}(c,d) \tag{F114}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F115}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F116}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F117}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F118}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F119}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F120}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F121}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F122}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F123}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}(c,d) \tag{F124}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F125}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F126}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}(a,d) \tag{F127}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{left}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{left}(c,d) \tag{F128}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{left}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{left}_{tc}(c,d) \tag{F129}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{left}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{left}_{tc}(c,d) \tag{F130}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{left}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{right}(b,d) \tag{F131}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{left}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{right}_{tc}(c,d) \tag{F132}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{left}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{right}_{tc}(c,d) \tag{F133}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{left}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{up}(c,d) \tag{F134}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{left}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{up}(c,d) \tag{F135}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{left}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{up}_{tc}(c,d) \tag{F136}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{left}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{up}_{tc}(c,d) \tag{F137}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{left}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{down}(c,d) \tag{F138}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{left}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})} \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{down}(c,d) \tag{F139}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{left}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})} \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{down}(c,d) \tag{F140}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{down}(c,d) \tag{F141}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{left}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{down}_{tc}(c,d) \tag{F142}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{left}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})} \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{down}_{tc}(c,d) \tag{F143}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{left}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})} \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{down}_{tc}(c,d) \tag{F144}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{down}_{tc}(c,d) \tag{F145}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{left}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})} \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{left}(c,d) \tag{F146}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{left}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})} \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{left}(c,d) \tag{F147}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{left}(c,d) \tag{F148}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}_{tc}(b,d) \tag{F149}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \text{right}(b,d) \tag{F150}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \text{right}(c,a) \tag{F151}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F152}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}(c,d) \tag{F153}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}_{tc}(c,a) \tag{F154}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(a,d) \tag{F155}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Robot})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{robot-at}(c,d) \tag{F156}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \forall_{c:\ (\text{type}(c)\leq^t\text{Robot})} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{robot-at}(c,d) \tag{F157}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F158}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F159}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F160}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F161}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F162}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F163}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F164}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F165}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F166}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}(c,d) \tag{F167}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F168}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F169}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}(c,d) \tag{F170}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}_{tc}(a,d) \tag{F171}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F172}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(b,d) \tag{F173}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F174}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F175}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F176}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{left}_{tc}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F177}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Color})}} \neg\,\text{painted}(a,b) \rightarrow \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Color})\wedge\\ d\neq b}} \text{painted}_{\text{g}}(c,d) \tag{F178}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Color})}} \neg\,\text{painted}(a,b) \rightarrow \exists_{c:\ (\text{type}(c)\leq^t\text{Robot})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Color})\wedge\\ d\neq b}} \text{robot-has}(c,d) \tag{F179}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Color})}} \neg\,\text{painted}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Robot})\\ }} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Color})\wedge\\ d\neq b}} \text{robot-has}(c,d) \tag{F180}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Color})}} \text{painted}_{\text{g}}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Color})\wedge\\ d\neq b}} \text{painted}_{\text{g}}(a,d) \tag{F181}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Color})}} \neg\,\text{painted}_{\text{g}}(a,b) \rightarrow \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Color})\wedge\\ d\neq b}} \text{painted}_{\text{g}}(c,d) \tag{F182}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}(c,d) \tag{F183}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F184}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}(c,d) \tag{F185}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F186}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F187}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F188}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F189}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F190}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \text{left}(c,a) \tag{F191}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\neq a}}{\exists} \text{left}(b,d) \tag{F192}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\neq b}}{\exists} \text{left}(c,a) \tag{F193}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \text{left}(c,d) \tag{F194}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\exists} \text{left}_{tc}(a,d) \tag{F195}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \text{left}_{tc}(c,d) \tag{F196}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{d:\ (\text{type}(d)\leq^t\text{Color})}{\exists} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\forall} \text{painted}_{\text{g}}(c,d) \tag{F197}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{d:\ (\text{type}(d)\leq^t\text{Tile})}{\exists} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\forall} \text{right}(c,d) \tag{F198}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{c:\ (\text{type}(c)\leq^t\text{Tile})}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \text{right}(c,d) \tag{F199}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \text{right}(c,d) \tag{F200}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\exists} \text{right}_{tc}(b,d) \tag{F201}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{c:\ (\text{type}(c)\leq^t\text{Robot})}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \text{robot-at}(c,d) \tag{F202}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{c:\ (\text{type}(c)\leq^t\text{Robot})}{\forall} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\exists} \text{robot-at}(c,d) \tag{F203}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\exists} \text{up}(c,d) \tag{F204}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{d:\ (\text{type}(d)\leq^t\text{Tile})}{\exists} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\forall} \text{up}(c,d) \tag{F205}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{c:\ (\text{type}(c)\leq^t\text{Tile})}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \text{up}(c,d) \tag{F206}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \text{up}(c,d) \tag{F207}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\exists} \text{up}_{tc}(c,d) \tag{F208}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{d:\ (\text{type}(d)\leq^t\text{Tile})}{\exists} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\forall} \text{up}_{tc}(c,d) \tag{F209}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{c:\ (\text{type}(c)\leq^t\text{Tile})}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \text{up}_{tc}(c,d) \tag{F210}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{right}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \text{up}_{tc}(c,d) \tag{F211}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F212}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}(c,d) \tag{F213}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F214}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F215}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}(b,d) \tag{F216}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}(c,d) \tag{F217}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F218}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F219}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}(a,d) \tag{F220}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F221}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F222}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F223}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F224}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F225}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F226}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F227}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}(c,d) \tag{F228}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F229}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}(c,d) \tag{F230}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F231}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F232}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F233}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F234}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F235}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \text{left}(b,d) \tag{F236}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \text{left}(c,a) \tag{F237}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}(c,d) \tag{F238}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}(c,d) \tag{F239}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}(c,d) \tag{F240}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}_{tc}(c,a) \tag{F241}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}_{tc}(a,d) \tag{F242}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F243}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F244}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}(c,d) \tag{F245}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F246}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(b,d) \tag{F247}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F248}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Robot})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{robot-at}(c,d) \tag{F249}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \forall_{c:\ (\text{type}(c)\leq^t\text{Robot})} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{robot-at}(c,d) \tag{F250}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F251}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F252}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F253}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F254}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F255}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F256}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F257}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F258}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F259}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}(c,d) \tag{F260}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F261}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F262}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}(c,d) \tag{F263}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{left}_{tc}(b,d) \tag{F264}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F265}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F266}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(a,d) \tag{F267}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F268}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F269}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F270}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F271}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{right}_{tc}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(c,d) \tag{F272}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Robot})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{robot-at}(a,b) \rightarrow \neg\,\text{clear}(b) \tag{F273}$$

$$\forall_{\substack{a,b:\\ (\text{type}(a)\leq^t\text{Robot})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{robot-at}(a,b) \rightarrow \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \text{left}_{tc}(b,d) \tag{F274}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Robot})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{robot-at}(a,b) \rightarrow \neg \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Robot})\wedge\\ c\neq a}}{\exists} \text{robot-at}(c,b) \tag{F275}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Robot})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{robot-at}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\neq b}}{\exists} \text{robot-at}(a,d) \tag{F276}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Robot})\wedge(\text{type}(b)\leq^t\text{Color})}}{\forall} \text{robot-has}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Color})\wedge\\ d\neq b}}{\exists} \text{robot-has}(a,d) \tag{F277}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Robot})\wedge(\text{type}(b)\leq^t\text{Color})}}{\forall} \neg\,\text{robot-has}(a,b) \rightarrow \underset{c:\ (\text{type}(c)\leq^t\text{Tile})}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Color})\wedge\\ d\neq b}}{\forall} \text{painted}_{\text{g}}(c,d) \tag{F278}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Robot})\wedge(\text{type}(b)\leq^t\text{Color})}}{\forall} \neg\,\text{robot-has}(a,b) \rightarrow \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Color})\wedge\\ d\neq b}}{\forall} \text{robot-has}(a,d) \tag{F279}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Robot})\wedge(\text{type}(b)\leq^t\text{Color})}}{\forall} \neg\,\text{robot-has}(a,b) \rightarrow \neg \underset{c:\ (\text{type}(c)\leq^t\text{Robot})}{\forall} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Color})\wedge\\ d\neq b}}{\exists} \text{robot-has}(c,d) \tag{F280}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Robot})\wedge(\text{type}(b)\leq^t\text{Color})}}{\forall} \neg\,\text{robot-has}_{\text{g}}(a,b) \rightarrow \underset{c:\ (\text{type}(c)\leq^t\text{Tile})}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Color})\wedge\\ d\neq b}}{\forall} \text{painted}_{\text{g}}(c,d) \tag{F281}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \text{clear}(c) \tag{F282}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \text{down}(c,a) \tag{F283}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\neq a}}{\exists} \text{down}(b,d) \tag{F284}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\neq b}}{\exists} \text{down}(c,a) \tag{F285}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \text{down}(c,d) \tag{F286}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\exists} \text{down}_{tc}(a,d) \tag{F287}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \text{down}_{tc}(c,d) \tag{F288}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\exists} \text{left}(c,d) \tag{F289}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{d:\ (\text{type}(d)\leq^t\text{Tile})}{\exists} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\forall} \text{left}(c,d) \tag{F290}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{c:\ (\text{type}(c)\leq^t\text{Tile})}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \text{left}(c,d) \tag{F291}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \text{left}(c,d) \tag{F292}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\exists} \text{left}_{tc}(c,d) \tag{F293}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{d:\ (\text{type}(d)\leq^t\text{Tile})}{\exists} \underset{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}}{\forall} \text{left}_{tc}(c,d) \tag{F294}$$

$$\underset{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}}{\forall} \text{up}(a,b) \rightarrow \neg \underset{c:\ (\text{type}(c)\leq^t\text{Tile})}{\exists} \underset{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}}{\forall} \text{left}_{tc}(c,d) \tag{F295}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{left}_{tc}(c,d) \tag{F296}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \exists_{d:\ (\text{type}(d)\leq^t\text{Color})} \text{painted}_{\text{g}}(a,d) \tag{F297}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Color})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{painted}_{\text{g}}(c,d) \tag{F298}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}(c,d) \tag{F299}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F300}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}(c,d) \tag{F301}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}(c,d) \tag{F302}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F303}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F304}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F305}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{right}_{tc}(c,d) \tag{F306}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Robot})\\}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{robot-at}(c,d) \tag{F307}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{c:\ (\text{type}(c)\leq^t\text{Robot})} \text{robot-at}(c,d) \tag{F308}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t\text{Tile})} \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F309}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t\text{Tile})} \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}(c,d) \tag{F310}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{up}(c,d) \tag{F311}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \text{up}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{up}_{tc}(b,d) \tag{F312}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{up}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}(b,d) \tag{F313}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}(c,d) \tag{F314}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F315}$$

$$\forall_{\substack{a,b:\\(\text{type}(a)\leq^t\text{Tile})\wedge(\text{type}(b)\leq^t\text{Tile})}} \neg\,\text{up}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\text{type}(c)\leq^t\text{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\text{type}(d)\leq^t\text{Tile})\wedge\\ d\in\{a,b\}}} \text{down}_{tc}(c,d) \tag{F316}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{left}(c,d) \tag{F317}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{left}(c,d) \tag{F318}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{left}_{tc}(c,d) \tag{F319}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{left}_{tc}(c,d) \tag{F320}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{right}(c,d) \tag{F321}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{right}(c,d) \tag{F322}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{right}_{tc}(c,d) \tag{F323}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{right}_{tc}(c,d) \tag{F324}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{up}(a,d) \tag{F325}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{up}(c,d) \tag{F326}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{up}_{tc}(c,d) \tag{F327}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \neg\,\mathrm{up}(a,b) \rightarrow \neg \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{up}_{tc}(c,d) \tag{F328}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{up}_{tc}(a,b) \rightarrow \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{clear}(c) \tag{F329}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{up}_{tc}(a,b) \rightarrow \exists_{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})} \mathrm{down}(b,d) \tag{F330}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{up}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})} \forall_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{down}(c,d) \tag{F331}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{up}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})} \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{down}(c,d) \tag{F332}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{up}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{down}(c,d) \tag{F333}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{up}_{tc}(a,b) \rightarrow \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{down}_{tc}(c,a) \tag{F334}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{up}_{tc}(a,b) \rightarrow \neg \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{down}_{tc}(a,d) \tag{F335}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{up}_{tc}(a,b) \rightarrow \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{down}_{tc}(c,d) \tag{F336}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq^t \mathrm{Tile})\wedge(\mathrm{type}(b)\leq^t \mathrm{Tile})}} \mathrm{up}_{tc}(a,b) \rightarrow \neg \exists_{\substack{c:\ (\mathrm{type}(c)\leq^t \mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq^t \mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{left}(c,d) \tag{F337}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t \text{Tile})} \forall_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \text{left}(c,d) \tag{F338}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})} \forall_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \text{left}(c,d) \tag{F339}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \text{left}(c,d) \tag{F340}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \exists_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \text{left}_{tc}(c,d) \tag{F341}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t \text{Tile})} \forall_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \text{left}_{tc}(c,d) \tag{F342}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})} \forall_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \text{left}_{tc}(c,d) \tag{F343}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \text{left}_{tc}(c,d) \tag{F344}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \exists_{d:\ (\text{type}(d)\leq^t \text{Color})} \text{painted}_{\text{g}}(a,d) \tag{F345}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \exists_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \text{right}(c,d) \tag{F346}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t \text{Tile})} \forall_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \text{right}(c,d) \tag{F347}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})} \forall_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \text{right}(c,d) \tag{F348}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \text{right}(c,d) \tag{F349}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \exists_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \text{right}_{tc}(c,d) \tag{F350}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t \text{Tile})} \forall_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \text{right}_{tc}(c,d) \tag{F351}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})} \forall_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \text{right}_{tc}(c,d) \tag{F352}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \text{right}_{tc}(c,d) \tag{F353}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \forall_{c:\ (\text{type}(c)\leq^t \text{Robot})} \exists_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \text{robot-at}(c,d) \tag{F354}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\text{type}(c)\leq^t \text{Robot})} \text{robot-at}(c,d) \tag{F355}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{d:\ (\text{type}(d)\leq^t \text{Tile})} \forall_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \text{up}(c,d) \tag{F356}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})} \forall_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \text{up}(c,d) \tag{F357}$$

$$\forall_{a,b:\ (\text{type}(a)\leq^t \text{Tile})\wedge(\text{type}(b)\leq^t \text{Tile})} \quad \text{up}_{tc}(a,b) \rightarrow \neg \forall_{d:\ (\text{type}(d)\leq^t \text{Tile})\wedge d\in\{a,b\}} \exists_{c:\ (\text{type}(c)\leq^t \text{Tile})\wedge c\in\{a,b\}} \text{up}(c,d) \tag{F358}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})}} \mathrm{up}_{tc}(a,b) \to \neg \exists_{\substack{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{up}_{tc}(b,d) \tag{F359}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})}} \mathrm{up}_{tc}(a,b) \to \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{up}_{tc}(c,d) \tag{F360}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})}} \neg\,\mathrm{up}_{tc}(a,b) \to \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{down}(c,d) \tag{F361}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})}} \neg\,\mathrm{up}_{tc}(a,b) \to \neg \exists_{\substack{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{down}_{tc}(b,d) \tag{F362}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})}} \neg\,\mathrm{up}_{tc}(a,b) \to \neg \forall_{\substack{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge\\ d\in\{a,b\}}} \exists_{\substack{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge\\ c\in\{a,b\}}} \mathrm{down}_{tc}(c,d) \tag{F363}$$

$$\forall_{\substack{a,b:\\ (\mathrm{type}(a)\leq {}^t\mathrm{Tile})\wedge(\mathrm{type}(b)\leq {}^t\mathrm{Tile})}} \neg\,\mathrm{up}_{tc}(a,b) \to \neg \forall_{\substack{c:\ (\mathrm{type}(c)\leq {}^t\mathrm{Tile})\wedge\\ c\in\{a,b\}}} \exists_{\substack{d:\ (\mathrm{type}(d)\leq {}^t\mathrm{Tile})\wedge\\ d\in\{a,b\}}} \mathrm{left}(c,d) \tag{F364}$$

All reduced Floortile constraints.

Grid

$$\text{at}_{tc}(x\colon obj,\ y\colon obj) \coloneqq \text{at}(x,y) \lor \exists_{z\colon obj}(\text{at}(x,z) \land \text{at}_{tc}(z,y))$$
$$\text{conn}_{tc}(x\colon obj,\ y\colon obj) \coloneqq \text{conn}(x,y) \lor \exists_{z\colon obj}(\text{conn}(x,z) \land \text{conn}_{tc}(z,y))$$
$$\text{key-shape}_{tc}(x\colon obj,\ y\colon obj) \coloneqq \text{key-shape}(x,y) \lor \exists_{z\colon obj}(\text{key-shape}(x,z) \land \text{key-shape}_{tc}(z,y))$$
$$\text{lock-shape}_{tc}(x\colon obj,\ y\colon obj) \coloneqq \text{lock-shape}(x,y) \lor \exists_{z\colon obj}(\text{lock-shape}(x,z) \land \text{lock-shape}_{tc}(z,y))$$

$$\top \to \text{arm-empty}() \tag{G1}$$
$$\top \to \neg\,\text{arm-empty}_{\text{g}}() \tag{G2}$$
$$\top \to \exists_{a}\,\text{at-robot}(a) \tag{G3}$$
$$\top \to \neg\exists_{a}\,\text{at-robot}_{\text{g}}(a) \tag{G4}$$
$$\top \to \neg\exists_{a}\,\text{holding}(a) \tag{G5}$$
$$\top \to \neg\exists_{a}\,\text{holding}_{\text{g}}(a) \tag{G6}$$
$$\top \to \neg\exists_{a}\,\text{key}_{\text{g}}(a) \tag{G7}$$
$$\top \to \neg\exists_{a}\,\text{locked}_{\text{g}}(a) \tag{G8}$$
$$\top \to \neg\exists_{a}\,\text{open}_{\text{g}}(a) \tag{G9}$$
$$\top \to \neg\exists_{a}\,\text{place}_{\text{g}}(a) \tag{G10}$$
$$\top \to \neg\exists_{a}\,\text{shape}_{\text{g}}(a) \tag{G11}$$
$$\top \to \exists_{a}\exists_{b}\,\text{key-shape}_{tc}(a,b) \tag{G12}$$
$$\top \to \exists_{a}\exists_{b}\,\text{lock-shape}_{tc}(a,b) \tag{G13}$$
$$\forall_{a}\ \text{at-robot}(a) \to \neg \exists_{b\colon b\neq a}\,\text{at-robot}(b) \tag{G14}$$
$$\forall_{a}\ \text{at-robot}(a) \to \neg\exists_{b}\,\text{at}_{\text{g}}(b,a) \tag{G15}$$
$$\forall_{a}\ \text{key}(a) \to \exists_{c}\,\text{key-shape}_{tc}(a,c) \tag{G16}$$
$$\forall_{a}\neg\,\text{key}(a) \to \neg\exists_{c}\,\text{at}_{\text{g}}(a,c) \tag{G17}$$
$$\forall_{a}\ \text{locked}(a) \to \exists_{c}\,\text{lock-shape}_{tc}(a,c) \tag{G18}$$
$$\forall_{a}\neg\,\text{open}(a) \to \neg\,\text{at-robot}(a) \tag{G19}$$
$$\forall_{a}\ \text{place}(a) \to \neg\exists_{c}\,\text{at}_{\text{g}}(a,c) \tag{G20}$$
$$\forall_{a}\ \text{place}(a) \to \exists_{b}\,\text{conn}(b,a) \tag{G21}$$
$$\forall_{a}\neg\,\text{place}(a) \to \neg\exists_{b}\,\text{at}_{\text{g}}(b,a) \tag{G22}$$
$$\forall_{a}\ \text{shape}(a) \to \neg\exists_{c}\,\text{at}_{\text{g}}(a,c) \tag{G23}$$
$$\forall_{a}\ \text{shape}(a) \to \neg\exists_{b}\,\text{at}_{\text{g}}(b,a) \tag{G24}$$
$$\forall_{a}\ \text{shape}(a) \to \exists_{b}\,\text{key-shape}_{tc}(b,a) \tag{G25}$$
$$\forall_{a,b}\ \text{at}(a,b) \to \neg \exists_{c\colon c\in\{a,b\}}\,\text{shape}(c) \tag{G26}$$
$$\forall_{a,b}\ \text{at}(a,b) \to \neg \exists_{c\colon c\in\{a,b\}}\exists_{d\colon d\neq b}\,\text{at}(c,d) \tag{G27}$$
$$\forall_{a,b}\ \text{at}(a,b) \to \neg\exists_{d}\,\text{at}_{\text{g}}(b,d) \tag{G28}$$

$$\underset{a,b}{\forall}\ \mathrm{at}(a,b) \rightarrow \neg \underset{c}{\exists}\, \mathrm{at_g}(c,a) \tag{G29}$$

$$\underset{a,b}{\forall}\ \mathrm{at}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\exists}\ \underset{d:\ d\neq b}{\exists}\, \mathrm{at}_{tc}(c,d) \tag{G30}$$

$$\underset{a,b}{\forall}\ \mathrm{at}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\exists}\ \underset{d:\ d\in\{a,b\}}{\exists}\, \mathrm{conn}(c,d) \tag{G31}$$

$$\underset{a,b}{\forall}\ \mathrm{at}(a,b) \rightarrow \neg \underset{c}{\exists}\ \underset{d:\ d\in\{a,b\}}{\forall}\, \mathrm{conn}_{tc}(c,d) \tag{G32}$$

$$\underset{a,b}{\forall} \neg\, \mathrm{at}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall}\ \underset{c:\ c\in\{a,b\}}{\exists}\, \mathrm{at}(c,d) \tag{G33}$$

$$\underset{a,b}{\forall} \neg\, \mathrm{at}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall}\ \underset{c:\ c\in\{a,b\}}{\exists}\, \mathrm{at_g}(c,d) \tag{G34}$$

$$\underset{a,b}{\forall} \neg\, \mathrm{at}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\forall}\ \underset{d:\ d\in\{a,b\}}{\exists}\, \mathrm{at_g}(c,d) \tag{G35}$$

$$\underset{a,b}{\forall} \neg\, \mathrm{at}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\exists}\, \mathrm{at}_{tc}(a,d) \tag{G36}$$

$$\underset{a,b}{\forall}\ \mathrm{at_g}(a,b) \rightarrow\ \underset{c}{\exists}\ \underset{d:\ d\neq b}{\exists}\, \mathrm{at_g}(c,d) \tag{G37}$$

$$\underset{a,b}{\forall}\ \mathrm{at_g}(a,b) \rightarrow\ \underset{c:\ c\neq b}{\exists}\ \underset{d}{\exists}\, \mathrm{at_g}(c,d) \tag{G38}$$

$$\underset{a,b}{\forall}\ \mathrm{at_g}(a,b) \rightarrow\ \underset{c:\ c\neq a}{\exists}\ \underset{d}{\exists}\, \mathrm{at_g}(c,d) \tag{G39}$$

$$\underset{a,b}{\forall}\ \mathrm{at_g}(a,b) \rightarrow \neg \underset{c}{\exists}\, \mathrm{at_g}(c,a) \tag{G40}$$

$$\underset{a,b}{\forall}\ \mathrm{at_g}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\exists}\ \underset{d:\ d\neq b}{\exists}\, \mathrm{at_g}(c,d) \tag{G41}$$

$$\underset{a,b}{\forall}\ \mathrm{at_g}(a,b) \rightarrow \neg \underset{d}{\exists}\ \underset{c:\ c\in\{a,b\}}{\forall}\, \mathrm{at_g}(c,d) \tag{G42}$$

$$\underset{a,b}{\forall} \neg\, \mathrm{at_g}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\exists}\, \mathrm{at_g}(a,d) \tag{G43}$$

$$\underset{a,b}{\forall}\ \mathrm{at}_{tc}(a,b) \rightarrow \neg\, \mathrm{open}(a) \tag{G44}$$

$$\underset{a,b}{\forall}\ \mathrm{at}_{tc}(a,b) \rightarrow\ \mathrm{place}(b) \tag{G45}$$

$$\underset{a,b}{\forall}\ \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\exists}\, \mathrm{shape}(c) \tag{G46}$$

$$\underset{a,b}{\forall}\ \mathrm{at}_{tc}(a,b) \rightarrow\ \underset{c:\ c\in\{a,b\}}{\exists}\, \mathrm{at}(c,b) \tag{G47}$$

$$\underset{a,b}{\forall}\ \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\exists}\ \underset{d:\ d\neq b}{\exists}\, \mathrm{at}(c,d) \tag{G48}$$

$$\underset{a,b}{\forall}\ \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{c}{\exists}\, \mathrm{at_g}(c,a) \tag{G49}$$

$$\underset{a,b}{\forall}\ \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\exists}\ \underset{d:\ d\neq b}{\exists}\, \mathrm{at}_{tc}(c,d) \tag{G50}$$

$$\underset{a,b}{\forall}\ \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\exists}\ \underset{d:\ d\in\{a,b\}}{\exists}\, \mathrm{conn}(c,d) \tag{G51}$$

$$\underset{a,b}{\forall}\ \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{d}{\exists}\ \underset{c:\ c\in\{a,b\}}{\forall}\, \mathrm{conn}_{tc}(c,d) \tag{G52}$$

$$\underset{a,b}{\forall}\ \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{c}{\exists}\ \underset{d:\ d\in\{a,b\}}{\forall}\, \mathrm{conn}_{tc}(c,d) \tag{G53}$$

$$\underset{a,b}{\forall}\ \mathrm{at}_{tc}(a,b) \rightarrow\ \underset{d}{\exists}\, \mathrm{key\text{-}shape}_{tc}(a,d) \tag{G54}$$

$$\underset{a,b}{\forall} \neg\, \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall}\ \underset{c:\ c\in\{a,b\}}{\exists}\, \mathrm{at}(c,d) \tag{G55}$$

$$\underset{a,b}{\forall} \neg\, \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\forall}\ \underset{d:\ d\in\{a,b\}}{\exists}\, \mathrm{at_g}(c,d) \tag{G56}$$

$$\underset{a,b}{\forall} \neg\, \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\exists}\, \mathrm{at}_{tc}(a,d) \tag{G57}$$

$$\underset{a,b}{\forall} \neg\, \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall}\ \underset{c:\ c\in\{a,b\}}{\exists}\, \mathrm{at}_{tc}(c,d) \tag{G58}$$

$$\underset{a,b}{\forall} \neg\, \mathrm{at}_{tc}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall}\ \underset{c:\ c\in\{a,b\}}{\exists}\, \mathrm{key\text{-}shape}(c,d) \tag{G59}$$

$$\forall_{a,b} \neg \mathrm{at}_{tc}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \text{key-shape}(c,d) \tag{G60}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \mathrm{key}(c) \tag{G61}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \forall_{c:\, c\in\{a,b\}} \mathrm{place}(c) \tag{G62}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \mathrm{shape}(c) \tag{G63}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d} \mathrm{at}(c,d) \tag{G64}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \exists_{d} \forall_{c:\, c\in\{a,b\}} \mathrm{at}(c,d) \tag{G65}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \exists_{c} \forall_{d:\, d\in\{a,b\}} \mathrm{at}(c,d) \tag{G66}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{at}(c,d) \tag{G67}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d} \mathrm{at}_{\mathrm{g}}(c,d) \tag{G68}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \exists_{d} \forall_{c:\, c\in\{a,b\}} \mathrm{at}_{\mathrm{g}}(c,d) \tag{G69}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \exists_{c} \forall_{d:\, d\in\{a,b\}} \mathrm{at}_{\mathrm{g}}(c,d) \tag{G70}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{at}_{\mathrm{g}}(c,d) \tag{G71}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \exists_{c} \forall_{d:\, d\in\{a,b\}} \mathrm{at}_{tc}(c,d) \tag{G72}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{at}_{tc}(c,d) \tag{G73}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \ \mathrm{conn}(b,a) \tag{G74}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \forall_{c:\, c\in\{a,b\}} \exists_{d} \mathrm{conn}(c,d) \tag{G75}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \forall_{d:\, d\in\{a,b\}} \exists_{c} \mathrm{conn}(c,d) \tag{G76}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{conn}(c,d) \tag{G77}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \forall_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{conn}(c,d) \tag{G78}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \exists_{d} \forall_{c:\, c\in\{a,b\}} \mathrm{conn}(c,d) \tag{G79}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \neg \exists_{c} \forall_{d:\, d\in\{a,b\}} \mathrm{conn}(c,d) \tag{G80}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \forall_{c:\, c\in\{a,b\}} \forall_{d:\, d\in\{a,b\}} \mathrm{conn}_{tc}(c,d) \tag{G81}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \forall_{d:\, d\in\{a,b\}} \exists_{c} \mathrm{conn}_{tc}(c,d) \tag{G82}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{conn}_{tc}(c,d) \tag{G83}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \exists_{d} \forall_{c:\, c\in\{a,b\}} \mathrm{conn}_{tc}(c,d) \tag{G84}$$

$$\forall_{a,b} \ \mathrm{conn}(a,b) \rightarrow \exists_{d:\, d\in\{a,b\}} \forall_{c:\, c\in\{a,b\}} \mathrm{conn}_{tc}(c,d) \tag{G85}$$

$$\forall_{a,b} \neg \mathrm{conn}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{at}(c,d) \tag{G86}$$

$$\forall_{a,b} \neg \mathrm{conn}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{at}(c,d) \tag{G87}$$

$$\forall_{a,b} \neg \mathrm{conn}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{at}_{\mathrm{g}}(c,d) \tag{G88}$$

$$\forall_{a,b} \neg \mathrm{conn}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{at}_{tc}(c,d) \tag{G89}$$

$$\forall_{a,b} \neg \mathrm{conn}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{at}_{tc}(c,d) \tag{G90}$$

$$\forall_{a,b} \neg \mathrm{conn}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{conn}(c,d) \tag{G91}$$

$$\forall_{a,b} \neg \mathrm{conn}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{conn}(c,d) \tag{G92}$$

$$\forall_{a,b} \neg \mathrm{conn}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \text{lock-shape}_{tc}(c,d) \tag{G93}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \mathrm{key}(c) \tag{G94}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \forall_{c:\, c\in\{a,b\}} \mathrm{place}(c) \tag{G95}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \mathrm{shape}(c) \tag{G96}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d} \mathrm{at}(c,d) \tag{G97}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \neg \exists_{d} \forall_{c:\, c\in\{a,b\}} \mathrm{at}(c,d) \tag{G98}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{at}(c,d) \tag{G99}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d} \mathrm{at}_{\mathrm{g}}(c,d) \tag{G100}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \neg \exists_{d} \forall_{c:\, c\in\{a,b\}} \mathrm{at}_{\mathrm{g}}(c,d) \tag{G101}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{at}_{\mathrm{g}}(c,d) \tag{G102}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \forall_{c:\, c\in\{a,b\}} \exists_{d} \mathrm{conn}(c,d) \tag{G103}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \forall_{d:\, d\in\{a,b\}} \exists_{c} \mathrm{conn}(c,d) \tag{G104}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \forall_{c:\, c\in\{a,b\}} \forall_{d:\, d\in\{a,b\}} \mathrm{conn}_{tc}(c,d) \tag{G105}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \forall_{d:\, d\in\{a,b\}} \exists_{c} \mathrm{conn}_{tc}(c,d) \tag{G106}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{conn}_{tc}(c,d) \tag{G107}$$

$$\forall_{a,b} \ \mathrm{conn}_{tc}(a,b) \rightarrow \exists_{d:\, d\in\{a,b\}} \forall_{c:\, c\in\{a,b\}} \mathrm{conn}_{tc}(c,d) \tag{G108}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \mathrm{locked}(c) \tag{G109}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \mathrm{open}(c) \tag{G110}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c} \mathrm{at}(c,d) \tag{G111}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{at}(c,d) \tag{G112}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c} \mathrm{at}_{\mathrm{g}}(c,d) \tag{G113}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{at}_{\mathrm{g}}(c,d) \tag{G114}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c} \mathrm{at}_{tc}(c,d) \tag{G115}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{at}_{tc}(c,d) \tag{G116}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{conn}(c,d) \tag{G117}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \exists_{d} \mathrm{conn}(c,d) \tag{G118}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c} \mathrm{conn}(c,d) \tag{G119}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \exists_{d} \mathrm{conn}_{tc}(c,d) \tag{G120}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c} \mathrm{conn}_{tc}(c,d) \tag{G121}$$

$$\forall_{a,b} \neg \mathrm{conn}_{tc}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{key\text{-}shape}_{tc}(c,d) \tag{G122}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \mathrm{open}(c) \tag{G123}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \mathrm{place}(c) \tag{G124}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}(a,b) \rightarrow \neg\exists_{c} \forall_{d:\, d\in\{a,b\}} \mathrm{at}(c,d) \tag{G125}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}(a,b) \rightarrow \neg\exists_{d} \mathrm{at_g}(b,d) \tag{G126}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}(a,b) \rightarrow \neg \exists_{d:\, d\in\{a,b\}} \exists_{c} \mathrm{at_g}(c,d) \tag{G127}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{at_g}(c,d) \tag{G128}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}(a,b) \rightarrow \neg\exists_{c} \forall_{d:\, d\in\{a,b\}} \mathrm{at_g}(c,d) \tag{G129}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}(a,b) \rightarrow \neg\exists_{d} \forall_{c:\, c\in\{a,b\}} \mathrm{conn}(c,d) \tag{G130}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}(a,b) \rightarrow \neg\exists_{c} \forall_{d:\, d\in\{a,b\}} \mathrm{conn}(c,d) \tag{G131}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d:\, d\neq b} \mathrm{key\text{-}shape}(c,d) \tag{G132}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d:\, d\neq b} \mathrm{key\text{-}shape}_{tc}(c,d) \tag{G133}$$

$$\forall_{a,b} \neg \mathrm{key\text{-}shape}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{at}(c,d) \tag{G134}$$

$$\forall_{a,b} \neg \mathrm{key\text{-}shape}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{at}_{tc}(c,d) \tag{G135}$$

$$\forall_{a,b} \neg \mathrm{key\text{-}shape}(a,b) \rightarrow \neg \forall_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{at}_{tc}(c,d) \tag{G136}$$

$$\forall_{a,b} \neg \mathrm{key\text{-}shape}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{key\text{-}shape}(c,d) \tag{G137}$$

$$\forall_{a,b} \neg \mathrm{key\text{-}shape}(a,b) \rightarrow \neg \exists_{d:\, d\in\{a,b\}} \mathrm{key\text{-}shape}_{tc}(a,d) \tag{G138}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}_{tc}(a,b) \rightarrow \ \mathrm{key}(a) \tag{G139}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}_{tc}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \mathrm{open}(c) \tag{G140}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}_{tc}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \mathrm{place}(c) \tag{G141}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}_{tc}(a,b) \rightarrow \ \mathrm{shape}(b) \tag{G142}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}_{tc}(a,b) \rightarrow \ \exists_{d} \mathrm{at}(a,d) \tag{G143}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}_{tc}(a,b) \rightarrow \neg\exists_{c} \forall_{d:\, d\in\{a,b\}} \mathrm{at}(c,d) \tag{G144}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}_{tc}(a,b) \rightarrow \neg \exists_{d:\, d\in\{a,b\}} \exists_{c} \mathrm{at_g}(c,d) \tag{G145}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}_{tc}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d:\, d\in\{a,b\}} \mathrm{at_g}(c,d) \tag{G146}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}_{tc}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d:\, d\neq b} \mathrm{key\text{-}shape}(c,d) \tag{G147}$$

$$\forall_{a,b} \ \mathrm{key\text{-}shape}_{tc}(a,b) \rightarrow \neg \exists_{c:\, c\in\{a,b\}} \exists_{d:\, d\neq b} \mathrm{key\text{-}shape}_{tc}(c,d) \tag{G148}$$

$$\forall_{a,b} \neg \mathrm{key\text{-}shape}_{tc}(a,b) \rightarrow \neg \forall_{d:\, d\in\{a,b\}} \exists_{c:\, c\in\{a,b\}} \mathrm{at}_{tc}(c,d) \tag{G149}$$

$$\underset{a,b}{\forall} \neg \text{key-shape}_{tc}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\forall} \underset{d:\ d\in\{a,b\}}{\exists} \text{at}_{tc}(c,d) \tag{G150}$$

$$\underset{a,b}{\forall} \neg \text{key-shape}_{tc}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall} \underset{c:\ c\in\{a,b\}}{\exists} \text{key-shape}_{tc}(c,d) \tag{G151}$$

$$\underset{a,b}{\forall} \text{lock-shape}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\exists} \underset{d}{\exists} \text{at}_{\text{g}}(c,d) \tag{G152}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall} \underset{c:\ c\in\{a,b\}}{\exists} \text{at}(c,d) \tag{G153}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\forall} \underset{d:\ d\in\{a,b\}}{\exists} \text{at}(c,d) \tag{G154}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall} \underset{c:\ c\in\{a,b\}}{\exists} \text{at}_{tc}(c,d) \tag{G155}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\forall} \underset{d:\ d\in\{a,b\}}{\exists} \text{at}_{tc}(c,d) \tag{G156}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall} \underset{c:\ c\in\{a,b\}}{\exists} \text{key-shape}_{tc}(c,d) \tag{G157}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\forall} \underset{d:\ d\in\{a,b\}}{\exists} \text{key-shape}_{tc}(c,d) \tag{G158}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\exists} \text{lock-shape}_{tc}(a,d) \tag{G159}$$

$$\underset{a,b}{\forall} \text{lock-shape}_{tc}(a,b) \rightarrow \text{locked}(a) \tag{G160}$$

$$\underset{a,b}{\forall} \text{lock-shape}_{tc}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\exists} \text{open}(c) \tag{G161}$$

$$\underset{a,b}{\forall} \text{lock-shape}_{tc}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\exists} \underset{d}{\exists} \text{at}_{\text{g}}(c,d) \tag{G162}$$

$$\underset{a,b}{\forall} \text{lock-shape}_{tc}(a,b) \rightarrow \underset{c}{\exists} \text{conn}(c,a) \tag{G163}$$

$$\underset{a,b}{\forall} \text{lock-shape}_{tc}(a,b) \rightarrow \underset{c}{\exists} \text{key-shape}(c,b) \tag{G164}$$

$$\underset{a,b}{\forall} \text{lock-shape}_{tc}(a,b) \rightarrow \underset{c:\ c\neq a}{\exists} \text{lock-shape}_{tc}(c,b) \tag{G165}$$

$$\underset{a,b}{\forall} \text{lock-shape}_{tc}(a,b) \rightarrow \neg \underset{c}{\exists} \underset{d:\ d\neq b}{\exists} \text{lock-shape}_{tc}(c,d) \tag{G166}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}_{tc}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall} \underset{c:\ c\in\{a,b\}}{\exists} \text{at}_{tc}(c,d) \tag{G167}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}_{tc}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\forall} \underset{d:\ d\in\{a,b\}}{\exists} \text{at}_{tc}(c,d) \tag{G168}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}_{tc}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall} \underset{c:\ c\in\{a,b\}}{\exists} \text{key-shape}(c,d) \tag{G169}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}_{tc}(a,b) \rightarrow \neg \underset{d:\ d\in\{a,b\}}{\forall} \underset{c:\ c\in\{a,b\}}{\exists} \text{key-shape}_{tc}(c,d) \tag{G170}$$

$$\underset{a,b}{\forall} \neg \text{lock-shape}_{tc}(a,b) \rightarrow \neg \underset{c:\ c\in\{a,b\}}{\forall} \underset{d:\ d\in\{a,b\}}{\exists} \text{key-shape}_{tc}(c,d) \tag{G171}$$

$$\underset{a}{\forall} (\text{type}(a) \leq^t \textsf{Obj}) \rightarrow \neg \underset{b}{\exists} \text{conn}_{\text{g}}(b,a) \tag{G172}$$

$$\underset{a}{\forall} (\text{type}(a) \leq^t \textsf{Obj}) \rightarrow \neg \underset{b}{\exists} \text{key-shape}_{\text{g}}(b,a) \tag{G173}$$

$$\underset{a}{\forall} (\text{type}(a) \leq^t \textsf{Obj}) \rightarrow \neg \underset{b}{\exists} \text{lock-shape}_{\text{g}}(b,a) \tag{G174}$$

All reduced Grid constraints.

Automatically Uncovering Intended Domain Constraints in Automated Planning

Appendix

F Candidate and Learned Constraints — $\mathcal{C}_{\text{LLM}}$

For each constraint in $\mathcal{C}_{\text{LLM}}$, free variables are universally quantified.

Blocksworld

Candidate constraints:

$$\text{above}(x\colon \mathit{block},\ y\colon \mathit{block}) \coloneqq \text{on}(x,y) \vee \exists_{z\colon \mathit{block}}(\text{on}(x,z) \wedge \text{above}(z,y))$$

$$\text{above}_g(x\colon \mathit{block},\ y\colon \mathit{block}) \coloneqq \text{on}_g(x,y) \vee \exists_{z\colon \mathit{block}}(\text{on}_g(x,z) \wedge \text{above}_g(z,y))$$

$$\text{has}_{on\_above}(x\colon \mathit{block}) \coloneqq \exists_{y\colon \mathit{block}}(\text{on}(y,x))$$

$$\text{has}_{on\_above\_g}(x\colon \mathit{block}) \coloneqq \exists_{y\colon \mathit{block}}(\text{on}_g(y,x))$$

$$\text{supported}(x\colon \mathit{block}) \coloneqq \text{ontable}(x) \vee \exists_{y\colon \mathit{block}}(\text{on}(x,y))$$

$$\text{supported}_g(x\colon \mathit{block}) \coloneqq \text{ontable}_g(x) \vee \exists_{y\colon \mathit{block}}(\text{on}_g(x,y))$$

$$\text{top}(x\colon \mathit{block}) \coloneqq \text{clear}(x) \wedge \neg(\exists_{y\colon \mathit{block}}(\text{on}(y,x)))$$

$$\text{top}_g(x\colon \mathit{block}) \coloneqq \text{clear}_g(x) \wedge \neg(\exists_{y\colon \mathit{block}}(\text{on}_g(y,x)))$$

$$(\exists_{x\colon \mathit{block}}(\text{holding}(x)) \rightarrow \neg(\text{handempty}())) \tag{B-LLM-C1}$$

$$(\exists_{x\colon \mathit{block}}(\text{holding}_g(x)) \rightarrow \neg(\text{handempty}_g())) \tag{B-LLM-C2}$$

$$(\neg(\text{handempty}()) \rightarrow \exists_{x\colon \mathit{block}}(\text{holding}(x))) \tag{B-LLM-C3}$$

$$\exists_{x\colon \mathit{block}}(\exists_{y\colon \mathit{block}}(\text{on}(x,y))) \tag{B-LLM-C4}$$

$$\exists_{x\colon \mathit{block}}(\exists_{y\colon \mathit{block}}(\text{on}_g(x,y))) \tag{B-LLM-C5}$$

$$\exists_{x\colon \mathit{block}}(\neg(\text{holding}(x))) \tag{B-LLM-C6}$$

$$\exists_{x\colon \mathit{block}}(\text{supported}(x) \vee \text{holding}(x)) \tag{B-LLM-C7}$$

$$\exists_{x\colon \mathit{block}}(\text{clear}(x)) \tag{B-LLM-C8}$$

$$\exists_{x\colon \mathit{block}}(\text{ontable}(x)) \tag{B-LLM-C9}$$

$$\exists_{x\colon \mathit{block}}(\text{supported}(x)) \tag{B-LLM-C10}$$

$$\exists_{x\colon \mathit{block}}(\text{top}(x)) \tag{B-LLM-C11}$$

$$\forall_{x\colon \mathit{block}}((\neg(\text{holding}(x)) \wedge \neg(\exists_{y\colon \mathit{block}}(\text{on}(y,x))) \rightarrow \text{clear}(x))) \tag{B-LLM-C12}$$

$$\forall_{x\colon \mathit{block}}((\neg(\text{holding}(x)) \rightarrow \text{ontable}(x) \vee \exists_{y\colon \mathit{block}}(\text{on}(x,y)))) \tag{B-LLM-C13}$$

$$\forall_{x\colon \mathit{block}}((\text{ontable}(x) \vee \exists_{y\colon \mathit{block}}(\text{on}(x,y)) \rightarrow \text{supported}(x))) \tag{B-LLM-C14}$$

$$\forall_{x\colon \mathit{block}}((\text{ontable}_g(x) \vee \exists_{y\colon \mathit{block}}(\text{on}_g(x,y)) \rightarrow \text{supported}_g(x))) \tag{B-LLM-C15}$$

$$\forall_{x\colon \mathit{block}}(\forall_{y\colon \mathit{block}}((\text{above}(x,y) \wedge \text{above}(y,x) \rightarrow \neg(x \neq y)))) \tag{B-LLM-C16}$$

$$\forall_{x\colon \mathit{block}}(\forall_{y\colon \mathit{block}}((\text{above}_g(x,y) \wedge \text{above}_g(y,x) \rightarrow \neg(x \neq y)))) \tag{B-LLM-C17}$$

$$\forall_{x\colon \mathit{block}}(\forall_{y\colon \mathit{block}}((\text{holding}(x) \wedge x \neq y \rightarrow \neg(\text{holding}(y))))) \tag{B-LLM-C18}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,(\,\forall_{z:\,block}\,((\mathrm{on}(x,y)\wedge\mathrm{on}(x,z)\rightarrow\neg(y\neq z)))))\tag{B-LLM-C19}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,(\,\forall_{z:\,block}\,((\mathrm{on}(x,y)\wedge\mathrm{on}(y,z)\rightarrow\mathrm{above}(x,z)))))\tag{B-LLM-C20}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,(\,\forall_{z:\,block}\,((\mathrm{on}_g(x,y)\wedge\mathrm{on}_g(x,z)\rightarrow\neg(y\neq z)))))\tag{B-LLM-C21}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,(\neg(\mathrm{holding}(x)\wedge\mathrm{on}(x,y))))\tag{B-LLM-C22}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,(\neg(\mathrm{holding}(x)\wedge\mathrm{on}(y,x))))\tag{B-LLM-C23}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,(\neg(\mathrm{holding}_g(x)\wedge\mathrm{on}_g(x,y))))\tag{B-LLM-C24}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,(\neg(\mathrm{on}(x,y)\wedge\mathrm{on}(y,x))))\tag{B-LLM-C25}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,(\neg(\mathrm{on}(y,x)\wedge\mathrm{clear}(x))))\tag{B-LLM-C26}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,(\neg(\mathrm{on}_g(x,y)\wedge\mathrm{on}_g(y,x))))\tag{B-LLM-C27}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{above}(x,y)\rightarrow\neg(\mathrm{clear}(y)))))\tag{B-LLM-C28}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{above}(x,y)\rightarrow\neg(\mathrm{holding}(y)))))\tag{B-LLM-C29}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{above}(x,y)\rightarrow\neg(\mathrm{ontable}(x)))))\tag{B-LLM-C30}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{above}(x,y)\rightarrow x\neq y)))\tag{B-LLM-C31}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{above}(x,y)\rightarrow\mathrm{supported}(x))))\tag{B-LLM-C32}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{above}_g(x,y)\rightarrow\neg(\mathrm{clear}_g(y)))))\tag{B-LLM-C33}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{above}_g(x,y)\rightarrow x\neq y)))\tag{B-LLM-C34}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{holding}(x)\rightarrow\neg(\mathrm{on}(x,y)))))\tag{B-LLM-C35}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{holding}(x)\rightarrow\neg(\mathrm{on}(y,x)))))\tag{B-LLM-C36}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{holding}_g(x)\rightarrow\neg(\mathrm{on}_g(x,y)))))\tag{B-LLM-C37}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{holding}_g(x)\rightarrow\neg(\mathrm{on}_g(y,x)))))\tag{B-LLM-C38}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{on}(x,y)\rightarrow\neg(\mathrm{clear}(y)))))\tag{B-LLM-C39}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{on}(x,y)\rightarrow\neg(\mathrm{holding}(x)))))\tag{B-LLM-C40}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{on}(x,y)\rightarrow\neg(\mathrm{ontable}(x)))))\tag{B-LLM-C41}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{on}(x,y)\rightarrow x\neq y)))\tag{B-LLM-C42}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{on}(x,y)\rightarrow\mathrm{above}(x,y))))\tag{B-LLM-C43}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{on}(y,x)\rightarrow\neg(\mathrm{clear}(x)))))\tag{B-LLM-C44}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{on}(y,x)\rightarrow\neg(\mathrm{holding}(x)))))\tag{B-LLM-C45}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{on}_g(x,y)\rightarrow\neg(\mathrm{clear}_g(y)))))\tag{B-LLM-C46}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{on}_g(x,y)\rightarrow\neg(\mathrm{holding}_g(x)))))\tag{B-LLM-C47}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{on}_g(x,y)\rightarrow\neg(\mathrm{on}_g(y,x)))))\tag{B-LLM-C48}$$

$$\forall_{x:\,block}\,(\,\forall_{y:\,block}\,((\mathrm{on}_g(x,y)\rightarrow\neg(\mathrm{ontable}_g(x)))))\tag{B-LLM-C49}$$

$$\forall_{x:\, block}(\forall_{y:\, block}((\text{on}_g(x,y) \rightarrow x \neq y))) \tag{B-LLM-C50}$$

$$\forall_{x:\, block}(\forall_{y:\, block}((\text{on}_g(x,y) \rightarrow \text{above}_g(x,y)))) \tag{B-LLM-C51}$$

$$\forall_{x:\, block}(\neg(\text{holding}(x) \wedge \text{clear}(x))) \tag{B-LLM-C52}$$

$$\forall_{x:\, block}(\neg(\text{holding}(x) \wedge \text{ontable}(x))) \tag{B-LLM-C53}$$

$$\forall_{x:\, block}(\neg(\text{holding}_g(x) \wedge \text{clear}_g(x))) \tag{B-LLM-C54}$$

$$\forall_{x:\, block}(\neg(\text{holding}_g(x) \wedge \text{ontable}_g(x))) \tag{B-LLM-C55}$$

$$\forall_{x:\, block}(\neg(\text{above}(x,x))) \tag{B-LLM-C56}$$

$$\forall_{x:\, block}(\neg(\text{above}_g(x,x))) \tag{B-LLM-C57}$$

$$\forall_{x:\, block}((\text{clear}(x) \rightarrow \forall_{y:\, block}(\neg(\text{on}(y,x))))) \tag{B-LLM-C58}$$

$$\forall_{x:\, block}((\text{clear}(x) \rightarrow \neg(\exists_{y:\, block}(\text{on}(y,x))))) \tag{B-LLM-C59}$$

$$\forall_{x:\, block}((\text{clear}(x) \rightarrow \neg(\text{holding}(x)))) \tag{B-LLM-C60}$$

$$\forall_{x:\, block}((\text{clear}(x) \rightarrow \text{ontable}(x) \vee \exists_{y:\, block}(\text{on}(x,y)))) \tag{B-LLM-C61}$$

$$\forall_{x:\, block}((\text{clear}(x) \rightarrow \text{top}(x))) \tag{B-LLM-C62}$$

$$\forall_{x:\, block}((\text{clear}_g(x) \rightarrow \neg(\exists_{y:\, block}(\text{on}_g(y,x))))) \tag{B-LLM-C63}$$

$$\forall_{x:\, block}((\text{clear}_g(x) \rightarrow \neg(\text{holding}_g(x)))) \tag{B-LLM-C64}$$

$$\forall_{x:\, block}((\text{clear}_g(x) \rightarrow \text{top}_g(x))) \tag{B-LLM-C65}$$

$$\forall_{x:\, block}((\text{has}_{on\_above}(x) \rightarrow \exists_{y:\, block}(\text{above}(y,x)))) \tag{B-LLM-C66}$$

$$\forall_{x:\, block}((\text{has}_{on\_above}(x) \rightarrow \neg(\text{clear}(x)))) \tag{B-LLM-C67}$$

$$\forall_{x:\, block}((\text{has}_{on\_above}(x) \rightarrow \neg(\text{holding}(x)))) \tag{B-LLM-C68}$$

$$\forall_{x:\, block}((\text{has}_{on\_above}(x) \rightarrow \neg(\text{top}(x)))) \tag{B-LLM-C69}$$

$$\forall_{x:\, block}((\text{has}_{on\_above\_g}(x) \rightarrow \exists_{y:\, block}(\text{above}_g(y,x)))) \tag{B-LLM-C70}$$

$$\forall_{x:\, block}((\text{has}_{on\_above\_g}(x) \rightarrow \neg(\text{clear}_g(x)))) \tag{B-LLM-C71}$$

$$\forall_{x:\, block}((\text{has}_{on\_above\_g}(x) \rightarrow \neg(\text{holding}_g(x)))) \tag{B-LLM-C72}$$

$$\forall_{x:\, block}((\text{holding}(x) \rightarrow \neg(\text{clear}(x)))) \tag{B-LLM-C73}$$

$$\forall_{x:\, block}((\text{holding}(x) \rightarrow \neg(\text{handempty}()))) \tag{B-LLM-C74}$$

$$\forall_{x:\, block}((\text{holding}(x) \rightarrow \neg(\text{ontable}(x)))) \tag{B-LLM-C75}$$

$$\forall_{x:\, block}((\text{holding}(x) \rightarrow \neg(\text{supported}(x)))) \tag{B-LLM-C76}$$

$$\forall_{x:\, block}((\text{holding}_g(x) \rightarrow \neg(\text{handempty}_g()))) \tag{B-LLM-C77}$$

$$\forall_{x:\, block}((\text{holding}_g(x) \rightarrow \neg(\text{ontable}_g(x)))) \tag{B-LLM-C78}$$

$$\forall_{x:\, block}(\text{ontable}(x) \vee \exists_{y:\, block}(\text{on}(x,y)) \vee \text{holding}(x)) \tag{B-LLM-C79}$$

$$\forall_{x:\, block}((\text{ontable}(x) \rightarrow \neg(\text{holding}(x)))) \tag{B-LLM-C80}$$

$$\forall_{x:\, block}((\text{ontable}_g(x) \rightarrow \neg(\text{holding}_g(x)))) \tag{B-LLM-C81}$$

$$\forall_{x:\, block}(\text{supported}(x) \vee \text{holding}(x)) \tag{B-LLM-C82}$$

$$\underset{x:\, block}{\forall}((\text{supported}(x) \rightarrow \text{ontable}(x) \vee \underset{y:\, block}{\exists}(\text{on}(x,y)))) \tag{B-LLM-C83}$$

$$\underset{x:\, block}{\forall}((\text{supported}_g(x) \rightarrow \neg(\text{holding}_g(x)))) \tag{B-LLM-C84}$$

$$\underset{x:\, block}{\forall}((\text{supported}_g(x) \rightarrow \text{ontable}_g(x) \vee \underset{y:\, block}{\exists}(\text{on}_g(x,y)))) \tag{B-LLM-C85}$$

$$\underset{x:\, block}{\forall}((\text{top}(x) \rightarrow \neg(\text{has}_{on\_above}(x)))) \tag{B-LLM-C86}$$

$$\underset{x:\, block}{\forall}((\text{top}(x) \rightarrow \neg(\text{holding}(x)))) \tag{B-LLM-C87}$$

$$\underset{x:\, block}{\forall}((\text{top}(x) \rightarrow \text{clear}(x))) \tag{B-LLM-C88}$$

$$\underset{x:\, block}{\forall}((\text{top}_g(x) \rightarrow \neg(\text{has}_{on\_above\_g}(x)))) \tag{B-LLM-C89}$$

$$\underset{x:\, block}{\forall}((\text{top}_g(x) \rightarrow \neg(\text{holding}_g(x)))) \tag{B-LLM-C90}$$

$$\underset{x:\, block}{\forall}((\text{top}_g(x) \rightarrow \text{clear}_g(x))) \tag{B-LLM-C91}$$

$$\underset{y:\, block}{\forall}(\underset{x:\, block}{\forall}(\underset{z:\, block}{\forall}((\text{on}(x,y) \wedge \text{on}(z,y) \rightarrow \neg(x \neq z))))) \tag{B-LLM-C92}$$

$$\underset{y:\, block}{\forall}(\underset{x:\, block}{\forall}(\underset{z:\, block}{\forall}((\text{on}_g(x,y) \wedge \text{on}_g(z,y) \rightarrow \neg(x \neq z))))) \tag{B-LLM-C93}$$

$$\underset{y:\, block}{\forall}(\underset{x:\, block}{\forall}(\neg(\text{holding}_g(x) \wedge \text{on}_g(y,x)))) \tag{B-LLM-C94}$$

$$\underset{y:\, block}{\forall}(\underset{x:\, block}{\forall}(\neg(\text{on}(x,y) \wedge \text{clear}(y)))) \tag{B-LLM-C95}$$

$$\underset{y:\, block}{\forall}(\underset{x:\, block}{\forall}(\neg(\text{on}_g(x,y) \wedge \text{clear}_g(y)))) \tag{B-LLM-C96}$$

$$\text{handempty}() \vee \underset{x:\, block}{\exists}(\text{holding}(x)) \tag{B-LLM-C97}$$

$$(\text{handempty}() \rightarrow \underset{x:\, block}{\exists}(\text{clear}(x))) \tag{B-LLM-C98}$$

$$(\text{handempty}() \rightarrow \neg(\underset{x:\, block}{\exists}(\text{holding}(x)))) \tag{B-LLM-C99}$$

$$(\text{handempty}_g() \rightarrow \neg(\underset{x:\, block}{\exists}(\text{holding}_g(x)))) \tag{B-LLM-C100}$$

With LLM-generated descriptions:

- B-LLM-C1: If any block is held, then the hand is not empty (initially).
- B-LLM-C2: If any block is held in the goal, then handempty-goal is false.
- B-LLM-C3: If the hand is not empty initially, then some block is being held.
- B-LLM-C4: There is at least one on relation initially.
- B-LLM-C5: At least one on relation appears in the goal (common in benchmark tasks).
- B-LLM-C6: At least one block is not held initially.
- B-LLM-C7: There exists at least one block that is supported or held initially.
- B-LLM-C8: There is at least one clear block initially.
- B-LLM-C9: There is at least one block on the table initially.
- B-LLM-C10: At least one block is supported initially.
- B-LLM-C11: There exists at least one top block initially.
- B-LLM-C12: If x is not held and nothing is on x, then x is clear (initially).
- B-LLM-C13: If a block is not held, it is either on the table or on some other block (initially).
- B-LLM-C14: Definition direction: ontable or on implies supported (initially).
- B-LLM-C15: Definition direction: ontable-goal or on-goal implies supported-goal.
- B-LLM-C16: Above is antisymmetric initially (no two-way above unless same block).
- B-LLM-C17: above-goal is antisymmetric (no mutual-above cycles in goal).
- B-LLM-C18: At most one block can be held initially.
- B-LLM-C19: Uniqueness of support: a block cannot be on two different blocks initially.

- B-LLM-C20: Transitivity helper: if x on y and y on z initially, then x is above z.
- B-LLM-C21: Goal uniqueness of support: a block cannot be on two different blocks in the goal.
- B-LLM-C22: No block is both held and simultaneously on another block initially.
- B-LLM-C23: No block is held while another block is on it (initially).
- B-LLM-C24: No block is both held and on another block in the goal.
- B-LLM-C25: No mutual on relation initially (no two-cycle).
- B-LLM-C26: No block is clear while another block is on it (initially).
- B-LLM-C27: No mutual on-goal relation in the goal.
- B-LLM-C28: If x is above y initially, then y is not clear initially.
- B-LLM-C29: If x is above y initially, then y is not being held.
- B-LLM-C30: If x is above y initially, x is not on the table.
- B-LLM-C31: above implies distinct blocks initially.
- B-LLM-C32: If x is above y initially, x is supported initially.
- B-LLM-C33: If x is above y in the goal, then y is not clear in the goal.
- B-LLM-C34: above-goal implies distinct blocks in the goal.
- B-LLM-C35: A held block cannot be on any support (initially).
- B-LLM-C36: Nothing can be on a block while it is held (initially).
- B-LLM-C37: If a block is held in the goal, it is not on any block in the goal.
- B-LLM-C38: If a block is held in the goal, nothing is on it in the goal.
- B-LLM-C39: If x is on y initially, y is not clear.
- B-LLM-C40: A block on something cannot be held at the same time.
- B-LLM-C41: A block on another block is not on the table.
- B-LLM-C42: No block can be on itself initially.
- B-LLM-C43: Direct on implies transitive above (initially).
- B-LLM-C44: If something is on x, then x is not clear.
- B-LLM-C45: If some y is on x initially, then x is not held.
- B-LLM-C46: If x is on y in the goal, then y is not clear in the goal.
- B-LLM-C47: If x is on y in the goal, x is not held in the goal.
- B-LLM-C48: No 2-cycles in goal on-goal relation.
- B-LLM-C49: If x is on y in the goal, x is not on the table in the goal.
- B-LLM-C50: No block can be on itself in the goal.
- B-LLM-C51: Direct on-goal implies transitive above-goal.
- B-LLM-C52: No block is both held and clear initially.
- B-LLM-C53: No block is both held and on the table initially.
- B-LLM-C54: No block is both held and clear in the goal.
- B-LLM-C55: No block is both held and on the table in the goal.
- B-LLM-C56: The above relation is irreflexive initially (no cycles).
- B-LLM-C57: above-goal is irreflexive (goal towers are acyclic).
- B-LLM-C58: If x is clear initially, then no y is on x.
- B-LLM-C59: If a block is clear initially, nothing is on it.
- B-LLM-C60: Clear blocks are not held (initially).
- B-LLM-C61: If x is clear initially, then x is either on the table or on another block.
- B-LLM-C62: Clear implies top (given no block is on a clear block initially).
- B-LLM-C63: Clear-goal implies nothing is on x in the goal.
- B-LLM-C64: If x is clear in the goal, then x is not held in the goal.
- B-LLM-C65: Clear-goal implies top-goal (given no block is on a clear-goal block).
- B-LLM-C66: If something is on x, then some block is above x initially.
- B-LLM-C67: If something is on x initially, x is not clear.
- B-LLM-C68: If something is on x initially, x is not held.

- B-LLM-C69: If something is on x, then x is not a top block (initially).
- B-LLM-C70: If something is on x in the goal, then some block is above x in the goal.
- B-LLM-C71: If something is on x in the goal, x is not clear in the goal.
- B-LLM-C72: If something is on x in the goal, x is not held in the goal.
- B-LLM-C73: A held block is not clear (initially).
- B-LLM-C74: If some block is held, the hand is not empty (initially).
- B-LLM-C75: A held block is not on the table (initially).
- B-LLM-C76: Held blocks are not supported (initially).
- B-LLM-C77: If any block is held in the goal, the hand is not empty in the goal.
- B-LLM-C78: A held block in the goal is not on the table in the goal.
- B-LLM-C79: Exhaustive placement initially: every block is on the table, on another block, or held.
- B-LLM-C80: A block on the table is not being held (initially).
- B-LLM-C81: If x is on the table in the goal, then x is not held in the goal.
- B-LLM-C82: Every block is either supported or held initially.
- B-LLM-C83: Definition direction: supported implies ontable or on (initially).
- B-LLM-C84: If x is supported in the goal, it is not held in the goal.
- B-LLM-C85: Definition direction: supported-goal implies ontable-goal or on-goal.
- B-LLM-C86: Top blocks have nothing on them (initially).
- B-LLM-C87: Top blocks are not held (initially).
- B-LLM-C88: Top blocks are clear (initially).
- B-LLM-C89: Top-goal blocks have nothing on them in the goal.
- B-LLM-C90: Top-goal blocks are not held in the goal.
- B-LLM-C91: Top-goal blocks are clear in the goal.
- B-LLM-C92: Uniqueness of occupant: at most one block can be on a given block initially.
- B-LLM-C93: Goal uniqueness of occupant: at most one block can be on a given block in the goal.
- B-LLM-C94: No block is held while another block is on it in the goal.
- B-LLM-C95: A block cannot be on a clear block initially.
- B-LLM-C96: A block cannot be on a clear block in the goal.
- B-LLM-C97: Initially, either the hand is empty or some block is held.
- B-LLM-C98: If the hand is empty initially, then some block is clear.
- B-LLM-C99: If the hand is empty, then no block is held (initially).
- B-LLM-C100: If the goal specifies handempty, then no block is held in the goal.

Which gives the reduced set:

$$\exists_{x:\, block}(\exists_{y:\, block}(\text{on}_g(x,y))) \tag{B-LLM-1}$$

$$\exists_{x:\, block}(\text{ontable}(x)) \tag{B-LLM-2}$$

$$\exists_{x:\, block}(\text{top}(x)) \tag{B-LLM-3}$$

$$\forall_{x:\, block}((\neg(\text{holding}(x)) \wedge \neg(\exists_{y:\, block}(\text{on}(y,x))) \rightarrow \text{clear}(x))) \tag{B-LLM-4}$$

$$\forall_{x:\, block}(\forall_{y:\, block}((\text{holding}(x) \wedge x \neq y \rightarrow \neg(\text{holding}(y))))) \tag{B-LLM-5}$$

$$\forall_{x:\, block}(\forall_{y:\, block}(\forall_{z:\, block}((\text{on}(x,y) \wedge \text{on}(x,z) \rightarrow \neg(y \neq z))))) \tag{B-LLM-6}$$

$$\forall_{x:\, block}(\forall_{y:\, block}(\forall_{z:\, block}((\text{on}_g(x,y) \wedge \text{on}_g(x,z) \rightarrow \neg(y \neq z))))) \tag{B-LLM-7}$$

$$\forall_{x:\, block}(\forall_{y:\, block}(\neg(\text{on}(x,y) \wedge \text{on}(y,x)))) \tag{B-LLM-8}$$

$$\forall_{x:\, block}(\forall_{y:\, block}((\text{above}(x,y) \rightarrow \neg(\text{holding}(y))))) \tag{B-LLM-9}$$

$$\forall_{x:\, block}(\forall_{y:\, block}((\text{above}(x,y) \rightarrow \text{supported}(x)))) \tag{B-LLM-10}$$

$$\forall_{x:\, block}(\forall_{y:\, block}((\text{above}_g(x,y) \rightarrow \neg(\text{clear}_g(y))))) \tag{B-LLM-11}$$

$$\forall_{x:\, block}(\forall_{y:\, block}((\text{on}(x,y) \rightarrow \neg(\text{ontable}(x))))) \tag{B-LLM-12}$$

$$\forall_{x:\, block}(\forall_{y:\, block}((\text{on}_g(x,y) \rightarrow \neg(\text{on}_g(y,x))))) \tag{B-LLM-13}$$

$$\forall_{x:\, block}(\forall_{y:\, block}((\text{on}_g(x,y) \rightarrow \neg(\text{ontable}_g(x))))) \tag{B-LLM-14}$$

$$\forall_{x:\, block}((\text{holding}(x) \rightarrow \neg(\text{supported}(x)))) \tag{B-LLM-15}$$

$$\forall_{x:\, block}(\text{supported}(x) \vee \text{holding}(x)) \tag{B-LLM-16}$$

$$\forall_{x:\, block}((\text{supported}_g(x) \rightarrow \neg(\text{holding}_g(x)))) \tag{B-LLM-17}$$

$$\forall_{x:\, block}((\text{top}(x) \rightarrow \neg(\text{holding}(x)))) \tag{B-LLM-18}$$

$$\forall_{x:\, block}((\text{top}_g(x) \rightarrow \neg(\text{holding}_g(x)))) \tag{B-LLM-19}$$

$$\forall_{y:\, block}(\forall_{x:\, block}(\forall_{z:\, block}((\text{on}(x,y) \wedge \text{on}(z,y) \rightarrow \neg(x \neq z))))) \tag{B-LLM-20}$$

$$\forall_{y:\, block}(\forall_{x:\, block}(\forall_{z:\, block}((\text{on}_g(x,y) \wedge \text{on}_g(z,y) \rightarrow \neg(x \neq z))))) \tag{B-LLM-21}$$

$$\forall_{y:\, block}(\forall_{x:\, block}(\neg(\text{holding}_g(x) \wedge \text{on}_g(y,x)))) \tag{B-LLM-22}$$

$$\forall_{y:\, block}(\forall_{x:\, block}(\neg(\text{on}(x,y) \wedge \text{clear}(y)))) \tag{B-LLM-23}$$

$$\text{handempty}() \vee \exists_{x:\, block}(\text{holding}(x)) \tag{B-LLM-24}$$

$$(\text{handempty}() \rightarrow \neg(\exists_{x:\, block}(\text{holding}(x)))) \tag{B-LLM-25}$$

$$(\text{handempty}_g() \rightarrow \neg(\exists_{x:\, block}(\text{holding}_g(x)))) \tag{B-LLM-26}$$

With LLM-generated descriptions:

- B-LLM-1: At least one on relation appears in the goal (common in benchmark tasks).
- B-LLM-2: There is at least one block on the table initially.

- B-LLM-3: There exists at least one top block initially.
- B-LLM-4: If x is not held and nothing is on x, then x is clear (initially).
- B-LLM-5: At most one block can be held initially.
- B-LLM-6: Uniqueness of support: a block cannot be on two different blocks initially.
- B-LLM-7: Goal uniqueness of support: a block cannot be on two different blocks in the goal.
- B-LLM-8: No mutual on relation initially (no two-cycle).
- B-LLM-9: If x is above y initially, then y is not being held.
- B-LLM-10: If x is above y initially, x is supported initially.
- B-LLM-11: If x is above y in the goal, then y is not clear in the goal.
- B-LLM-12: A block on another block is not on the table.
- B-LLM-13: No 2-cycles in goal on-goal relation.
- B-LLM-14: If x is on y in the goal, x is not on the table in the goal.
- B-LLM-15: Held blocks are not supported (initially).
- B-LLM-16: Every block is either supported or held initially.
- B-LLM-17: If x is supported in the goal, it is not held in the goal.
- B-LLM-18: Top blocks are not held (initially).
- B-LLM-19: Top-goal blocks are not held in the goal.
- B-LLM-20: Uniqueness of occupant: at most one block can be on a given block initially.
- B-LLM-21: Goal uniqueness of occupant: at most one block can be on a given block in the goal.
- B-LLM-22: No block is held while another block is on it in the goal.
- B-LLM-23: A block cannot be on a clear block initially.
- B-LLM-24: Initially, either the hand is empty or some block is held.
- B-LLM-25: If the hand is empty, then no block is held (initially).
- B-LLM-26: If the goal specifies handempty, then no block is held in the goal.

Logistics

Candidate constraints:

$$\text{city}_{has\_airport}(c\colon city) \coloneqq \underset{a\colon airport}{\exists}(\text{in-city}(a, c))$$

$$\text{goal}_{at\_in\_city}(o\colon physobj,\ c\colon city) \coloneqq \underset{l\colon place}{\exists}(\text{at}_g(o, l) \land \text{loc}_{has\_city}(l, c))$$

$$\text{is}_{loaded}(p\colon package) \coloneqq \underset{v\colon vehicle}{\exists}(\text{in}(p, v))$$

$$\text{loc}_{has\_city}(l\colon place,\ c\colon city) \coloneqq \text{in-city}(l, c) \lor \text{in-city}_g(l, c)$$

$$\text{obj}_{at\_some\_place}(o\colon physobj) \coloneqq \underset{l\colon place}{\exists}(\text{at}(o, l))$$

$$\text{obj}_{in\_city}(o\colon physobj,\ c\colon city) \coloneqq \underset{l\colon place}{\exists}(\text{at}(o, l) \land \text{in-city}(l, c))$$

$$\text{pkg}_{in\_city}(p\colon package,\ c\colon city) \coloneqq \text{obj}_{in\_city}(p, c) \lor \underset{v\colon vehicle}{\exists}(\underset{l\colon place}{\exists}(\text{in}(p, v) \land \text{at}(v, l) \land \text{in-city}(l, c)))$$

$$\text{place}_{has\_some\_city}(l\colon place) \coloneqq \underset{c\colon city}{\exists}(\text{in-city}(l, c)) \lor \underset{c\colon city}{\exists}(\text{in-city}_g(l, c))$$

$$\text{same}_{city\_places}(l1\colon place,\ l2\colon place) \coloneqq \underset{c\colon city}{\exists}(\text{in-city}(l1, c) \land \text{in-city}(l2, c))$$

$$\text{vehicle}_{in\_city}(v\colon vehicle,\ c\colon city) \coloneqq \underset{l\colon place}{\exists}(\text{at}(v, l) \land \text{in-city}(l, c))$$

$$(\underset{a\colon airplane}{\exists}(\text{at}(a, ap)) \rightarrow \underset{ap2\colon airport}{\exists}(\text{at}(a, ap2))) \quad \text{(L-LLM-C1)}$$

$$(\underset{a\colon airplane}{\exists}(\text{at}(a, ap)) \rightarrow \underset{c\colon city}{\exists}(\text{city}_{has\_airport}(c))) \quad \text{(L-LLM-C2)}$$

$$(\underset{l\colon place}{\exists}(\text{in-city}(l, c)) \rightarrow \underset{c2\colon city}{\exists}(\text{in-city}(l, c2))) \quad \text{(L-LLM-C3)}$$

$$\underset{p\colon package}{\exists}(\underset{l\colon place}{\exists}(\text{at}_g(p, l))) \lor \underset{p\colon package}{\exists}(\underset{v\colon vehicle}{\exists}(\text{in}_g(p, v))) \quad \text{(L-LLM-C4)}$$

$$(\underset{p\colon package}{\exists}(\text{at}(p, l)) \rightarrow \underset{l2\colon place}{\exists}(\text{at}(p, l2)) \lor \underset{v2\colon vehicle}{\exists}(\text{in}(p, v2))) \quad \text{(L-LLM-C5)}$$

$$(\underset{p\colon package}{\exists}(\text{pkg}_{in\_city}(p, c)) \rightarrow \underset{c2\colon city}{\exists}(\text{pkg}_{in\_city}(p, c2))) \quad \text{(L-LLM-C6)}$$

$$(\underset{t\colon truck}{\exists}(\underset{l\colon place}{\exists}(\text{at}(t, l))) \rightarrow \underset{a\colon airport}{\exists}(\underset{c\colon city}{\exists}(\text{in-city}(a, c) \land \text{in-city}(l, c)))) \quad \text{(L-LLM-C7)}$$

$$(\underset{t\colon truck}{\exists}(\text{at}(t, l)) \rightarrow \underset{c\colon city}{\exists}(\underset{l2\colon place}{\exists}(\text{in-city}(l, c) \land \text{in-city}(l2, c)))) \quad \text{(L-LLM-C8)}$$

$$(\underset{t\colon truck}{\exists}(\text{at}(t, l)) \rightarrow \underset{l2\colon place}{\exists}(\text{at}(t, l2))) \quad \text{(L-LLM-C9)}$$

$$(\underset{v\colon vehicle}{\exists}(\text{at}(v, l)) \rightarrow \underset{l2\colon place}{\exists}(\underset{c2\colon city}{\exists}(\text{at}(v, l2) \land \text{in-city}(l2, c2)))) \quad \text{(L-LLM-C10)}$$

$$\underset{a\colon airplane}{\exists}(\underset{ap\colon airport}{\exists}(\underset{c\colon city}{\exists}(\text{at}(a, ap) \land \text{in-city}(ap, c)))) \quad \text{(L-LLM-C11)}$$

$$\underset{ap\colon airport}{\exists}(\underset{c\colon city}{\exists}(\text{in-city}(ap, c))) \quad \text{(L-LLM-C12)}$$

$$\underset{l\colon place}{\exists}(\underset{c\colon city}{\exists}(\text{in-city}(l, c))) \quad \text{(L-LLM-C13)}$$

$$\underset{o\colon physobj}{\exists}(\underset{l\colon place}{\exists}(\text{at}(o, l))) \quad \text{(L-LLM-C14)}$$

$$\underset{t\colon truck}{\exists}(\underset{c\colon city}{\exists}(\text{vehicle}_{in\_city}(t, c))) \quad \text{(L-LLM-C15)}$$

$$\underset{a\colon airplane}{\forall}(\underset{ap\colon airport}{\exists}(\text{at}(a, ap))) \quad \text{(L-LLM-C16)}$$

$$\forall_{a:\, airplane}\,(\,\exists_{ap:\, airport}\,(\mathrm{at}(a, ap)))$$

(L-LLM-C17)

$$\forall_{a:\, airplane}\,(\,\exists_{ap:\, airport}\,(\mathrm{at}(a, ap)))$$

(L-LLM-C18)

$$\forall_{a:\, airplane}\,(\,\forall_{ap1:\, airport}\,(\,\forall_{ap2:\, airport}\,((\mathrm{at}(a, ap1) \wedge \mathrm{at}(a, ap2) \rightarrow ap1 \neq ap2))))$$

(L-LLM-C19)

$$\forall_{a:\, airplane}\,(\,\forall_{ap1:\, airport}\,(\,\forall_{ap2:\, airport}\,((\mathrm{at}(a, ap1) \wedge \mathrm{at}(a, ap2) \rightarrow ap1 \neq ap2))))$$

(L-LLM-C20)

$$\forall_{a:\, airplane}\,(\,\forall_{l1:\, place}\,(\,\forall_{l2:\, place}\,((\mathrm{at}(a, l1) \wedge \text{in-city}(l1, c) \wedge \text{in-city}(l2, c) \rightarrow \mathrm{same}_{city\_places}(l1, l2)))))$$

(L-LLM-C21)

$$\forall_{a:\, airplane}\,(\,\forall_{l1:\, place}\,(\,\forall_{l2:\, place}\,((\mathrm{at}_g(a, l1) \wedge \mathrm{at}_g(a, l2) \rightarrow l1 \neq l2))))$$

(L-LLM-C22)

$$\forall_{a:\, airplane}\,(\,\forall_{l:\, location}\,(\neg(\mathrm{at}(a, l))))$$

(L-LLM-C23)

$$\forall_{a:\, airplane}\,(\,\forall_{l:\, place}\,((\mathrm{at}(a, l) \rightarrow \neg(\,\exists_{loc:\, location}\,(\mathrm{at}(a, loc))))))$$

(L-LLM-C24)

$$\forall_{a:\, airplane}\,(\,\forall_{l:\, place}\,((\mathrm{at}(a, l) \rightarrow \neg(\,\exists_{loc:\, location}\,(\mathrm{at}(a, loc))))))$$

(L-LLM-C25)

$$\forall_{ap:\, airport}\,(\,\exists_{c:\, city}\,(\text{in-city}(ap, c)))$$

(L-LLM-C26)

$$\forall_{ap:\, airport}\,(\,\forall_{c1:\, city}\,(\,\forall_{c2:\, city}\,((\text{in-city}(ap, c1) \wedge \text{in-city}(ap, c2) \rightarrow c1 \neq c2))))$$

(L-LLM-C27)

$$\forall_{c:\, city}\,((\,\exists_{a:\, airport}\,(\text{in-city}(a, c)) \rightarrow \mathrm{city}_{has\_airport}(c)))$$

(L-LLM-C28)

$$\forall_{c:\, city}\,(\,\exists_{p:\, place}\,(\text{in-city}(p, c)))$$

(L-LLM-C29)

$$\forall_{c:\, city}\,((\mathrm{city}_{has\_airport}(c) \rightarrow \,\exists_{a:\, airport}\,(\text{in-city}(a, c))))$$

(L-LLM-C30)

$$\forall_{c:\, city}\,(\mathrm{city}_{has\_airport}(c))$$

(L-LLM-C31)

$$\forall_{l1:\, place}\,(\,\forall_{l2:\, place}\,(\,\forall_{c:\, city}\,((\text{in-city}(l1, c) \wedge \text{in-city}(l2, c) \rightarrow \mathrm{same}_{city\_places}(l1, l2)))))$$

(L-LLM-C32)

$$\forall_{l1:\, place}\,(\,\forall_{l2:\, place}\,((\mathrm{same}_{city\_places}(l1, l2) \rightarrow \,\exists_{c:\, city}\,(\text{in-city}(l1, c) \wedge \text{in-city}(l2, c)))))$$

(L-LLM-C33)

$$\forall_{l:\, place}\,((\,\exists_{c:\, city}\,(\text{in-city}(l, c)) \vee \,\exists_{c:\, city}\,(\text{in-city}_g(l, c)) \rightarrow \mathrm{place}_{has\_some\_city}(l)))$$

(L-LLM-C34)

$$\forall_{l:\, place}\,(\,\forall_{c1:\, city}\,(\,\forall_{c2:\, city}\,((\text{in-city}(l, c1) \wedge \text{in-city}(l, c2) \rightarrow c1 \neq c2))))$$

(L-LLM-C35)

$$\forall_{l:\, place}\,(\,\forall_{c1:\, city}\,(\,\forall_{c2:\, city}\,((\text{in-city}_g(l, c1) \wedge \text{in-city}_g(l, c2) \rightarrow c1 \neq c2))))$$

(L-LLM-C36)

$$\forall_{l:\, place}(\forall_{c:\, city}((\text{in-city}(l,c) \vee \text{in-city}_g(l,c) \rightarrow \text{loc}_{has\_city}(l,c))))$$ (L-LLM-C37)

$$\forall_{l:\, place}(\forall_{c:\, city}((\text{loc}_{has\_city}(l,c) \rightarrow \text{in-city}(l,c) \vee \text{in-city}_g(l,c))))$$ (L-LLM-C38)

$$\forall_{l:\, place}((\text{place}_{has\_some\_city}(l) \rightarrow \exists_{c:\, city}(\text{in-city}(l,c)) \vee \exists_{c:\, city}(\text{in-city}_g(l,c))))$$ (L-LLM-C39)

$$\forall_{l:\, place}(\text{place}_{has\_some\_city}(l))$$ (L-LLM-C40)

$$\forall_{loc:\, location}(\exists_{c:\, city}(\text{in-city}(loc,c)))$$ (L-LLM-C41)

$$\forall_{loc:\, location}(\forall_{c1:\, city}(\forall_{c2:\, city}((\text{in-city}(loc,c1) \wedge \text{in-city}(loc,c2) \rightarrow c1 \neq c2))))$$ (L-LLM-C42)

$$\forall_{o:\, physobj}((\exists_{l:\, place}(\text{at}(o,l)) \rightarrow \text{obj}_{at\_some\_place}(o)))$$ (L-LLM-C43)

$$\forall_{o:\, physobj}(\forall_{c1:\, city}(\forall_{c2:\, city}((\text{goal}_{at\_in\_city}(o,c1) \wedge \text{goal}_{at\_in\_city}(o,c2) \rightarrow c1 \neq c2))))$$ (L-LLM-C44)

$$\forall_{o:\, physobj}(\forall_{c1:\, city}(\forall_{c2:\, city}((\text{obj}_{in\_city}(o,c1) \wedge \text{obj}_{in\_city}(o,c2) \rightarrow c1 \neq c2))))$$ (L-LLM-C45)

$$\forall_{o:\, physobj}(\forall_{c:\, city}((\text{goal}_{at\_in\_city}(o,c) \rightarrow \exists_{l:\, place}(\text{at}_g(o,l) \wedge \text{loc}_{has\_city}(l,c)))))$$ (L-LLM-C46)

$$\forall_{o:\, physobj}(\forall_{c:\, city}((\text{obj}_{in\_city}(o,c) \rightarrow \exists_{l:\, place}(\text{at}(o,l) \wedge \text{in-city}(l,c)))))$$ (L-LLM-C47)

$$\forall_{o:\, physobj}(\forall_{l1:\, place}(\forall_{l2:\, place}((\text{at}(o,l1) \wedge \text{at}(o,l2) \rightarrow l1 \neq l2))))$$ (L-LLM-C48)

$$\forall_{o:\, physobj}(\forall_{l1:\, place}(\forall_{l2:\, place}((\text{at}_g(o,l1) \wedge \text{at}_g(o,l2) \rightarrow l1 \neq l2))))$$ (L-LLM-C49)

$$\forall_{o:\, physobj}(\forall_{l1:\, place}(\forall_{l2:\, place}(\forall_{c:\, city}((\text{at}(o,l1) \wedge \text{in-city}(l1,c) \wedge \text{in-city}(l2,c) \rightarrow \text{same}_{city\_places}(l1,l2))))))$$ (L-LLM-C50)

$$\forall_{o:\, physobj}(\forall_{l:\, place}(\forall_{c:\, city}((\text{at}_g(o,l) \wedge \text{in-city}(l,c) \rightarrow \text{goal}_{at\_in\_city}(o,c)))))$$ (L-LLM-C51)

$$\forall_{o:\, physobj}(\forall_{l:\, place}(\forall_{c:\, city}((\text{at}_g(o,l) \wedge \text{in-city}_g(l,c) \rightarrow \text{goal}_{at\_in\_city}(o,c)))))$$ (L-LLM-C52)

$$\forall_{o:\, physobj}(\forall_{l:\, place}(\forall_{c:\, city}((\text{at}_g(o,l) \wedge \text{loc}_{has\_city}(l,c) \rightarrow \text{goal}_{at\_in\_city}(o,c)))))$$ (L-LLM-C53)

$$\forall_{o:\, physobj}(\forall_{l:\, place}((\text{at}(o,l) \rightarrow \exists_{c:\, city}(\text{in-city}(l,c)))))$$ (L-LLM-C54)

$$\forall_{o:\, physobj}(\forall_{l:\, place}((\text{at}(o,l) \rightarrow \text{place}_{has\_some\_city}(l))))$$ (L-LLM-C55)

$$\forall_{o:\, physobj}(\forall_{l:\, place}((\text{at}_g(o,l) \rightarrow \exists_{c:\, city}(\text{in-city}(l,c)) \vee \exists_{c:\, city}(\text{in-city}_g(l,c)))))$$ (L-LLM-C56)

$$\forall_{o:\, physobj}((\text{obj}_{at\_some\_place}(o) \rightarrow \exists_{l:\, place}(\text{at}(o,l))))$$
(L-LLM-C57)

$$\forall_{p:\, package}(\exists_{l:\, place}(\text{at}(p,l)) \vee \exists_{v:\, vehicle}(\text{in}(p,v)))$$
(L-LLM-C58)

$$\forall_{p:\, package}((\exists_{l:\, place}(\text{at}(p,l)) \rightarrow \exists_{c:\, city}(\text{pkg}_{in\_city}(p,c))))$$
(L-LLM-C59)

$$\forall_{p:\, package}(\exists_{c:\, city}(\text{pkg}_{in\_city}(p,c)))$$
(L-LLM-C60)

$$\forall_{p:\, package}(\forall_{ap:\, airport}(\forall_{l:\, place}((\text{at}(p,ap) \wedge \text{at}(p,l) \rightarrow ap \neq l))))$$
(L-LLM-C61)

$$\forall_{p:\, package}(\forall_{c1:\, city}(\forall_{c2:\, city}((\text{pkg}_{in\_city}(p,c1) \wedge \text{pkg}_{in\_city}(p,c2) \rightarrow c1 \neq c2))))$$
(L-LLM-C62)

$$\forall_{p:\, package}(\forall_{l1:\, place}(\forall_{l2:\, place}((\text{at}(p,ap) \wedge \text{at}(p,l2) \rightarrow ap \neq l2))))$$
(L-LLM-C63)

$$\forall_{p:\, package}(\forall_{l1:\, place}(\forall_{l2:\, place}((\text{at}(p,l1) \wedge \text{at}(p,l2) \rightarrow l1 \neq l2))))$$
(L-LLM-C64)

$$\forall_{p:\, package}(\forall_{l:\, place}(\forall_{v:\, vehicle}(\neg(\text{at}_g(p,l) \wedge \text{in}_g(p,v)))))$$
(L-LLM-C65)

$$\forall_{p:\, package}(\forall_{l:\, place}((\text{at}(p,l) \rightarrow \exists_{c:\, city}(\text{in-city}(l,c)))))$$
(L-LLM-C66)

$$\forall_{p:\, package}(\forall_{l:\, place}((\text{at}(p,l) \rightarrow \text{place}_{has\_some\_city}(l))))$$
(L-LLM-C67)

$$\forall_{p:\, package}(\forall_{l:\, place}((\text{at}_g(p,l) \rightarrow \neg(\exists_{v:\, vehicle}(\text{in}_g(p,v))))))$$
(L-LLM-C68)

$$\forall_{p:\, package}(\forall_{v1:\, vehicle}(\forall_{v2:\, vehicle}((\text{in}(p,v1) \wedge \text{in}(p,v2) \rightarrow v1 \neq v2))))$$
(L-LLM-C69)

$$\forall_{p:\, package}(\forall_{v1:\, vehicle}(\forall_{v2:\, vehicle}((\text{in}_g(p,v1) \wedge \text{in}_g(p,v2) \rightarrow v1 \neq v2))))$$
(L-LLM-C70)

$$\forall_{p:\, package}(\forall_{v1:\, vehicle}(\forall_{v2:\, vehicle}((\text{in}_g(p,v1) \wedge \text{in}_g(p,v2) \rightarrow v1 \neq v2))))$$
(L-LLM-C71)

$$\forall_{p:\, package}(\forall_{v:\, vehicle}(\forall_{l:\, place}(\neg(\text{in}(p,v) \wedge \text{at}(p,l)))))$$
(L-LLM-C72)

$$\forall_{p:\, package}(\forall_{v:\, vehicle}((\text{in}(p,v) \rightarrow \exists_{c:\, city}(\text{vehicle}_{in\_city}(v,c)))))$$
(L-LLM-C73)

$$\forall_{p:\, package}(\forall_{v:\, vehicle}((\text{in}(p,v) \rightarrow \exists_{l:\, place}(\text{at}(v,l)))))$$
(L-LLM-C74)

$$\forall_{p:\, package}(\forall_{v:\, vehicle}((\text{in}(p,v) \rightarrow \neg(\exists_{l:\, place}(\text{at}(p,l))))))$$
(L-LLM-C75)

$$\forall_{p:\, package}(\forall_{v:\, vehicle}((\text{in}(p,v) \rightarrow \text{obj}_{at\_some\_place}(v))))$$
(L-LLM-C76)

$$\forall_{p:\, package}\,(\forall_{v:\, vehicle}\,((\text{in}_g(p,v) \rightarrow \neg(\exists_{l:\, place}\,(\text{at}_g(p,l))))))$$ (L-LLM-C77)

$$\forall_{p:\, package}\,(\neg(\exists_{l:\, place}\,(\text{at}(p,l)) \wedge \exists_{v:\, vehicle}\,(\text{in}(p,v))))$$ (L-LLM-C78)

$$\forall_{p:\, package}\,((\text{is}_{loaded}(p) \rightarrow \exists_{c:\, city}\,(\text{pkg}_{in\_city}(p,c))))$$ (L-LLM-C79)

$$\forall_{p:\, package}\,((\text{is}_{loaded}(p) \rightarrow \exists_{v:\, vehicle}\,(\text{in}(p,v))))$$ (L-LLM-C80)

$$\forall_{p:\, package}\,((\text{is}_{loaded}(p) \rightarrow \neg(\text{obj}_{at\_some\_place}(p))))$$ (L-LLM-C81)

$$\forall_{p:\, package}\,(\text{obj}_{at\_some\_place}(p) \vee \text{is}_{loaded}(p))$$ (L-LLM-C82)

$$\forall_{p:\, package}\,((\text{obj}_{at\_some\_place}(p) \rightarrow \neg(\text{is}_{loaded}(p))))$$ (L-LLM-C83)

$$\forall_{p:\, package}\,(\text{at}_g(p,l1))$$ (L-LLM-C84)

$$\forall_{pl:\, place}\,(\exists_{c:\, city}\,(\text{in-city}(pl,c)))$$ (L-LLM-C85)

$$\forall_{t:\, truck}\,(\exists_{l:\, place}\,(\text{at}(t,l)))$$ (L-LLM-C86)

$$\forall_{t:\, truck}\,(\forall_{l1:\, place}\,(\forall_{l2:\, place}\,((\text{at}(t,l1) \wedge \text{at}(t,l2) \rightarrow l1 \neq l2))))$$ (L-LLM-C87)

$$\forall_{t:\, truck}\,(\forall_{l1:\, place}\,(\forall_{l2:\, place}\,((\text{at}_g(t,l1) \wedge \text{at}_g(t,l2) \rightarrow l1 \neq l2))))$$ (L-LLM-C88)

$$\forall_{t:\, truck}\,(\forall_{l1:\, place}\,(\forall_{l2:\, place}\,((\text{at}_g(t,l1) \wedge \text{at}_g(t,l2) \rightarrow l1 \neq l2))))$$ (L-LLM-C89)

$$\forall_{v:\, vehicle}\,(\exists_{c:\, city}\,(\text{obj}_{in\_city}(v,c)))$$ (L-LLM-C90)

$$\forall_{v:\, vehicle}\,(\exists_{c:\, city}\,(\text{vehicle}_{in\_city}(v,c)))$$ (L-LLM-C91)

$$\forall_{v:\, vehicle}\,(\exists_{l:\, place}\,(\text{at}(v,l)))$$ (L-LLM-C92)

$$\forall_{v:\, vehicle}\,(\forall_{c1:\, city}\,(\forall_{c2:\, city}\,((\text{vehicle}_{in\_city}(v,c1) \wedge \text{vehicle}_{in\_city}(v,c2) \rightarrow c1 \neq c2))))$$ (L-LLM-C93)

$$\forall_{v:\, vehicle}\,(\forall_{l1:\, place}\,(\forall_{l2:\, place}\,((\text{at}(v,l1) \wedge \text{at}(v,l2) \rightarrow l1 \neq l2))))$$ (L-LLM-C94)

$$\forall_{v:\, vehicle}\,(\forall_{l1:\, place}\,(\forall_{l2:\, place}\,(\forall_{ap:\, airport}\,((\text{at}(v,l1) \wedge \text{in-city}(l1,c) \wedge \text{in-city}(ap,c) \rightarrow \text{same}_{city\_places}(l1,ap))))))$$ (L-LLM-C95)

$$\forall_{v:\, vehicle}\,(\forall_{l1:\, place}\,(\forall_{l2:\, place}\,(\forall_{c:\, city}\,((\text{at}(v,l1) \wedge \text{in-city}(l1,c) \wedge \text{in-city}(l2,c) \rightarrow \text{same}_{city\_places}(l1,l2))))))$$ (L-LLM-C96)

$$\forall_{v:\, vehicle}(\forall_{l:\, place}((\text{at}(v,l) \rightarrow \exists_{c:\, city}(\text{in-city}(l,c)))))$$ (L-LLM-C97)

$$\forall_{v:\, vehicle}(\forall_{l:\, place}((\text{at}(v,l) \rightarrow \text{place}_{has\_some\_city}(l))))$$ (L-LLM-C98)

$$\forall_{v:\, vehicle}(\forall_{l:\, place}((\text{at}_g(v,l) \rightarrow \text{place}_{has\_some\_city}(l))))$$ (L-LLM-C99)

$$\forall_{v:\, vehicle}(\text{obj}_{at\_some\_place}(v))$$ (L-LLM-C100)

With LLM-generated descriptions:

- L-LLM-C1: If there is any airplane, there exists an airport where some airplane is located (restating existence).
- L-LLM-C2: If there is any airplane, there is some city with an airport (typical setup).
- L-LLM-C3: If there exists any place-city fact, then there exists a city participating in such a fact (trivial existence form).
- L-LLM-C4: Goal specifies at least one package goal (either at some place or inside some vehicle).
- L-LLM-C5: If there exists a package, then either it is at some place or in some vehicle (existential form).
- L-LLM-C6: If there exists a package, then there exists some city that (by definition) contains it initially.
- L-LLM-C7: If some truck exists at a place, its city has at least one airport (typical property of the domain instances).
- L-LLM-C8: If a truck exists at some place, then some city with at least one place exists (restates city-place existence via the truck's location).
- L-LLM-C9: If there is any truck, there exists some place where a truck is located (restating existence).
- L-LLM-C10: If there exists any vehicle, then some vehicle-place pairing is associated with a city (some place has a city).
- L-LLM-C11: There exists an airplane at an airport that belongs to some city (applicable when airplanes exist).
- L-LLM-C12: There exists an airport in some city (typical of the instances).
- L-LLM-C13: There exists at least one place-city association in the initial state.
- L-LLM-C14: There exists at least one at fact initially.
- L-LLM-C15: There exists a truck that is in some city initially (applicable when trucks exist).
- L-LLM-C16: Every airplane is at some airport initially.
- L-LLM-C17: Each airplane's at-fact references an airport (existentially specialized).
- L-LLM-C18: Each airplane has at least one at-fact to an airport (restated).
- L-LLM-C19: Airplane is at at most one airport initially (airport-specialized uniqueness).
- L-LLM-C20: Airplane is at most one airport initially (duplicate specialization for robustness).
- L-LLM-C21: Airplane location shares a city with any place in that city (uses same-city-places).
- L-LLM-C22: Airplane goal locations are unique if specified.
- L-LLM-C23: Airplanes are not initially at non-airport locations.
- L-LLM-C24: Airplanes are never at typed locations initially (restates airplane-location restriction).
- L-LLM-C25: Airplanes are only at airports initially (no at-fact to typed locations).
- L-LLM-C26: Every airport belongs to some city initially.
- L-LLM-C27: Each airport has at most one city association initially.
- L-LLM-C28: If there exists an airport in a city, then city-has-airport holds for that city (derived definition direction).
- L-LLM-C29: Every city has at least one place initially.
- L-LLM-C30: Derived predicate city-has-airport implies there is some airport place in that city.
- L-LLM-C31: Every city has at least one airport (typical of logistics benchmarks).
- L-LLM-C32: If two places share a city in the initial facts, they are marked as same-city-places.
- L-LLM-C33: same-city-places implies there exists a city they both belong to.
- L-LLM-C34: Definition check: if a place has a city via init/goal facts, then place-has-some-city holds.
- L-LLM-C35: Each place is associated with at most one city in the initial facts.

- L-LLM-C36: Each place has at most one city association in the goal facts.
- L-LLM-C37: Definition check: if a location-city relation holds in init or goal, then loc-has-city holds.
- L-LLM-C38: Definition check: loc-has-city implies the location-city relation comes from init or goal facts.
- L-LLM-C39: Definition check: place-has-some-city implies there exists a city via init or goal facts.
- L-LLM-C40: Every place is associated with some city, via init or goal facts (derived).
- L-LLM-C41: Every location belongs to some city initially.
- L-LLM-C42: Each location has at most one city association initially.
- L-LLM-C43: Definition check: if an object has an at-fact then obj-at-some-place holds.
- L-LLM-C44: Each object is required by the goal to be in at most one city (via its goal at-location).
- L-LLM-C45: Any object is counted in at most one city initially (because it has at most one location).
- L-LLM-C46: Definition check: goal-at-in-city implies an at-goal at some place whose city is that city.
- L-LLM-C47: Definition check: obj-in-city implies object at a place whose city matches.
- L-LLM-C48: No physobj can be at two different places simultaneously in the initial state.
- L-LLM-C49: No object is required to be at two different places in the goal specification.
- L-LLM-C50: If an object is at l1 in city c and l2 is also in c, then l1 and l2 are same-city-places.
- L-LLM-C51: If a goal requires an object at l and l is known in init to be in city c, then the goal implies the object is in city c.
- L-LLM-C52: If a goal requires an object at a place with a goal city assignment, then the goal implies the object is in that city.
- L-LLM-C53: Definition check: at-goal at a place with a known city implies goal-at-in-city.
- L-LLM-C54: Every place that holds an object is associated with some city initially.
- L-LLM-C55: Any place with an at-fact has some city association (via init or goal facts, derived).
- L-LLM-C56: Any place appearing in a goal at-fact is associated with a city (from init or goal facts).
- L-LLM-C57: Definition check: obj-at-some-place implies there exists an at-fact.
- L-LLM-C58: Each package is either at some place or inside some vehicle initially.
- L-LLM-C59: Packages that are at some place are in some city (via pkg-in-city).
- L-LLM-C60: Every package is located in some city initially (either at a place or inside a vehicle at a place).
- L-LLM-C61: A package cannot be at two different places when one of them is an airport (redundant specialization).
- L-LLM-C62: Each package is associated with at most one city initially (derived).
- L-LLM-C63: If a package is at an airport and at another place, those places must be different (restates at-uniqueness with airport specialization).
- L-LLM-C64: Any package has at most one location initially.
- L-LLM-C65: A package goal cannot require being at a place and inside a vehicle at the same time.
- L-LLM-C66: Any place that holds a package is linked to a city initially.
- L-LLM-C67: Any place holding a package has some city association (derived).
- L-LLM-C68: If a package is required at a place in the goal, it is not simultaneously required to be in any vehicle.
- L-LLM-C69: A package cannot simultaneously be inside two different vehicles initially.
- L-LLM-C70: No package is required to be in two different vehicles in the goal specification.
- L-LLM-C71: Goal in-facts for a package do not specify two different vehicles (redundant check).
- L-LLM-C72: Packages cannot be both at a place and inside a vehicle initially.
- L-LLM-C73: If a package is inside a vehicle, then that vehicle is in some city initially.
- L-LLM-C74: If a package is inside a vehicle initially, that vehicle is at some place initially.
- L-LLM-C75: If a package is inside a vehicle initially, it is not at any place initially (implication form).
- L-LLM-C76: If a package is inside a vehicle initially, that vehicle has an at fact (derived form).
- L-LLM-C77: If a package is required to be in a vehicle in the goal, it is not simultaneously required to be at any place (goal exclusivity as implication).
- L-LLM-C78: No package is simultaneously specified as both at some place and inside some vehicle initially (global exclusivity).
- L-LLM-C79: Loaded packages belong to some city via the carrying vehicle's location (derived).
- L-LLM-C80: Derived predicate is-loaded implies there exists a corresponding in-fact.
- L-LLM-C81: If a package is loaded, then it is not at any place (derived consistency).

- L-LLM-C82: Each package either has an at fact or is loaded (derived version of the disjunction).
- L-LLM-C83: Packages that are at some place are not loaded (derived exclusivity).
- L-LLM-C84: Packages that appear in at-goal are goal-positioned at some place (tautological existence via quantifier).
- L-LLM-C85: Every place belongs to some city initially.
- L-LLM-C86: Every truck is at some place initially.
- L-LLM-C87: Truck is at at most one place initially (restates uniqueness for trucks).
- L-LLM-C88: Truck goal locations are unique if specified.
- L-LLM-C89: Truck goal locations are unique (redundant specialization).
- L-LLM-C90: Every vehicle belongs to some city via obj-in-city as well (redundant direction).
- L-LLM-C91: Every vehicle is in some city initially (via its location).
- L-LLM-C92: Every vehicle is placed at some place in the initial state.
- L-LLM-C93: Vehicles have at most one city in which they are located initially (via their single location).
- L-LLM-C94: Any vehicle has at most one location initially.
- L-LLM-C95: Any vehicle's location l1 shares a city with any airport in the same city (uses same-city-places).
- L-LLM-C96: If a vehicle is at l1 in city c and l2 is also in c, then l1 and l2 are same-city-places.
- L-LLM-C97: Any place that holds a vehicle is linked to a city initially.
- L-LLM-C98: Any place holding a vehicle has some city association (derived).
- L-LLM-C99: Any place appearing in a vehicle at-goal also has some city association.
- L-LLM-C100: Every vehicle has some at fact initially (derived).

Which gives the reduced set:

$$\left(\exists_{a:\, airplane}(\mathrm{at}(a, ap)) \rightarrow \exists_{ap2:\, airport}(\mathrm{at}(a, ap2))\right) \quad \text{(L-LLM-1)}$$

$$\left(\exists_{l:\, place}(\text{in-city}(l, c)) \rightarrow \exists_{c2:\, city}(\text{in-city}(l, c2))\right) \quad \text{(L-LLM-2)}$$

$$\exists_{p:\, package}\left(\exists_{l:\, place}(\mathrm{at}_{goal}(p, l))\right) \vee \exists_{p:\, package}\left(\exists_{v:\, vehicle}(\mathrm{in}_{goal}(p, v))\right) \quad \text{(L-LLM-3)}$$

$$\left(\exists_{p:\, package}(\mathrm{at}(p, l)) \rightarrow \exists_{l2:\, place}(\mathrm{at}(p, l2)) \vee \exists_{v2:\, vehicle}(\mathrm{in}(p, v2))\right) \quad \text{(L-LLM-4)}$$

$$\left(\exists_{p:\, package}(\mathrm{pkg}_{in\_city}(p, c)) \rightarrow \exists_{c2:\, city}(\mathrm{pkg}_{in\_city}(p, c2))\right) \quad \text{(L-LLM-5)}$$

$$\left(\exists_{t:\, truck}(\mathrm{at}(t, l)) \rightarrow \exists_{c:\, city}\left(\exists_{l2:\, place}(\text{in-city}(l, c) \wedge \text{in-city}(l2, c))\right)\right) \quad \text{(L-LLM-6)}$$

$$\left(\exists_{t:\, truck}(\mathrm{at}(t, l)) \rightarrow \exists_{l2:\, place}(\mathrm{at}(t, l2))\right) \quad \text{(L-LLM-7)}$$

$$\left(\exists_{v:\, vehicle}(\mathrm{at}(v, l)) \rightarrow \exists_{l2:\, place}\left(\exists_{c2:\, city}(\mathrm{at}(v, l2) \wedge \text{in-city}(l2, c2))\right)\right) \quad \text{(L-LLM-8)}$$

$$\exists_{t:\, truck}\left(\exists_{c:\, city}(\mathrm{vehicle}_{in\_city}(t, c))\right) \quad \text{(L-LLM-9)}$$

$$\forall_{a:\, airplane}\left(\forall_{l1:\, place}\left(\forall_{l2:\, place}((\mathrm{at}(a, l1) \wedge \text{in-city}(l1, c) \wedge \text{in-city}(l2, c) \rightarrow \mathrm{same}_{city\_places}(l1, l2))\right)\right)\right) \quad \text{(L-LLM-10)}$$

$$\forall_{a:\, airplane}\left(\forall_{l1:\, place}\left(\forall_{l2:\, place}((\mathrm{at}_{goal}(a, l1) \wedge \mathrm{at}_{goal}(a, l2) \rightarrow l1 \neq l2))\right)\right) \quad \text{(L-LLM-11)}$$

$$\forall_{a:\, airplane}\left(\forall_{l:\, place}((\mathrm{at}(a, l) \rightarrow \neg(\exists_{loc:\, location}(\mathrm{at}(a, loc)))))\right) \quad \text{(L-LLM-12)}$$

$$\forall_{c:\, city}(\mathrm{city}_{has\_airport}(c)) \quad \text{(L-LLM-13)}$$

$$\forall_{l:\, place}\left(\forall_{c1:\, city}\left(\forall_{c2:\, city}((\text{in-city}_{goal}(l, c1) \wedge \text{in-city}_{goal}(l, c2) \rightarrow c1 \neq c2))\right)\right) \quad \text{(L-LLM-14)}$$

$$\forall_{p:\, package}\left(\forall_{l1:\, place}\left(\forall_{l2:\, place}((\mathrm{at}(p, ap) \wedge \mathrm{at}(p, l2) \rightarrow ap \neq l2))\right)\right) \quad \text{(L-LLM-15)}$$

$$\forall_{p:\, package}\left(\forall_{v1:\, vehicle}\left(\forall_{v2:\, vehicle}((\mathrm{in}(p, v1) \wedge \mathrm{in}(p, v2) \rightarrow v1 \neq v2))\right)\right) \quad \text{(L-LLM-16)}$$

$$\forall_{p:\, package}\left(\forall_{v1:\, vehicle}\left(\forall_{v2:\, vehicle}((\mathrm{in}_{goal}(p, v1) \wedge \mathrm{in}_{goal}(p, v2) \rightarrow v1 \neq v2))\right)\right) \quad \text{(L-LLM-17)}$$

$$\forall_{p:\, package}\,(\mathrm{obj}_{at\_some\_place}(p) \vee \mathrm{is}_{loaded}(p)) \quad \text{(L-LLM-18)}$$

$$\forall_{pl:\, place}\,(\exists_{c:\, city}\,(\text{in-city}(pl, c))) \quad \text{(L-LLM-19)}$$

$$\forall_{t:\, truck}\,(\exists_{l:\, place}\,(\mathrm{at}(t, l))) \quad \text{(L-LLM-20)}$$

$$\forall_{t:\, truck}\,(\forall_{l1:\, place}\,(\forall_{l2:\, place}\,((\mathrm{at}_{goal}(t, l1) \wedge \mathrm{at}_{goal}(t, l2) \rightarrow l1 \neq l2)))) \quad \text{(L-LLM-21)}$$

$$\forall_{v:\, vehicle}\,(\forall_{l1:\, place}\,(\forall_{l2:\, place}\,(\forall_{ap:\, airport}\,((\mathrm{at}(v, l1) \wedge \text{in-city}(l1, c) \wedge \text{in-city}(ap, c) \rightarrow \mathrm{same}_{city\_places}(l1, ap)))))) \quad \text{(L-LLM-22)}$$

With LLM-generated descriptions:

- L-LLM-1: If there is any airplane, there exists an airport where some airplane is located (restating existence).
- L-LLM-2: If there exists any place-city fact, then there exists a city participating in such a fact (trivial existence form).
- L-LLM-3: Goal specifies at least one package goal (either at some place or inside some vehicle).
- L-LLM-4: If there exists a package, then either it is at some place or in some vehicle (existential form).
- L-LLM-5: If there exists a package, then there exists some city that (by definition) contains it initially.
- L-LLM-6: If a truck exists at some place, then some city with at least one place exists (restates city-place existence via the truck's location).
- L-LLM-7: If there is any truck, there exists some place where a truck is located (restating existence).
- L-LLM-8: If there exists any vehicle, then some vehicle-place pairing is associated with a city (some place has a city).
- L-LLM-9: There exists a truck that is in some city initially (applicable when trucks exist).
- L-LLM-10: Airplane location shares a city with any place in that city (uses same-city-places).
- L-LLM-11: Airplane goal locations are unique if specified.
- L-LLM-12: Airplanes are only at airports initially (no at-fact to typed locations).
- L-LLM-13: Every city has at least one airport (typical of logistics benchmarks).
- L-LLM-14: Each place has at most one city association in the goal facts.
- L-LLM-15: If a package is at an airport and at another place, those places must be different (restates at-uniqueness with airport specialization).
- L-LLM-16: A package cannot simultaneously be inside two different vehicles initially.
- L-LLM-17: Goal in-facts for a package do not specify two different vehicles (redundant check).
- L-LLM-18: Each package either has an at fact or is loaded (derived version of the disjunction).
- L-LLM-19: Every place belongs to some city initially.
- L-LLM-20: Every truck is at some place initially.
- L-LLM-21: Truck goal locations are unique (redundant specialization).
- L-LLM-22: Any vehicle's location l1 shares a city with any airport in the same city (uses same-city-places).

Transport

Candidate constraints:

$$\text{cap}_{reach}(s1\text{: Cap-Num},\ s2\text{: Cap-Num}) \coloneqq \neg(s1 \neq s2) \vee \text{cap-pre}(s1, s2) \vee \exists_{m\text{: Cap-Num}}(\text{cap-pre}(s1, m) \wedge \text{cap}_{reach}(m, s2))$$

$$\text{goal}_{loc}(l\text{: } location) \coloneqq \exists_{p\text{: } package}(\text{at}_g(p, l))$$

$$\text{has}_{incoming}(l\text{: } location) \coloneqq \exists_{m\text{: } location}(\text{road}(m, l))$$

$$\text{has}_{outgoing}(l\text{: } location) \coloneqq \exists_{m\text{: } location}(\text{road}(l, m))$$

$$\text{max}_{capacity}(s\text{: Cap-Num}) \coloneqq \neg(\exists_{t\text{: Cap-Num}}(\text{cap-pre}(s, t)))$$

$$\text{min}_{capacity}(s\text{: Cap-Num}) \coloneqq \neg(\exists_{t\text{: Cap-Num}}(\text{cap-pre}(t, s)))$$

$$\text{pkg}_{is\_g}(p\text{: } package) \coloneqq \exists_{l\text{: } location}(\text{at}_g(p, l))$$

$$\text{reach}(l1\text{: } location,\ l2\text{: } location) \coloneqq \neg(l1 \neq l2) \vee \text{road}(l1, l2) \vee \exists_{m\text{: } location}(\text{road}(l1, m) \wedge \text{reach}(m, l2))$$

$$\text{same}_{comp}(a\text{: } location,\ b\text{: } location) \coloneqq \text{reach}(a, b) \wedge \text{reach}(b, a)$$

$$\text{veh}_{can\_reach\_g}(v\text{: } vehicle,\ l\text{: } location) \coloneqq \exists_{lv\text{: } location}(\text{at}(v, lv) \wedge \text{reach}(lv, l))$$

$$(\exists_{l1\text{: } location}(\exists_{l2\text{: } location}(\text{road}(l1, l2))) \rightarrow \exists_{l\text{: } location}(\text{has}_{outgoing}(l))) \quad \text{(T-LLM-C1)}$$

$$(\exists_{l\text{: } location}(\exists_{p\text{: } package}(\text{at}(p, l))) \rightarrow \exists_{v\text{: } vehicle}(\text{veh}_{can\_reach\_g}(v, l))) \quad \text{(T-LLM-C2)}$$

$$(\exists_{l\text{: } location}(\text{goal}_{loc}(l)) \rightarrow \exists_{p\text{: } package}(\text{pkg}_{is\_g}(p))) \quad \text{(T-LLM-C3)}$$

$$(\exists_{p\text{: } package}(\exists_{l\text{: } location}(\text{at}(p, l)) \vee \exists_{v\text{: } vehicle}(\text{in}(p, v))) \rightarrow \exists_{v\text{: } vehicle}(\exists_{s\text{: Cap-Num}}(\text{capacity}(v, s)))) \quad \text{(T-LLM-C4)}$$

$$(\exists_{p\text{: } package}(\text{pkg}_{is\_g}(p)) \rightarrow \exists_{a\text{: } location}(\exists_{b\text{: } location}(\text{road}(a, b)))) \quad \text{(T-LLM-C5)}$$

$$(\exists_{p\text{: } package}(\text{pkg}_{is\_g}(p)) \rightarrow \exists_{p2\text{: } package}(\exists_{l\text{: } location}(\text{at}(p2, l)) \vee \exists_{v\text{: } vehicle}(\text{in}(p2, v)))) \quad \text{(T-LLM-C6)}$$

$$(\exists_{p\text{: } package}(\text{pkg}_{is\_g}(p)) \rightarrow \exists_{v\text{: } vehicle}(\exists_{l\text{: } location}(\text{at}(v, l) \wedge \text{reach}(l, l)))) \quad \text{(T-LLM-C7)}$$

$$(\exists_{p\text{: } package}(\text{pkg}_{is\_g}(p)) \rightarrow \exists_{v\text{: } vehicle}(\exists_{l\text{: } location}(\text{veh}_{can\_reach\_g}(v, l)))) \quad \text{(T-LLM-C8)}$$

$$(\exists_{p\text{: } package}(\text{pkg}_{is\_g}(p)) \rightarrow \exists_{v\text{: } vehicle}(\exists_{s\text{: Cap-Num}}(\text{capacity}(v, s)))) \quad \text{(T-LLM-C9)}$$

$$(\exists_{v\text{: } vehicle}(\exists_{s\text{: Cap-Num}}(\text{capacity}(v, s))) \rightarrow \exists_{x\text{: } locatable}(\exists_{l\text{: } location}(\text{at}(x, l)))) \quad \text{(T-LLM-C10)}$$

$$(\exists_{x\text{: } locatable}(\exists_{l\text{: } location}(\text{at}(x, l))) \rightarrow \exists_{a\text{: } location}(\exists_{b\text{: } location}(\text{road}(a, b)))) \quad \text{(T-LLM-C11)}$$

$$(\exists_{x\text{: } locatable}(\exists_{l\text{: } location}(\text{at}_g(x, l))) \rightarrow \exists_{a\text{: Cap-Num}}(\text{max}_{capacity}(a))) \quad \text{(T-LLM-C12)}$$

$$(\exists_{x\text{: } locatable}(\exists_{l\text{: } location}(\text{at}_g(x, l))) \rightarrow \exists_{l\text{: } location}(\text{goal}_{loc}(l))) \quad \text{(T-LLM-C13)}$$

$$(\exists_{x\text{: } locatable}(\exists_{lg\text{: } location}(\text{at}_g(x, lg))) \rightarrow \exists_{lg2\text{: } location}(\text{goal}_{loc}(lg2))) \quad \text{(T-LLM-C14)}$$

$$\exists_{a\text{: Cap-Num}}(\exists_{b\text{: Cap-Num}}(\text{cap-pre}(a, b))) \quad \text{(T-LLM-C15)}$$

$$\exists_{a\text{: } location}(\exists_{b\text{: } location}(\text{road}(a, b))) \quad \text{(T-LLM-C16)}$$

$$\exists_{l\text{: } location}(\text{goal}_{loc}(l)) \quad \text{(T-LLM-C17)}$$

$$\exists_{l\text{: } location}(\text{has}_{incoming}(l)) \quad \text{(T-LLM-C18)}$$

$$\exists_{l\text{: } location}(\text{has}_{outgoing}(l)) \quad \text{(T-LLM-C19)}$$

$$\exists_{p\text{: } package}(\exists_{l\text{: } location}(\text{at}(p, l)) \vee \exists_{v\text{: } vehicle}(\text{in}(p, v))) \quad \text{(T-LLM-C20)}$$

$$\exists_{p:\, package}(\text{pkg}_{is\_g}(p)) \quad \text{(T-LLM-C21)}$$

$$\exists_{s:\, \text{Cap-Num}}(\max_{capacity}(s)) \quad \text{(T-LLM-C22)}$$

$$\exists_{s:\, \text{Cap-Num}}(\min_{capacity}(s)) \quad \text{(T-LLM-C23)}$$

$$\exists_{v:\, vehicle}(\exists_{l:\, location}(\text{goal}_{loc}(l) \wedge \text{veh}_{can\_reach\_g}(v,l))) \quad \text{(T-LLM-C24)}$$

$$\exists_{v:\, vehicle}(\exists_{l:\, location}(\text{veh}_{can\_reach\_g}(v,l))) \quad \text{(T-LLM-C25)}$$

$$\exists_{v:\, vehicle}(\exists_{s:\, \text{Cap-Num}}(\text{capacity}(v,s))) \quad \text{(T-LLM-C26)}$$

$$\exists_{x:\, locatable}(\exists_{l:\, location}(\text{at}(x,l))) \quad \text{(T-LLM-C27)}$$

$$\exists_{x:\, locatable}(\exists_{l:\, location}(\text{at}_g(x,l))) \quad \text{(T-LLM-C28)}$$

$$\forall_{a:\, \text{Cap-Num}}(\forall_{b:\, \text{Cap-Num}}((\text{cap}_{reach}(a,b) \wedge \text{cap}_{reach}(b,a) \rightarrow \neg(a \neq b)))) \quad \text{(T-LLM-C29)}$$

$$\forall_{a:\, \text{Cap-Num}}(\forall_{b:\, \text{Cap-Num}}((\max_{capacity}(a) \wedge \max_{capacity}(b) \rightarrow \neg(a \neq b)))) \quad \text{(T-LLM-C30)}$$

$$\forall_{a:\, \text{Cap-Num}}(\forall_{b:\, \text{Cap-Num}}((\min_{capacity}(a) \wedge \min_{capacity}(b) \rightarrow \neg(a \neq b)))) \quad \text{(T-LLM-C31)}$$

$$\forall_{a:\, \text{Cap-Num}}(\forall_{b:\, \text{Cap-Num}}(\forall_{c:\, \text{Cap-Num}}((\text{cap}_{reach}(a,b) \wedge \text{cap}_{reach}(b,c) \rightarrow \text{cap}_{reach}(a,c))))) \quad \text{(T-LLM-C32)}$$

$$\forall_{a:\, \text{Cap-Num}}(\forall_{b:\, \text{Cap-Num}}(\neg(\textsf{cap-pre}_g(a,b)))) \quad \text{(T-LLM-C33)}$$

$$\forall_{a:\, \text{Cap-Num}}(\forall_{b:\, \text{Cap-Num}}((\textsf{cap-pre}(a,b) \rightarrow \text{cap}_{reach}(a,b)))) \quad \text{(T-LLM-C34)}$$

$$\forall_{a:\, location}(\forall_{b:\, location}((\text{road}(a,b) \wedge \text{road}(b,a) \rightarrow \text{same}_{comp}(a,b)))) \quad \text{(T-LLM-C35)}$$

$$\forall_{a:\, location}(\forall_{b:\, location}(\forall_{c:\, location}((\text{reach}(a,b) \wedge \text{reach}(b,c) \rightarrow \text{reach}(a,c))))) \quad \text{(T-LLM-C36)}$$

$$\forall_{a:\, location}(\forall_{b:\, location}(\forall_{c:\, location}((\text{road}(a,b) \wedge \text{road}(b,c) \rightarrow \text{reach}(a,c))))) \quad \text{(T-LLM-C37)}$$

$$\forall_{a:\, location}(\forall_{b:\, location}((\text{road}(a,b) \rightarrow \text{reach}(a,b)))) \quad \text{(T-LLM-C38)}$$

$$\forall_{a:\, location}(\forall_{b:\, location}((\text{road}(a,b) \rightarrow \text{road}(b,a)))) \quad \text{(T-LLM-C39)}$$

$$\forall_{l1:\, location}(\forall_{l2:\, location}((\text{reach}(l1,l2) \wedge \text{reach}(l2,l1) \rightarrow \text{reach}(l1,l1)))) \quad \text{(T-LLM-C40)}$$

$$\forall_{l1:\, location}(\forall_{l2:\, location}((\text{reach}(l1,l2) \wedge \text{reach}(l2,l1) \rightarrow \text{reach}(l2,l2)))) \quad \text{(T-LLM-C41)}$$

$$\forall_{l1:\, location}(\forall_{l2:\, location}((\text{reach}(l1,l2) \wedge \text{reach}(l2,l1) \rightarrow \text{same}_{comp}(l1,l2)))) \quad \text{(T-LLM-C42)}$$

$$\forall_{l1:\, location}(\forall_{l2:\, location}((\text{road}(l1,l2) \wedge \text{has}_{outgoing}(l1) \rightarrow \text{has}_{incoming}(l2)))) \quad \text{(T-LLM-C43)}$$

$$\forall_{l:\, location}(\neg(\text{road}(l,l))) \quad \text{(T-LLM-C44)}$$

$$\forall_{l:\, location}((\text{goal}_{loc}(l) \rightarrow \exists_{p:\, package}(\text{at}_g(p,l)))) \quad \text{(T-LLM-C45)}$$

$$\forall_{l:\, location}((\text{goal}_{loc}(l) \rightarrow \exists_{v:\, vehicle}(\text{veh}_{can\_reach\_g}(v,l)))) \quad \text{(T-LLM-C46)}$$

$$\forall_{l:\, location}((\text{goal}_{loc}(l) \rightarrow \text{has}_{incoming}(l))) \quad \text{(T-LLM-C47)}$$

$$\forall_{l:\, location}((\text{goal}_{loc}(l) \rightarrow \text{has}_{outgoing}(l))) \quad \text{(T-LLM-C48)}$$

$$\forall_{l:\, location}((\text{has}_{incoming}(l) \rightarrow \exists_{m:\, location}(\text{road}(m,l)))) \quad \text{(T-LLM-C49)}$$

$$\forall_{l:\, location}((\text{has}_{outgoing}(l) \rightarrow \exists_{m:\, location}(\text{road}(l,m)))) \quad \text{(T-LLM-C50)}$$

$$\forall_{l:\, location}(\text{reach}(l,l)) \quad \text{(T-LLM-C51)}$$

$$\forall_{p:\, package}(\exists_{l:\, location}(\text{at}(p,l)) \vee \exists_{v:\, vehicle}(\text{in}(p,v))) \quad \text{(T-LLM-C52)}$$

$$\forall_{p:\, package}((\exists_{lg:\, location}(\text{at}_g(p,lg)) \rightarrow \text{pkg}_{is\_g}(p))) \quad \text{(T-LLM-C53)}$$

$$\forall_{p:\, package}(\forall_{l0:\, location}(\forall_{lg:\, location}((\mathrm{at}(p, l0) \wedge \mathrm{at}_g(p, lg) \rightarrow \mathrm{reach}(l0, lg)))))) \quad \text{(T-LLM-C54)}$$

$$\forall_{p:\, package}(\forall_{l0:\, location}(\forall_{lg:\, location}((\mathrm{at}(p, l0) \wedge \mathrm{at}_g(p, lg) \rightarrow \mathrm{same}_{comp}(l0, lg)))))) \quad \text{(T-LLM-C55)}$$

$$\forall_{p:\, package}(\forall_{l:\, location}(\forall_{v:\, vehicle}(\neg(\mathrm{at}(p, l) \wedge \mathrm{in}(p, v)))))) \quad \text{(T-LLM-C56)}$$

$$\forall_{p:\, package}(\forall_{l:\, location}(\forall_{v:\, vehicle}((\mathrm{at}(p, l) \rightarrow \neg(\mathrm{in}(p, v))))))) \quad \text{(T-LLM-C57)}$$

$$\forall_{p:\, package}(\forall_{l:\, location}((\mathrm{at}(p, l) \rightarrow \mathrm{reach}(l, l)))) \quad \text{(T-LLM-C58)}$$

$$\forall_{p:\, package}(\forall_{lg:\, location}((\mathrm{at}_g(p, lg) \rightarrow \exists_{a:\, location}(\exists_{b:\, location}(\mathrm{road}(a, b))))))) \quad \text{(T-LLM-C59)}$$

$$\forall_{p:\, package}(\forall_{v1:\, vehicle}(\forall_{v2:\, vehicle}((\mathrm{in}(p, v1) \wedge \mathrm{in}(p, v2) \rightarrow \neg(v1 \neq v2)))))) \quad \text{(T-LLM-C60)}$$

$$\forall_{p:\, package}(\forall_{v:\, vehicle}(\forall_{l:\, location}((\mathrm{in}(p, v) \rightarrow \neg(\mathrm{at}(p, l))))))) \quad \text{(T-LLM-C61)}$$

$$\forall_{p:\, package}(\forall_{v:\, vehicle}(\forall_{lg:\, location}((\mathrm{at}_g(p, lg) \rightarrow \neg(\mathrm{in}(p, v))))))) \quad \text{(T-LLM-C62)}$$

$$\forall_{p:\, package}(\forall_{v:\, vehicle}(\neg(\mathrm{in}(p, v)))) \quad \text{(T-LLM-C63)}$$

$$\forall_{p:\, package}(\forall_{v:\, vehicle}(\neg(\mathrm{in}_g(p, v)))) \quad \text{(T-LLM-C64)}$$

$$\forall_{p:\, package}(\forall_{v:\, vehicle}((\mathrm{in}(p, v) \rightarrow \exists_{l:\, location}(\mathrm{at}(v, l)))))) \quad \text{(T-LLM-C65)}$$

$$\forall_{p:\, package}(\forall_{v:\, vehicle}((\mathrm{in}(p, v) \rightarrow \exists_{s:\, \text{Cap-Num}}(\mathrm{capacity}(v, s)))))) \quad \text{(T-LLM-C66)}$$

$$\forall_{p:\, package}((\mathrm{pkg}_{is\_g}(p) \rightarrow \exists_{l0:\, location}(\mathrm{at}(p, l0)) \vee \exists_{v:\, vehicle}(\mathrm{in}(p, v)))) \quad \text{(T-LLM-C67)}$$

$$\forall_{p:\, package}((\mathrm{pkg}_{is\_g}(p) \rightarrow \exists_{l:\, location}(\mathrm{at}_g(p, l) \wedge \mathrm{has}_{incoming}(l)))) \quad \text{(T-LLM-C68)}$$

$$\forall_{p:\, package}((\mathrm{pkg}_{is\_g}(p) \rightarrow \exists_{l:\, location}(\mathrm{at}_g(p, l) \wedge \mathrm{has}_{outgoing}(l)))) \quad \text{(T-LLM-C69)}$$

$$\forall_{s0:\, \text{Cap-Num}}((\mathrm{min}_{capacity}(s0) \rightarrow \exists_{sM:\, \text{Cap-Num}}(\mathrm{max}_{capacity}(sM) \wedge \mathrm{cap}_{reach}(s0, sM)))) \quad \text{(T-LLM-C70)}$$

$$\forall_{s1:\, \text{Cap-Num}}(\forall_{a:\, \text{Cap-Num}}(\forall_{b:\, \text{Cap-Num}}((\text{cap-pre}(s1, a) \wedge \text{cap-pre}(s1, b) \rightarrow \neg(a \neq b)))))) \quad \text{(T-LLM-C71)}$$

$$\forall_{s1:\, \text{Cap-Num}}(\forall_{s2:\, \text{Cap-Num}}((\text{cap-pre}(s1, s2) \rightarrow \neg(\text{cap-pre}(s2, s1)))))) \quad \text{(T-LLM-C72)}$$

$$\forall_{s2:\, \text{Cap-Num}}(\forall_{a:\, \text{Cap-Num}}(\forall_{b:\, \text{Cap-Num}}((\text{cap-pre}(a, s2) \wedge \text{cap-pre}(b, s2) \rightarrow \neg(a \neq b)))))) \quad \text{(T-LLM-C73)}$$

$$\forall_{s:\, \text{Cap-Num}}(\neg(\text{cap-pre}(s, s))) \quad \text{(T-LLM-C74)}$$

$$\forall_{s:\, \text{Cap-Num}}((\mathrm{max}_{capacity}(s) \rightarrow \forall_{t:\, \text{Cap-Num}}(\neg(\text{cap-pre}(s, t)))))) \quad \text{(T-LLM-C75)}$$

$$\forall_{s:\, \text{Cap-Num}}((\mathrm{min}_{capacity}(s) \rightarrow \forall_{t:\, \text{Cap-Num}}(\neg(\text{cap-pre}(t, s)))))) \quad \text{(T-LLM-C76)}$$

$$\forall_{s:\, \text{Cap-Num}}(\mathrm{cap}_{reach}(s, s)) \quad \text{(T-LLM-C77)}$$

$$\forall_{sM:\, \text{Cap-Num}}((\mathrm{max}_{capacity}(sM) \rightarrow \exists_{s0:\, \text{Cap-Num}}(\mathrm{min}_{capacity}(s0) \wedge \mathrm{cap}_{reach}(s0, sM)))) \quad \text{(T-LLM-C78)}$$

$$\forall_{u:\, location}(\forall_{v:\, location}((\mathrm{same}_{comp}(u, v) \rightarrow \mathrm{reach}(u, v) \wedge \mathrm{reach}(v, u)))) \quad \text{(T-LLM-C79)}$$

$$\forall_{u:\, location}(\forall_{v:\, location}((\mathrm{same}_{comp}(u, v) \rightarrow \mathrm{same}_{comp}(v, u)))) \quad \text{(T-LLM-C80)}$$

$$\forall_{u:\, location}(\mathrm{same}_{comp}(u, u)) \quad \text{(T-LLM-C81)}$$

$$\forall_{v:\, vehicle}(\exists_{l:\, location}(\mathrm{at}(v, l))) \quad \text{(T-LLM-C82)}$$

$$\forall_{v:\, vehicle}(\exists_{s:\, \text{Cap-Num}}(\mathrm{capacity}(v, s))) \quad \text{(T-LLM-C83)}$$

$$\forall_{v:\, vehicle}(\forall_{l1:\, location}(\forall_{l2:\, location}((\mathrm{at}(v, l1) \wedge \mathrm{at}(v, l2) \rightarrow \neg(l1 \neq l2)))))) \quad \text{(T-LLM-C84)}$$

$$\forall_{v:\, vehicle}(\forall_{l:\, location}(\neg(\mathrm{at}_g(v, l)))) \quad \text{(T-LLM-C85)}$$

$$\forall_{v:\, vehicle}\Big(\forall_{l:\, location}((\mathrm{at}(v,l) \rightarrow \exists_{s:\, \text{Cap-Num}}(\mathrm{capacity}(v,s))))\Big) \quad \text{(T-LLM-C86)}$$

$$\forall_{v:\, vehicle}\Big(\forall_{l:\, location}((\mathrm{at}(v,l) \rightarrow \mathrm{has}_{incoming}(l))))\Big) \quad \text{(T-LLM-C87)}$$

$$\forall_{v:\, vehicle}\Big(\forall_{l:\, location}((\mathrm{veh}_{can\_reach\_g}(v,l) \rightarrow \mathrm{reach}(l,l))))\Big) \quad \text{(T-LLM-C88)}$$

$$\forall_{v:\, vehicle}\Big(\forall_{s1:\, \text{Cap-Num}}\Big(\forall_{s2:\, \text{Cap-Num}}((\mathrm{capacity}(v,s1) \wedge \mathrm{capacity}(v,s2) \rightarrow \neg(s1 \neq s2)))\Big)\Big) \quad \text{(T-LLM-C89)}$$

$$\forall_{v:\, vehicle}\Big(\forall_{s:\, \text{Cap-Num}}((\mathrm{capacity}(v,s) \rightarrow \exists_{l:\, location}(\mathrm{at}(v,l))))\Big) \quad \text{(T-LLM-C90)}$$

$$\forall_{v:\, vehicle}\Big(\forall_{s:\, \text{Cap-Num}}((\mathrm{capacity}(v,s) \rightarrow \exists_{s0:\, \text{Cap-Num}}(\mathrm{min}_{capacity}(s0) \wedge \mathrm{cap}_{reach}(s0,s))))\Big) \quad \text{(T-LLM-C91)}$$

$$\forall_{v:\, vehicle}\Big(\forall_{s:\, \text{Cap-Num}}((\mathrm{capacity}(v,s) \rightarrow \exists_{sM:\, \text{Cap-Num}}(\mathrm{max}_{capacity}(sM) \wedge \mathrm{cap}_{reach}(s,sM))))\Big) \quad \text{(T-LLM-C92)}$$

$$\forall_{vv:\, vehicle}\Big(\forall_{s:\, \text{Cap-Num}}(\neg(\mathrm{capacity}_g(vv,s)))\Big) \quad \text{(T-LLM-C93)}$$

$$\forall_{x:\, locatable}\Big(\forall_{l1:\, location}\Big(\forall_{l2:\, location}((\mathrm{at}(x,l1) \wedge \mathrm{at}(x,l2) \rightarrow \neg(l1 \neq l2)))\Big)\Big) \quad \text{(T-LLM-C94)}$$

$$\forall_{x:\, locatable}\Big(\forall_{l1:\, location}\Big(\forall_{l2:\, location}((\mathrm{at}_g(x,l1) \wedge \mathrm{at}_g(x,l2) \rightarrow \neg(l1 \neq l2)))\Big)\Big) \quad \text{(T-LLM-C95)}$$

$$\forall_{x:\, locatable}\Big(\forall_{l:\, location}((\mathrm{at}(x,l) \rightarrow \mathrm{has}_{outgoing}(l)))\Big) \quad \text{(T-LLM-C96)}$$

$$\forall_{x:\, locatable}\Big(\forall_{l:\, location}((\mathrm{at}_g(x,l) \rightarrow \exists_{p:\, package}(\neg(x \neq p))))\Big) \quad \text{(T-LLM-C97)}$$

$$\forall_{x:\, locatable}\Big(\forall_{lg:\, location}((\mathrm{at}_g(x,lg) \rightarrow \mathrm{goal}_{loc}(lg)))\Big) \quad \text{(T-LLM-C98)}$$

$$\neg\Big(\exists_{l1:\, location}(\mathrm{road}_g(l1,l1))\Big) \quad \text{(T-LLM-C99)}$$

$$\mathrm{cap}_{reach}(s_{dummy}, s_{dummy}) \quad \text{(T-LLM-C100)}$$

With LLM-generated descriptions:

- T-LLM-C1: If any road exists, then some location has-outgoing (consistency between predicates).
- T-LLM-C2: Every location with an initial package is reachable by some vehicle (candidate).
- T-LLM-C3: If there exists a goal location, then at least one package is in the goals.
- T-LLM-C4: If there is at least one package initially, then there is at least one vehicle (as in datasets).
- T-LLM-C5: Goals imply there is at least one road (redundant but safe for these instances).
- T-LLM-C6: If there are goal packages, there is at least one package present initially (candidate).
- T-LLM-C7: If there are goal packages, there exists a vehicle at some location (trivial reach to itself).
- T-LLM-C8: If there are goal packages, then some vehicle can reach some location (trivially true).
- T-LLM-C9: If there are goal packages, then there is at least one vehicle available.
- T-LLM-C10: If there is any vehicle, then some object is placed at a location initially (vehicles are placed).
- T-LLM-C11: If any object is placed, the instance contains at least one road (as in provided datasets).
- T-LLM-C12: If there are at-goal facts, a max capacity exists (as in provided instances).
- T-LLM-C13: If there are at-goal facts, there exists at least one goal-loc.
- T-LLM-C14: Presence of at-goal implies there exists a goal-loc (consistency).
- T-LLM-C15: There is at least one link in the capacity chain.
- T-LLM-C16: There is at least one directed road in the map.
- T-LLM-C17: There exists a location that is a goal destination.
- T-LLM-C18: At least one location has an incoming road.
- T-LLM-C19: At least one location has an outgoing road.
- T-LLM-C20: There is at least one package initially present (either placed or loaded).
- T-LLM-C21: There is at least one package mentioned in the goals.
- T-LLM-C22: A maximal capacity value exists in the problem instance.
- T-LLM-C23: A minimal capacity value exists in the problem instance.

- T-LLM-C24: Some vehicle can reach at least one goal location initially (candidate).
- T-LLM-C25: Each vehicle can reach at least one location (e.g., its own start) via reach.
- T-LLM-C26: There is at least one vehicle (has a capacity fact).
- T-LLM-C27: There is at least one object placed at some location initially.
- T-LLM-C28: There is at least one goal location specified.
- T-LLM-C29: Capacity reachability is antisymmetric (no distinct values reach each other both ways).
- T-LLM-C30: There is a unique maximal capacity (up to equality).
- T-LLM-C31: There is a unique minimal capacity (up to equality).
- T-LLM-C32: Capacity reachability is transitive.
- T-LLM-C33: No cap-pre-goal facts are used in these problems.
- T-LLM-C34: cap-pre implies capacity reachability.
- T-LLM-C35: Mutual road connections imply being in the same component.
- T-LLM-C36: Reachability is transitive.
- T-LLM-C37: Two-step road connectivity implies reachability along that path.
- T-LLM-C38: Any direct road implies reachability along that road.
- T-LLM-C39: Roads are typically bidirectional (candidate symmetry constraint).
- T-LLM-C40: Mutual reachability implies reflexivity (sanity check for reach).
- T-LLM-C41: Mutual reachability also implies reflexivity at the other endpoint.
- T-LLM-C42: Mutual reachability implies being in the same component (consistency with definition).
- T-LLM-C43: Road endpoints have corresponding outgoing/incoming properties.
- T-LLM-C44: The road relation has no self-loops in valid instances.
- T-LLM-C45: Definition
- consistency: goal-loc implies some at-goal to that location.
- T-LLM-C46: Every goal location is reachable by at least one vehicle initially (candidate).
- T-LLM-C47: Every goal location has at least one incoming road.
- T-LLM-C48: Every goal location has at least one outgoing road.
- T-LLM-C49: has-incoming implies there exists a road into the location (definition sanity).
- T-LLM-C50: has-outgoing implies there exists a road out of the location (definition sanity).
- T-LLM-C51: Reachability is reflexive (every location reaches itself).
- T-LLM-C52: Every package is either at some location or inside some vehicle initially.
- T-LLM-C53: If a package appears in at-goal, then pkg-is-goal holds for it (definition alignment).
- T-LLM-C54: Each goal package's destination is reachable from its initial location.
- T-LLM-C55: Each goal package's start and destination lie in the same strongly connected component (candidate).
- T-LLM-C56: A package cannot be simultaneously at a location and inside a vehicle initially.
- T-LLM-C57: If a package is at some location initially, it is not inside any vehicle initially.
- T-LLM-C58: Each package's initial location is trivially reachable from itself (sanity of reach).
- T-LLM-C59: Presence of goals implies the instance has at least one road (candidate).
- T-LLM-C60: Each package is in at most one vehicle initially.
- T-LLM-C61: If a package is in a vehicle initially, it is not at any location initially.
- T-LLM-C62: Goal packages are not initially inside vehicles (as in provided instances).
- T-LLM-C63: Initially, no package is inside any vehicle (typical for these problems).
- T-LLM-C64: No in-goal facts are used in these problems.
- T-LLM-C65: If a package is initially in a vehicle, that vehicle has a location initially.
- T-LLM-C66: Any vehicle that carries a package has a capacity fact initially.
- T-LLM-C67: Every goal package exists initially either at a location or in a vehicle.
- T-LLM-C68: Every goal package's destination has at least one incoming road.
- T-LLM-C69: Every goal package's destination has at least one outgoing road.
- T-LLM-C70: From the min capacity you can reach some max capacity along the chain (candidate).
- T-LLM-C71: A capacity level has at most one distinct successor (right-unique).

- T-LLM-C72: The cap-pre relation is antisymmetric (no immediate two-cycles).
- T-LLM-C73: A capacity level has at most one distinct predecessor (left-unique).
- T-LLM-C74: The cap-pre relation is irreflexive.
- T-LLM-C75: Max capacity has no outgoing cap-pre edge.
- T-LLM-C76: Min capacity has no incoming cap-pre edge.
- T-LLM-C77: Capacity reachability is reflexive.
- T-LLM-C78: Every max capacity is reachable from some min capacity (candidate).
- T-LLM-C79: same-comp implies mutual reachability.
- T-LLM-C80: same-comp is symmetric.
- T-LLM-C81: Every location is in the same component as itself.
- T-LLM-C82: Every vehicle has some initial location (is placed somewhere).
- T-LLM-C83: Every vehicle has some initial capacity level fact.
- T-LLM-C84: Each vehicle is at most at one location initially.
- T-LLM-C85: Vehicles do not appear in at-goal facts.
- T-LLM-C86: Any placed vehicle has a capacity fact initially (as in provided instances).
- T-LLM-C87: Any location that initially hosts a vehicle has at least one incoming road.
- T-LLM-C88: Any location reachable by a vehicle is trivially reachable from itself (sanity of reach).
- T-LLM-C89: Each vehicle has at most one capacity fact initially.
- T-LLM-C90: Any vehicle with a capacity fact is placed at some location initially (as in provided instances).
- T-LLM-C91: Any vehicle capacity is above or equal to some min capacity along the chain.
- T-LLM-C92: Any vehicle capacity is below or equal to some max capacity along the chain.
- T-LLM-C93: No capacity-goal facts are used in these problems.
- T-LLM-C94: Each locatable is at most at one location initially.
- T-LLM-C95: Each locatable appears at most at one goal location.
- T-LLM-C96: Any location that initially hosts an object has at least one outgoing road.
- T-LLM-C97: Only packages appear in at-goal facts.
- T-LLM-C98: Any location appearing in at-goal is a goal-loc (definition sanity).
- T-LLM-C99: No road-goal facts are used (none appear in goals).
- T-LLM-C100: Placeholder removed by next constraints (not used).

Which gives the reduced set:

$$\left(\exists_{p:\, package}\left(\mathrm{pkg}_{is\_g}(p)\right) \rightarrow \exists_{p2:\, package}\left(\exists_{l:\, location}\left(\mathrm{at}(p2,l)\right) \vee \exists_{v:\, vehicle}\left(\mathrm{in}(p2,v)\right)\right)\right) \tag{T-LLM-1}$$

$$\exists_{a:\, \text{Cap-Num}}\left(\exists_{b:\, \text{Cap-Num}}\left(\text{cap-pre}(a,b)\right)\right) \tag{T-LLM-2}$$

$$\exists_{p:\, package}\left(\exists_{l:\, location}\left(\mathrm{at}(p,l)\right) \vee \exists_{v:\, vehicle}\left(\mathrm{in}(p,v)\right)\right) \tag{T-LLM-3}$$

$$\exists_{x:\, locatable}\left(\exists_{l:\, location}\left(\mathrm{at}_g(x,l)\right)\right) \tag{T-LLM-4}$$

$$\forall_{a:\, \text{Cap-Num}}\left(\forall_{b:\, \text{Cap-Num}}\left(\left(\mathrm{cap}_{reach}(a,b) \wedge \mathrm{cap}_{reach}(b,a) \rightarrow \neg(a \neq b)\right)\right)\right) \tag{T-LLM-5}$$

$$\forall_{a:\, \text{Cap-Num}}\left(\forall_{b:\, \text{Cap-Num}}\left(\left(\max_{capacity}(a) \wedge \max_{capacity}(b) \rightarrow \neg(a \neq b)\right)\right)\right) \tag{T-LLM-6}$$

$$\forall_{a:\, \text{Cap-Num}}\left(\forall_{b:\, \text{Cap-Num}}\left(\left(\min_{capacity}(a) \wedge \min_{capacity}(b) \rightarrow \neg(a \neq b)\right)\right)\right) \tag{T-LLM-7}$$

$$\forall_{a:\, \text{Cap-Num}}\left(\forall_{b:\, \text{Cap-Num}}\left(\forall_{c:\, \text{Cap-Num}}\left(\left(\mathrm{cap}_{reach}(a,b) \wedge \mathrm{cap}_{reach}(b,c) \rightarrow \mathrm{cap}_{reach}(a,c)\right)\right)\right)\right) \tag{T-LLM-8}$$

$$\forall_{a:\, \text{Cap-Num}}\left(\forall_{b:\, \text{Cap-Num}}\left(\neg(\text{cap-pre}_g(a,b))\right)\right) \tag{T-LLM-9}$$

$$\forall_{a:\, location}\left(\forall_{b:\, location}\left(\forall_{c:\, location}\left(\left(\mathrm{reach}(a,b) \wedge \mathrm{reach}(b,c) \rightarrow \mathrm{reach}(a,c)\right)\right)\right)\right) \tag{T-LLM-10}$$

$$\forall_{a:\, location}\left(\forall_{b:\, location}\left(\left(\mathrm{road}(a,b) \rightarrow \mathrm{road}(b,a)\right)\right)\right) \tag{T-LLM-11}$$

$$\forall_{l:\, location}(\neg(\mathrm{road}(l,l))) \quad \text{(T-LLM-12)}$$

$$\forall_{l:\, location}((\mathrm{goal}_{loc}(l) \rightarrow \exists_{v:\, vehicle}(\mathrm{veh}_{can\_reach\_g}(v,l)))) \quad \text{(T-LLM-13)}$$

$$\forall_{p:\, package}(\exists_{l:\, location}(\mathrm{at}(p,l)) \vee \exists_{v:\, vehicle}(\mathrm{in}(p,v))) \quad \text{(T-LLM-14)}$$

$$\forall_{p:\, package}(\forall_{l0:\, location}(\forall_{lg:\, location}((\mathrm{at}(p,l0) \wedge \mathrm{at}_g(p,lg) \rightarrow \mathrm{same}_{comp}(l0,lg))))) \quad \text{(T-LLM-15)}$$

$$\forall_{p:\, package}(\forall_{v:\, vehicle}(\neg(\mathrm{in}(p,v)))) \quad \text{(T-LLM-16)}$$

$$\forall_{p:\, package}(\forall_{v:\, vehicle}(\neg(\mathrm{in}_g(p,v)))) \quad \text{(T-LLM-17)}$$

$$\forall_{p:\, package}((\mathrm{pkg}_{is\_g}(p) \rightarrow \exists_{l:\, location}(\mathrm{at}_g(p,l) \wedge \mathrm{has}_{outgoing}(l)))) \quad \text{(T-LLM-18)}$$

$$\forall_{s1:\, \text{Cap-Num}}(\forall_{a:\, \text{Cap-Num}}(\forall_{b:\, \text{Cap-Num}}((\text{cap-pre}(s1,a) \wedge \text{cap-pre}(s1,b) \rightarrow \neg(a \neq b))))) \quad \text{(T-LLM-19)}$$

$$\forall_{s2:\, \text{Cap-Num}}(\forall_{a:\, \text{Cap-Num}}(\forall_{b:\, \text{Cap-Num}}((\text{cap-pre}(a,s2) \wedge \text{cap-pre}(b,s2) \rightarrow \neg(a \neq b))))) \quad \text{(T-LLM-20)}$$

$$\forall_{s:\, \text{Cap-Num}}(\neg(\text{cap-pre}(s,s))) \quad \text{(T-LLM-21)}$$

$$\forall_{v:\, vehicle}(\exists_{s:\, \text{Cap-Num}}(\mathrm{capacity}(v,s))) \quad \text{(T-LLM-22)}$$

$$\forall_{v:\, vehicle}(\forall_{s1:\, \text{Cap-Num}}(\forall_{s2:\, \text{Cap-Num}}((\mathrm{capacity}(v,s1) \wedge \mathrm{capacity}(v,s2) \rightarrow \neg(s1 \neq s2))))) \quad \text{(T-LLM-23)}$$

$$\forall_{v:\, vehicle}(\forall_{s:\, \text{Cap-Num}}((\mathrm{capacity}(v,s) \rightarrow \exists_{l:\, location}(\mathrm{at}(v,l))))) \quad \text{(T-LLM-24)}$$

$$\forall_{v:\, vehicle}(\forall_{s:\, \text{Cap-Num}}((\mathrm{capacity}(v,s) \rightarrow \exists_{s0:\, \text{Cap-Num}}(\mathrm{min}_{capacity}(s0) \wedge \mathrm{cap}_{reach}(s0,s))))) \quad \text{(T-LLM-25)}$$

$$\forall_{v:\, vehicle}(\forall_{s:\, \text{Cap-Num}}((\mathrm{capacity}(v,s) \rightarrow \exists_{sM:\, \text{Cap-Num}}(\mathrm{max}_{capacity}(sM) \wedge \mathrm{cap}_{reach}(s,sM))))) \quad \text{(T-LLM-26)}$$

$$\forall_{vv:\, vehicle}(\forall_{s:\, \text{Cap-Num}}(\neg(\mathrm{capacity}_g(vv,s)))) \quad \text{(T-LLM-27)}$$

$$\forall_{x:\, locatable}(\forall_{l1:\, location}(\forall_{l2:\, location}((\mathrm{at}(x,l1) \wedge \mathrm{at}(x,l2) \rightarrow \neg(l1 \neq l2))))) \quad \text{(T-LLM-28)}$$

$$\forall_{x:\, locatable}(\forall_{l1:\, location}(\forall_{l2:\, location}((\mathrm{at}_g(x,l1) \wedge \mathrm{at}_g(x,l2) \rightarrow \neg(l1 \neq l2))))) \quad \text{(T-LLM-29)}$$

$$\forall_{x:\, locatable}(\forall_{l:\, location}((\mathrm{at}(x,l) \rightarrow \mathrm{has}_{outgoing}(l)))) \quad \text{(T-LLM-30)}$$

$$\forall_{x:\, locatable}(\forall_{l:\, location}((\mathrm{at}_g(x,l) \rightarrow \exists_{p:\, package}(\neg(x \neq p))))) \quad \text{(T-LLM-31)}$$

$$\neg(\exists_{l1:\, location}(\mathrm{road}_g(l1,l1))) \quad \text{(T-LLM-32)}$$

With LLM-generated descriptions:

- T-LLM-1: If there are goal packages, there is at least one package present initially (candidate).
- T-LLM-2: There is at least one link in the capacity chain.
- T-LLM-3: There is at least one package initially present (either placed or loaded).
- T-LLM-4: There is at least one goal location specified.
- T-LLM-5: Capacity reachability is antisymmetric (no distinct values reach each other both ways).
- T-LLM-6: There is a unique maximal capacity (up to equality).
- T-LLM-7: There is a unique minimal capacity (up to equality).
- T-LLM-8: Capacity reachability is transitive.
- T-LLM-9: No cap-pre-goal facts are used in these problems.
- T-LLM-10: Reachability is transitive.
- T-LLM-11: Roads are typically bidirectional (candidate symmetry constraint).
- T-LLM-12: The road relation has no self-loops in valid instances.
- T-LLM-13: Every goal location is reachable by at least one vehicle initially (candidate).

- T-LLM-14: Every package is either at some location or inside some vehicle initially.
- T-LLM-15: Each goal package's start and destination lie in the same strongly connected component (candidate).
- T-LLM-16: Initially, no package is inside any vehicle (typical for these problems).
- T-LLM-17: No in-goal facts are used in these problems.
- T-LLM-18: Every goal package's destination has at least one outgoing road.
- T-LLM-19: A capacity level has at most one distinct successor (right-unique).
- T-LLM-20: A capacity level has at most one distinct predecessor (left-unique).
- T-LLM-21: The cap-pre relation is irreflexive.
- T-LLM-22: Every vehicle has some initial capacity level fact.
- T-LLM-23: Each vehicle has at most one capacity fact initially.
- T-LLM-24: Any vehicle with a capacity fact is placed at some location initially (as in provided instances).
- T-LLM-25: Any vehicle capacity is above or equal to some min capacity along the chain.
- T-LLM-26: Any vehicle capacity is below or equal to some max capacity along the chain.
- T-LLM-27: No capacity-goal facts are used in these problems.
- T-LLM-28: Each locatable is at most at one location initially.
- T-LLM-29: Each locatable appears at most at one goal location.
- T-LLM-30: Any location that initially hosts an object has at least one outgoing road.
- T-LLM-31: Only packages appear in at-goal facts.
- T-LLM-32: No road-goal facts are used (none appear in goals).

Floortile

Candidate constraints:

$$\text{adjacent}(x\colon tile,\ y\colon tile) \coloneqq \text{up}(x,y) \lor \text{down}(x,y) \lor \text{left}(x,y) \lor \text{right}(x,y)$$

$$\text{connected}(x\colon tile,\ y\colon tile) \coloneqq \text{adjacent}(x,y) \lor \exists_{z\colon tile}(\text{adjacent}(x,z) \land \text{connected}(z,y))$$

$$\text{free}_{tile}(x\colon tile) \coloneqq \text{clear}(x) \land \neg(\text{occupied}(x))$$

$$\text{goal}_{color}(c\colon color) \coloneqq \exists_{x\colon tile}(\text{painted}_g(x,c))$$

$$\text{goal}_{tile}(x\colon tile) \coloneqq \exists_{c\colon color}(\text{painted}_g(x,c))$$

$$\text{has}_{neighbor}(x\colon tile) \coloneqq \exists_{y\colon tile}(\text{adjacent}(x,y))$$

$$\text{hreach}(x\colon tile,\ y\colon tile) \coloneqq \text{right}(y,x) \lor \text{left}(y,x) \lor \exists_{z\colon tile}(\text{right}(z,x) \land \text{hreach}(z,y)) \lor \exists_{z\colon tile}(\text{left}(z,x) \land \text{hreach}(z,y))$$

$$\text{occupied}(x\colon tile) \coloneqq \exists_{r\colon robot}(\text{robot-at}(r,x))$$

$$\text{robot}_{reachable}(x\colon tile) \coloneqq \exists_{r\colon robot}(\exists_{y\colon tile}(\text{robot-at}(r,y) \land \text{connected}(y,x)))$$

$$\text{vreach}(x\colon tile,\ y\colon tile) \coloneqq \text{up}(y,x) \lor \text{down}(y,x) \lor \exists_{z\colon tile}(\text{up}(z,x) \land \text{vreach}(z,y)) \lor \exists_{z\colon tile}(\text{down}(z,x) \land \text{vreach}(z,y))$$

$$\exists_{c\colon color}(\text{available-color}(c)) \quad \text{(F-LLM-C1)}$$

$$\exists_{r\colon robot}(\exists_{x\colon tile}(\text{robot-at}(r,x))) \quad \text{(F-LLM-C2)}$$

$$\exists_{x\colon tile}(\exists_{c\colon color}(\text{painted}_g(x,c))) \quad \text{(F-LLM-C3)}$$

$$\forall_{c\colon color}((\text{available-color}(c) \rightarrow \exists_{r\colon robot}(\text{robot-has}(r,c)))) \quad \text{(F-LLM-C4)}$$

$$\forall_{c\colon color}((\text{available-color}_g(c) \rightarrow \text{available-color}(c))) \quad \text{(F-LLM-C5)}$$

$$\forall_{c\colon color}((\text{goal}_{color}(c) \rightarrow \exists_{x\colon tile}(\text{painted}_g(x,c)))) \quad \text{(F-LLM-C6)}$$

$$\forall_{c\colon color}((\text{goal}_{color}(c) \rightarrow \text{available-color}(c))) \quad \text{(F-LLM-C7)}$$

$$\forall_{r1\colon robot}(\forall_{r2\colon robot}(\forall_{x\colon tile}((\text{robot-at}(r1,x) \land \text{robot-at}(r2,x) \rightarrow r1 \neq r2)))) \quad \text{(F-LLM-C8)}$$

$$\forall_{r1\colon robot}(\forall_{r2\colon robot}(\forall_{x\colon tile}((\text{robot-at}_g(r1,x) \land \text{robot-at}_g(r2,x) \rightarrow r1 \neq r2)))) \quad \text{(F-LLM-C9)}$$

$$\forall_{r\colon robot}(\exists_{c\colon color}(\text{robot-has}(r,c))) \quad \text{(F-LLM-C10)}$$

$$\forall_{r\colon robot}(\forall_{c\colon color}(\forall_{d\colon color}((\text{robot-has}(r,c) \land \text{robot-has}(r,d) \rightarrow c \neq d)))) \quad \text{(F-LLM-C11)}$$

$$\forall_{r\colon robot}(\forall_{c\colon color}(\forall_{d\colon color}((\text{robot-has}_g(r,c) \land \text{robot-has}_g(r,d) \rightarrow c \neq d)))) \quad \text{(F-LLM-C12)}$$

$$\forall_{r\colon robot}(\forall_{c\colon color}((\text{robot-has}(r,c) \rightarrow \text{available-color}(c)))) \quad \text{(F-LLM-C13)}$$

$$\forall_{r\colon robot}(\forall_{c\colon color}((\text{robot-has}_g(r,c) \rightarrow \text{available-color}(c)))) \quad \text{(F-LLM-C14)}$$

$$\forall_{r\colon robot}(\forall_{c\colon color}((\text{robot-has}_g(r,c) \rightarrow \text{robot-has}(r,c)))) \quad \text{(F-LLM-C15)}$$

$$\forall_{r\colon robot}(\forall_{x\colon tile}(\forall_{c\colon color}((\text{robot-at}_g(r,x) \land \text{painted}_g(x,c) \rightarrow \text{available-color}(c))))) \quad \text{(F-LLM-C16)}$$

$$\forall_{r\colon robot}(\forall_{x\colon tile}(\forall_{y\colon tile}((\text{robot-at}(r,x) \land \text{adjacent}(y,x) \rightarrow \text{clear}(y))))) \quad \text{(F-LLM-C17)}$$

$$\forall_{r\colon robot}(\forall_{x\colon tile}(\forall_{y\colon tile}((\text{robot-at}(r,x) \land \text{down}(y,x) \rightarrow \text{clear}(y))))) \quad \text{(F-LLM-C18)}$$

$$\forall_{r\colon robot}(\forall_{x\colon tile}(\forall_{y\colon tile}((\text{robot-at}(r,x) \land \text{left}(y,x) \rightarrow \text{clear}(y))))) \quad \text{(F-LLM-C19)}$$

$$\forall_{r\colon robot}(\forall_{x\colon tile}(\forall_{y\colon tile}((\text{robot-at}(r,x) \land \text{right}(y,x) \rightarrow \text{clear}(y))))) \quad \text{(F-LLM-C20)}$$

$$\forall_{r:\, robot}\,(\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{robot-at}(r,x) \wedge \text{robot-at}(r,y) \rightarrow x \neq y)))) \quad \text{(F-LLM-C21)}$$

$$\forall_{r:\, robot}\,(\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{robot-at}(r,x) \wedge \text{up}(y,x) \rightarrow \text{clear}(y))))) \quad \text{(F-LLM-C22)}$$

$$\forall_{r:\, robot}\,(\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{robot-at}_g(r,x) \wedge \text{robot-at}_g(r,y) \rightarrow x \neq y)))) \quad \text{(F-LLM-C23)}$$

$$\forall_{r:\, robot}\,(\forall_{x:\, tile}\,(\neg(\text{robot-at}(r,x) \wedge \text{clear}(x)))) \quad \text{(F-LLM-C24)}$$

$$\forall_{r:\, robot}\,(\forall_{x:\, tile}\,(\neg(\text{robot-at}_g(r,x) \wedge \text{clear}_g(x)))) \quad \text{(F-LLM-C25)}$$

$$\forall_{r:\, robot}\,(\forall_{x:\, tile}\,((\text{robot-at}(r,x) \rightarrow \text{has}_{neighbor}(x)))) \quad \text{(F-LLM-C26)}$$

$$\forall_{r:\, robot}\,(\forall_{x:\, tile}\,((\text{robot-at}_g(r,x) \rightarrow \text{robot-at}(r,x)))) \quad \text{(F-LLM-C27)}$$

$$\forall_{r:\, robot}\,((\text{free-color}(r) \rightarrow \forall_{c:\, color}\,(\neg(\text{robot-has}(r,c))))) \quad \text{(F-LLM-C28)}$$

$$\forall_{r:\, robot}\,((\text{free-color}_g(r) \rightarrow \text{free-color}(r))) \quad \text{(F-LLM-C29)}$$

$$\forall_{x:\, tile}\,((\neg(\text{clear}(x)) \rightarrow \exists_{r:\, robot}\,(\text{robot-at}(r,x)))) \quad \text{(F-LLM-C30)}$$

$$\forall_{x:\, tile}\,((\neg(\text{clear}(x)) \rightarrow \text{occupied}(x))) \quad \text{(F-LLM-C31)}$$

$$\forall_{x:\, tile}\,((\text{clear}(x) \wedge \neg(\text{occupied}(x)) \rightarrow \text{free}_{tile}(x))) \quad \text{(F-LLM-C32)}$$

$$\forall_{x:\, tile}\,(\forall_{c:\, color}\,(\forall_{d:\, color}\,((\text{painted}_g(x,c) \wedge \text{painted}_g(x,d) \rightarrow c \neq d)))) \quad \text{(F-LLM-C33)}$$

$$\forall_{x:\, tile}\,(\forall_{c:\, color}\,(\neg(\text{painted}_g(x,c) \wedge \text{clear}_g(x)))) \quad \text{(F-LLM-C34)}$$

$$\forall_{x:\, tile}\,(\forall_{c:\, color}\,((\text{painted}(x,c) \rightarrow \neg(\text{clear}(x))))) \quad \text{(F-LLM-C35)}$$

$$\forall_{x:\, tile}\,(\forall_{c:\, color}\,((\text{painted}_g(x,c) \rightarrow \neg(\text{painted}(x,c))))) \quad \text{(F-LLM-C36)}$$

$$\forall_{x:\, tile}\,(\forall_{c:\, color}\,((\text{painted}_g(x,c) \rightarrow \text{available-color}(c)))) \quad \text{(F-LLM-C37)}$$

$$\forall_{x:\, tile}\,(\forall_{r:\, robot}\,(\forall_{c:\, color}\,(\neg(\text{painted}(x,c) \wedge \text{robot-at}(r,x))))) \quad \text{(F-LLM-C38)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{adjacent}(x,y) \wedge \text{adjacent}(y,x) \rightarrow \text{has}_{neighbor}(x)))) \quad \text{(F-LLM-C39)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{up}(x,y) \vee \text{down}(x,y) \vee \text{left}(x,y) \vee \text{right}(x,y) \rightarrow \text{adjacent}(x,y)))) \quad \text{(F-LLM-C40)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\forall_{z:\, tile}\,((\text{down}(x,y) \wedge \text{down}(x,z) \rightarrow y \neq z)))) \quad \text{(F-LLM-C41)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\forall_{z:\, tile}\,((\text{down}(y,x) \wedge \text{down}(z,x) \rightarrow y \neq z)))) \quad \text{(F-LLM-C42)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\forall_{z:\, tile}\,((\text{left}(x,y) \wedge \text{left}(x,z) \rightarrow y \neq z)))) \quad \text{(F-LLM-C43)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\forall_{z:\, tile}\,((\text{left}(y,x) \wedge \text{left}(z,x) \rightarrow y \neq z)))) \quad \text{(F-LLM-C44)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\forall_{z:\, tile}\,((\text{right}(x,y) \wedge \text{right}(x,z) \rightarrow y \neq z)))) \quad \text{(F-LLM-C45)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\forall_{z:\, tile}\,((\text{right}(y,x) \wedge \text{right}(z,x) \rightarrow y \neq z)))) \quad \text{(F-LLM-C46)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\forall_{z:\, tile}\,((\text{up}(x,y) \wedge \text{up}(x,z) \rightarrow y \neq z)))) \quad \text{(F-LLM-C47)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\forall_{z:\, tile}\,((\text{up}(y,x) \wedge \text{up}(z,x) \rightarrow y \neq z)))) \quad \text{(F-LLM-C48)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\neg(\text{down}(x,y) \wedge \text{down}(y,x)))) \quad \text{(F-LLM-C49)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\neg(\text{left}(x,y) \wedge \text{down}(x,y)))) \quad \text{(F-LLM-C50)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\neg(\text{left}(x,y) \wedge \text{left}(y,x)))) \quad \text{(F-LLM-C51)}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\neg(\text{left}(x,y) \wedge \text{up}(x,y)))) \quad \text{(F-LLM-C52)}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,(\neg(\text{left}_g(x,y)\wedge\text{right}_g(x,y))))\tag{F-LLM-C53}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,(\neg(\text{right}(x,y)\wedge\text{down}(x,y))))\tag{F-LLM-C54}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,(\neg(\text{right}(x,y)\wedge\text{left}(x,y))))\tag{F-LLM-C55}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,(\neg(\text{right}(x,y)\wedge\text{right}(y,x))))\tag{F-LLM-C56}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,(\neg(\text{right}(x,y)\wedge\text{up}(x,y))))\tag{F-LLM-C57}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,(\neg(\text{up}(x,y)\wedge\text{down}(x,y))))\tag{F-LLM-C58}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,(\neg(\text{up}(x,y)\wedge\text{up}(y,x))))\tag{F-LLM-C59}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,(\neg(\text{up}_g(x,y)\wedge\text{down}_g(x,y))))\tag{F-LLM-C60}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{adjacent}(x,y)\rightarrow\text{up}(x,y)\vee\text{down}(x,y)\vee\text{left}(x,y)\vee\text{right}(x,y))))\tag{F-LLM-C61}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{adjacent}(x,y)\rightarrow\text{adjacent}(y,x))))\tag{F-LLM-C62}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{adjacent}(x,y)\rightarrow\text{connected}(x,y))))\tag{F-LLM-C63}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{adjacent}(x,y)\rightarrow\text{connected}(y,x))))\tag{F-LLM-C64}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{connected}(x,y)\rightarrow\text{connected}(y,x))))\tag{F-LLM-C65}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{down}(x,y)\rightarrow\text{up}(y,x))))\tag{F-LLM-C66}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{down}(x,y)\rightarrow\text{vreach}(x,y))))\tag{F-LLM-C67}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{down}_g(x,y)\rightarrow\text{down}(x,y))))\tag{F-LLM-C68}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{down}_g(x,y)\rightarrow\text{up}(y,x))))\tag{F-LLM-C69}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{hreach}(x,y)\rightarrow\text{connected}(x,y))))\tag{F-LLM-C70}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{left}(x,y)\rightarrow\text{hreach}(x,y))))\tag{F-LLM-C71}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{left}(x,y)\rightarrow\text{right}(y,x))))\tag{F-LLM-C72}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{left}_g(x,y)\rightarrow\text{left}(x,y))))\tag{F-LLM-C73}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{left}_g(x,y)\rightarrow\text{right}(y,x))))\tag{F-LLM-C74}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{right}(x,y)\rightarrow\text{hreach}(x,y))))\tag{F-LLM-C75}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{right}(x,y)\rightarrow\text{left}(y,x))))\tag{F-LLM-C76}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{right}_g(x,y)\rightarrow\text{left}(y,x))))\tag{F-LLM-C77}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{right}_g(x,y)\rightarrow\text{right}(y,x))))\tag{F-LLM-C78}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{up}(x,y)\rightarrow\text{down}(y,x))))\tag{F-LLM-C79}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{up}(x,y)\rightarrow\text{vreach}(x,y))))\tag{F-LLM-C80}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{up}_g(x,y)\rightarrow\text{down}(y,x))))\tag{F-LLM-C81}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{up}_g(x,y)\rightarrow\text{up}(x,y))))\tag{F-LLM-C82}$$

$$\forall_{x:\,tile}\,(\forall_{y:\,tile}\,((\text{vreach}(x,y)\rightarrow\text{connected}(x,y))))\tag{F-LLM-C83}$$

$$\forall_{x:\,tile}(\neg(\text{adjacent}(x,x))) \quad \text{(F-LLM-C84)}$$

$$\forall_{x:\,tile}(\neg(\text{down}(x,x))) \quad \text{(F-LLM-C85)}$$

$$\forall_{x:\,tile}(\neg(\text{left}(x,x))) \quad \text{(F-LLM-C86)}$$

$$\forall_{x:\,tile}(\neg(\text{right}(x,x))) \quad \text{(F-LLM-C87)}$$

$$\forall_{x:\,tile}(\neg(\text{up}(x,x))) \quad \text{(F-LLM-C88)}$$

$$\forall_{x:\,tile}((\text{clear}_g(x) \rightarrow \text{clear}(x))) \quad \text{(F-LLM-C89)}$$

$$\forall_{x:\,tile}((\text{free}_{tile}(x) \rightarrow \neg(\text{occupied}(x)))) \quad \text{(F-LLM-C90)}$$

$$\forall_{x:\,tile}((\text{free}_{tile}(x) \rightarrow \text{clear}(x))) \quad \text{(F-LLM-C91)}$$

$$\forall_{x:\,tile}((\text{goal}_{tile}(x) \rightarrow \exists_{c:\,color}(\text{painted}_g(x,c)))) \quad \text{(F-LLM-C92)}$$

$$\forall_{x:\,tile}((\text{goal}_{tile}(x) \rightarrow \exists_{r:\,robot}(\exists_{y:\,tile}(\text{robot-at}(r,y) \wedge \text{connected}(y,x))))) \quad \text{(F-LLM-C93)}$$

$$\forall_{x:\,tile}((\text{goal}_{tile}(x) \rightarrow \text{has}_{neighbor}(x))) \quad \text{(F-LLM-C94)}$$

$$\forall_{x:\,tile}((\text{goal}_{tile}(x) \rightarrow \text{robot}_{reachable}(x))) \quad \text{(F-LLM-C95)}$$

$$\forall_{x:\,tile}((\text{has}_{neighbor}(x) \rightarrow \exists_{y:\,tile}(\text{adjacent}(x,y)))) \quad \text{(F-LLM-C96)}$$

$$\forall_{x:\,tile}((\text{occupied}(x) \rightarrow \neg(\text{clear}(x)))) \quad \text{(F-LLM-C97)}$$

$$\forall_{x:\,tile}((\text{occupied}(x) \rightarrow \neg(\text{free}_{tile}(x)))) \quad \text{(F-LLM-C98)}$$

$$\forall_{x:\,tile}((\text{occupied}(x) \rightarrow \text{robot}_{reachable}(x))) \quad \text{(F-LLM-C99)}$$

$$\forall_{x:\,tile}((\text{robot}_{reachable}(x) \rightarrow \exists_{r:\,robot}(\exists_{y:\,tile}(\text{robot-at}(r,y))))) \quad \text{(F-LLM-C100)}$$

With LLM-generated descriptions:

- F-LLM-C1: There is at least one available color.
- F-LLM-C2: There exists at least one robot-at fact initially.
- F-LLM-C3: There is at least one goal-painted tile.
- F-LLM-C4: Each available color is initially held by some robot.
- F-LLM-C5: Any available-color-goal, if present, implies available-color.
- F-LLM-C6: Definition sanity: goal-color implies it colors some goal tile.
- F-LLM-C7: Any color that appears in the painting goal is available.
- F-LLM-C8: No two distinct robots share a tile initially.
- F-LLM-C9: Distinctness for multiple robots at the same goal tile.
- F-LLM-C10: Every robot initially holds some color.
- F-LLM-C11: If a robot has two colors listed, they are different (functional-like constraint).
- F-LLM-C12: Distinctness for multiple robot-has-goal colors.
- F-LLM-C13: Any color held by a robot is available.
- F-LLM-C14: Any goal-assigned robot color should be an available color.
- F-LLM-C15: Any robot-has-goal atom, if present, corresponds to initial robot-has (vacuous here).
- F-LLM-C16: If goals mention a robot at a goal-painted tile, that goal color is available (vacuous here).
- F-LLM-C17: Any adjacent neighbor of a robot's tile is clear initially (derived version).
- F-LLM-C18: Any down neighbor of a robot's tile is clear initially (as in provided instances).
- F-LLM-C19: Any left neighbor of a robot's tile is clear initially (as in provided instances).
- F-LLM-C20: Any right neighbor of a robot's tile is clear initially (as in provided instances).

- F-LLM-C21: Each robot is at most at one tile.
- F-LLM-C22: Any up neighbor of a robot's tile is clear initially (as in provided instances).
- F-LLM-C23: Distinctness for multiple robot-at-goal positions per robot.
- F-LLM-C24: A robot cannot be on a clear tile initially.
- F-LLM-C25: No robot-at-goal tile is also clear-goal (if both present).
- F-LLM-C26: Every robot stands on a tile that has at least one neighbor (always true in grids).
- F-LLM-C27: Any robot-at-goal atom, if present, corresponds to initial robot-at (vacuous in instances).
- F-LLM-C28: If a robot is flagged with free-color, it holds no color initially.
- F-LLM-C29: Any free-color-goal implies free-color.
- F-LLM-C30: Every non-clear tile is initially occupied by some robot.
- F-LLM-C31: Every non-clear tile is occupied (derived restatement).
- F-LLM-C32: If a tile is clear and unoccupied, it is a free-tile (derived).
- F-LLM-C33: Distinctness for multiple painted-goal colors on the same tile.
- F-LLM-C34: No tile is both painted-goal and clear-goal (if both are present in goal).
- F-LLM-C35: Painted tiles are not clear.
- F-LLM-C36: Goal-painted tiles are not already painted initially (holds in provided instances).
- F-LLM-C37: Goal uses only available colors.
- F-LLM-C38: No tile is both painted and occupied initially.
- F-LLM-C39: If x is adjacent to y and vice versa, x has a neighbor (tautology).
- F-LLM-C40: Primitive adjacencies imply derived adjacent.
- F-LLM-C41: Distinct down predecessors to the same target must be different (distinctness).
- F-LLM-C42: Distinct down targets from the same source must be different (distinctness).
- F-LLM-C43: Distinct left predecessors to the same target must be different (distinctness).
- F-LLM-C44: Distinct left targets from the same source must be different (distinctness).
- F-LLM-C45: Distinct right predecessors to the same target must be different (distinctness).
- F-LLM-C46: Distinct right targets from the same source must be different (functional-like uniqueness).
- F-LLM-C47: Distinct up predecessors to the same target must be different (distinctness).
- F-LLM-C48: Distinct up targets from the same source must be different (distinctness).
- F-LLM-C49: down is antisymmetric (no two-way down).
- F-LLM-C50: A pair cannot be both left and down relations simultaneously.
- F-LLM-C51: left is antisymmetric (no two-way left).
- F-LLM-C52: A pair cannot be both left and up relations simultaneously.
- F-LLM-C53: No pair is both left-goal and right-goal simultaneously (if such atoms exist).
- F-LLM-C54: A pair cannot be both right and down relations simultaneously.
- F-LLM-C55: No pair of tiles is both right and left in the same direction ordering.
- F-LLM-C56: right is antisymmetric (no two-way right).
- F-LLM-C57: A pair cannot be both right and up relations simultaneously.
- F-LLM-C58: No pair is simultaneously up and down in the same ordering.
- F-LLM-C59: up is antisymmetric (no two-way up).
- F-LLM-C60: No pair is both up-goal and down-goal simultaneously (if such atoms exist).
- F-LLM-C61: Derived adjacent implies one of the primitive adjacencies.
- F-LLM-C62: Adjacency is symmetric in these grids (holds in provided instances).
- F-LLM-C63: Adjacency implies connectivity.
- F-LLM-C64: Adjacency in one direction implies connectivity in the reverse direction as well.
- F-LLM-C65: Connectivity is symmetric (given undirected adjacency in instances).
- F-LLM-C66: down is the inverse of up (other direction).
- F-LLM-C67: down implies vertical reachability (derived).
- F-LLM-C68: Any down-goal atom, if present, corresponds to a down relation.
- F-LLM-C69: If a down-goal edge were present, the inverse up edge should hold initially.

- F-LLM-C70: Horizontal reachability implies connectivity.
- F-LLM-C71: left implies horizontal reachability (derived).
- F-LLM-C72: left is the inverse of right (other direction).
- F-LLM-C73: Any left-goal atom, if present, corresponds to a left relation.
- F-LLM-C74: If a left-goal edge were present, the inverse right edge should hold initially.
- F-LLM-C75: right implies horizontal reachability (derived).
- F-LLM-C76: right is the inverse of left (one direction).
- F-LLM-C77: If a right-goal edge were present, the inverse left edge should hold initially.
- F-LLM-C78: Any right-goal atom, if present, corresponds to a right relation.
- F-LLM-C79: up is the inverse of down (one direction).
- F-LLM-C80: up implies vertical reachability (derived).
- F-LLM-C81: If an up-goal edge were present, the inverse down edge should hold initially.
- F-LLM-C82: Any up-goal atom, if present, corresponds to an up relation.
- F-LLM-C83: Vertical reachability implies connectivity.
- F-LLM-C84: No self-adjacency in derived adjacent.
- F-LLM-C85: No self-loop for down.
- F-LLM-C86: No self-loop for left.
- F-LLM-C87: No self-loop for right.
- F-LLM-C88: No self-loop for up.
- F-LLM-C89: Any clear-goal atom, if present, corresponds to clear.
- F-LLM-C90: Derived free-tile excludes occupancy.
- F-LLM-C91: Derived free-tile implies clear.
- F-LLM-C92: Definition sanity: goal-tile implies it has some painted-goal color.
- F-LLM-C93: Every goal tile is connected to some initial robot location (explicit form).
- F-LLM-C94: Every goal tile has at least one neighbor (true in provided grids).
- F-LLM-C95: Every goal tile is connected to some robot initially.
- F-LLM-C96: Sanity: has-neighbor implies existence of some adjacent tile.
- F-LLM-C97: Occupied tiles are not clear (using derived occupied).
- F-LLM-C98: Occupied implies not free-tile (derived).
- F-LLM-C99: Any occupied tile is trivially robot-reachable.
- F-LLM-C100: Robot reachability implies at least one robot exists (trivial sanity).

Which gives the reduced set:

$$\exists_{x:\, tile}\,(\exists_{c:\, color}\,(\text{painted}_g(x,c))) \tag{F-LLM-1}$$

$$\forall_{c:\, color}\,((\text{available-color}(c) \rightarrow \exists_{r:\, robot}\,(\text{robot-has}(r,c)))) \tag{F-LLM-2}$$

$$\forall_{c:\, color}\,((\text{available-color}_g(c) \rightarrow \text{available-color}(c))) \tag{F-LLM-3}$$

$$\forall_{r:\, robot}\,(\exists_{c:\, color}\,(\text{robot-has}(r,c))) \tag{F-LLM-4}$$

$$\forall_{r:\, robot}\,(\forall_{c:\, color}\,(\forall_{d:\, color}\,((\text{robot-has}_g(r,c) \wedge \text{robot-has}_g(r,d) \rightarrow c \neq d)))) \tag{F-LLM-5}$$

$$\forall_{r:\, robot}\,(\forall_{c:\, color}\,((\text{robot-has}(r,c) \rightarrow \text{available-color}(c)))) \tag{F-LLM-6}$$

$$\forall_{r:\, robot}\,(\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{robot-at}(r,x) \wedge \text{up}(y,x) \rightarrow \text{clear}(y))))) \tag{F-LLM-7}$$

$$\forall_{r:\, robot}\,(\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{robot-at}_g(r,x) \wedge \text{robot-at}_g(r,y) \rightarrow x \neq y)))) \tag{F-LLM-8}$$

$$\forall_{r:\, robot}\,(\forall_{x:\, tile}\,((\text{robot-at}(r,x) \rightarrow \text{has}_{neighbor}(x)))) \tag{F-LLM-9}$$

$$\forall_{r:\, robot}\,((\text{free-color}(r) \rightarrow \forall_{c:\, color}\,(\neg(\text{robot-has}(r,c))))) \tag{F-LLM-10}$$

$$\forall_{r:\, robot}\,((\text{free-color}_g(r) \rightarrow \text{free-color}(r))) \tag{F-LLM-11}$$

$$\forall_{x:\, tile}\,((\neg(\text{clear}(x)) \rightarrow \text{occupied}(x))) \tag{F-LLM-12}$$

$$\forall_{x:\, tile}\,(\forall_{c:\, color}\,(\neg(\text{painted}_g(x,c) \wedge \text{clear}_g(x)))) \tag{F-LLM-13}$$

$$\forall_{x:\, tile}\,(\forall_{c:\, color}\,((\text{painted}(x,c) \rightarrow \neg(\text{clear}(x))))) \tag{F-LLM-14}$$

$$\forall_{x:\, tile}\,(\forall_{c:\, color}\,((\text{painted}_g(x,c) \rightarrow \text{available-color}(c)))) \tag{F-LLM-15}$$

$$\forall_{x:\, tile}\,(\forall_{r:\, robot}\,(\forall_{c:\, color}\,(\neg(\text{painted}(x,c) \wedge \text{robot-at}(r,x))))) \tag{F-LLM-16}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\neg(\text{right}(x,y) \wedge \text{down}(x,y)))) \tag{F-LLM-17}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\neg(\text{right}(x,y) \wedge \text{right}(y,x)))) \tag{F-LLM-18}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\neg(\text{right}(x,y) \wedge \text{up}(x,y)))) \tag{F-LLM-19}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,(\neg(\text{up}(x,y) \wedge \text{up}(y,x)))) \tag{F-LLM-20}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{connected}(x,y) \rightarrow \text{connected}(y,x)))) \tag{F-LLM-21}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{down}(x,y) \rightarrow \text{up}(y,x)))) \tag{F-LLM-22}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{down}_g(x,y) \rightarrow \text{up}(y,x)))) \tag{F-LLM-23}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{hreach}(x,y) \rightarrow \text{connected}(x,y)))) \tag{F-LLM-24}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{left}(x,y) \rightarrow \text{right}(y,x)))) \tag{F-LLM-25}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{left}_g(x,y) \rightarrow \text{right}(y,x)))) \tag{F-LLM-26}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{right}(x,y) \rightarrow \text{left}(y,x)))) \tag{F-LLM-27}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{right}_g(x,y) \rightarrow \text{right}(x,y)))) \tag{F-LLM-28}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{up}(x,y) \rightarrow \text{down}(y,x)))) \tag{F-LLM-29}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\text{up}_g(x,y) \rightarrow \text{up}(x,y)))) \tag{F-LLM-30}$$

$$\forall_{x:\, tile}\,(\forall_{y:\, tile}\,((\mathrm{vreach}(x,y) \rightarrow \mathrm{connected}(x,y)))) \quad \text{(F-LLM-31)}$$

$$\forall_{x:\, tile}\,((\mathrm{clear}_g(x) \rightarrow \mathrm{clear}(x))) \quad \text{(F-LLM-32)}$$

$$\forall_{x:\, tile}\,((\mathrm{goal}_{tile}(x) \rightarrow \mathrm{has}_{neighbor}(x))) \quad \text{(F-LLM-33)}$$

$$\forall_{x:\, tile}\,((\mathrm{goal}_{tile}(x) \rightarrow \mathrm{robot}_{reachable}(x))) \quad \text{(F-LLM-34)}$$

$$\forall_{x:\, tile}\,((\mathrm{occupied}(x) \rightarrow \neg(\mathrm{clear}(x)))) \quad \text{(F-LLM-35)}$$

With LLM-generated descriptions:

- F-LLM-1: There is at least one goal-painted tile.
- F-LLM-2: Each available color is initially held by some robot.
- F-LLM-3: Any available-color-goal, if present, implies available-color.
- F-LLM-4: Every robot initially holds some color.
- F-LLM-5: Distinctness for multiple robot-has-goal colors.
- F-LLM-6: Any color held by a robot is available.
- F-LLM-7: Any up neighbor of a robot's tile is clear initially (as in provided instances).
- F-LLM-8: Distinctness for multiple robot-at-goal positions per robot.
- F-LLM-9: Every robot stands on a tile that has at least one neighbor (always true in grids).
- F-LLM-10: If a robot is flagged with free-color, it holds no color initially.
- F-LLM-11: Any free-color-goal implies free-color.
- F-LLM-12: Every non-clear tile is occupied (derived restatement).
- F-LLM-13: No tile is both painted-goal and clear-goal (if both are present in goal).
- F-LLM-14: Painted tiles are not clear.
- F-LLM-15: Goal uses only available colors.
- F-LLM-16: No tile is both painted and occupied initially.
- F-LLM-17: A pair cannot be both right and down relations simultaneously.
- F-LLM-18: right is antisymmetric (no two-way right).
- F-LLM-19: A pair cannot be both right and up relations simultaneously.
- F-LLM-20: up is antisymmetric (no two-way up).
- F-LLM-21: Connectivity is symmetric (given undirected adjacency in instances).
- F-LLM-22: down is the inverse of up (other direction).
- F-LLM-23: If a down-goal edge were present, the inverse up edge should hold initially.
- F-LLM-24: Horizontal reachability implies connectivity.
- F-LLM-25: left is the inverse of right (other direction).
- F-LLM-26: If a left-goal edge were present, the inverse right edge should hold initially.
- F-LLM-27: right is the inverse of left (one direction).
- F-LLM-28: Any right-goal atom, if present, corresponds to a right relation.
- F-LLM-29: up is the inverse of down (one direction).
- F-LLM-30: Any up-goal atom, if present, corresponds to an up relation.
- F-LLM-31: Vertical reachability implies connectivity.
- F-LLM-32: Any clear-goal atom, if present, corresponds to clear.
- F-LLM-33: Every goal tile has at least one neighbor (true in provided grids).
- F-LLM-34: Every goal tile is connected to some robot initially.
- F-LLM-35: Occupied tiles are not clear (using derived occupied).

Grid

Candidate constraints:

$$\text{connected}_{undirected}(x\colon obj,\ y\colon obj) \coloneqq \text{conn}(x,y) \vee \text{conn}(y,x)$$
$$\text{has}_{key\_shape}(k\colon obj) \coloneqq \underset{s\colon obj}{\exists}(\text{key-shape}(k,s))$$
$$\text{has}_{lock}(x\colon obj) \coloneqq \underset{s\colon obj}{\exists}(\text{lock-shape}(x,s))$$
$$\text{key}_{or\_held}(k\colon obj) \coloneqq \underset{p\colon obj}{\exists}(\text{at}(k,p)) \vee \text{holding}(k)$$
$$\text{located}(k\colon obj) \coloneqq \underset{p\colon obj}{\exists}(\text{at}(k,p))$$
$$\text{match}(k\colon obj,\ x\colon obj) \coloneqq \underset{s\colon obj}{\exists}(\text{key-shape}(k,s) \wedge \text{lock-shape}(x,s))$$

$$\underset{p\colon obj}{\exists}(\text{place}(p)) \tag{G-LLM-C1}$$
$$\underset{x\colon obj}{\exists}(\text{at-robot}(x)) \tag{G-LLM-C2}$$
$$\underset{a,b}{\forall}(\text{conn}_g(a,b) \rightarrow a \neq b) \tag{G-LLM-C3}$$
$$\underset{a,b}{\forall}(\text{connected}_{undirected}(a,b) \rightarrow \text{place}(a)) \tag{G-LLM-C4}$$
$$\underset{a,b}{\forall}(\text{connected}_{undirected}(a,b) \rightarrow \text{place}(b)) \tag{G-LLM-C5}$$
$$\underset{c,d}{\forall}(\text{conn}_g(c,d) \rightarrow \text{conn}_g(d,c)) \tag{G-LLM-C6}$$
$$\underset{c1,c2}{\forall}(\text{conn}(c1,c2) \rightarrow \text{connected}_{undirected}(c1,c2)) \tag{G-LLM-C7}$$
$$\underset{cx,cy}{\forall}(\text{conn}_g(cx,cy) \rightarrow \text{conn}(cx,cy)) \tag{G-LLM-C8}$$
$$\underset{d1,d2}{\forall}(\text{connected}_{undirected}(d1,d2) \rightarrow \text{connected}_{undirected}(d2,d1)) \tag{G-LLM-C9}$$
$$\underset{gx}{\forall}(\text{open}_g(gx) \rightarrow \neg(\text{locked}_g(gx))) \tag{G-LLM-C10}$$
$$\underset{hk}{\forall}(\text{has}_{key\_shape}(hk) \rightarrow \text{key}(hk)) \tag{G-LLM-C11}$$
$$\underset{hx}{\forall}(\text{has}_{lock}(hx) \rightarrow \text{place}(hx)) \tag{G-LLM-C12}$$
$$\underset{k,p}{\forall}(\text{at}(k,p) \rightarrow \neg(\text{holding}(k))) \tag{G-LLM-C13}$$
$$\underset{k,p}{\forall}(\text{at}(k,p) \rightarrow k \neq p) \tag{G-LLM-C14}$$
$$\underset{k,p}{\forall}(\text{at}(k,p) \rightarrow \text{key}(k)) \tag{G-LLM-C15}$$
$$\underset{k,p}{\forall}(\text{at}(k,p) \rightarrow \text{place}(p)) \tag{G-LLM-C16}$$
$$\underset{k,p}{\forall}(\text{holding}(k) \rightarrow \neg(\text{at}(k,p))) \tag{G-LLM-C17}$$
$$\underset{k,p1,p2}{\forall}(\text{at}(k,p1) \wedge \text{at}(k,p2) \rightarrow \neg(p1 \neq p2)) \tag{G-LLM-C18}$$
$$\underset{k,s}{\forall}(\text{key-shape}(k,s) \rightarrow \text{key}(k)) \tag{G-LLM-C19}$$
$$\underset{k,s}{\forall}(\text{key-shape}(k,s) \rightarrow \text{shape}(s)) \tag{G-LLM-C20}$$
$$\underset{k,s1,s2}{\forall}(\text{key-shape}(k,s1) \wedge \text{key-shape}(k,s2) \rightarrow \neg(s1 \neq s2)) \tag{G-LLM-C21}$$
$$\underset{k}{\forall}(\text{key}(k) \wedge \text{arm-empty}() \rightarrow \neg(\text{holding}(k))) \tag{G-LLM-C22}$$
$$\underset{k}{\forall}\underset{s\colon obj}{\exists}((\text{key}(k) \rightarrow \text{shape}(s) \wedge \text{key-shape}(k,s))) \tag{G-LLM-C23}$$
$$\underset{k}{\forall}\underset{s\colon obj}{\forall}((\text{key}(k) \rightarrow \neg(\text{lock-shape}(k,s)))) \tag{G-LLM-C24}$$

$$\forall_{k}(\text{holding}(k) \rightarrow \neg(\text{arm-empty}())) \tag{G-LLM-C25}$$

$$\forall_{k}(\text{holding}(k) \rightarrow \text{key}(k)) \tag{G-LLM-C26}$$

$$\forall_{k}(\text{key}(k) \rightarrow \exists_{p:\, obj}(\text{at}(k,p)) \vee \text{holding}(k)) \tag{G-LLM-C27}$$

$$\forall_{k1,k2}(\text{holding}(k1) \wedge \text{holding}(k2) \rightarrow \neg(k1 \neq k2)) \tag{G-LLM-C28}$$

$$\forall_{k1,s1}(\text{shape}(s1) \rightarrow \neg(\text{at}(k1,s1))) \tag{G-LLM-C29}$$

$$\forall_{kg,p1,p2}(\text{at}_g(kg,p1) \wedge \text{at}_g(kg,p2) \rightarrow \neg(p1 \neq p2)) \tag{G-LLM-C30}$$

$$\forall_{kg,pg}(\text{at}_g(kg,pg) \rightarrow \neg(\text{holding}_g(kg))) \tag{G-LLM-C31}$$

$$\forall_{kg,pg}(\text{at}_g(kg,pg) \rightarrow kg \neq pg) \tag{G-LLM-C32}$$

$$\forall_{kg,pg}(\text{at}_g(kg,pg) \rightarrow \text{key}(kg)) \tag{G-LLM-C33}$$

$$\forall_{kg,pg}(\text{at}_g(kg,pg) \rightarrow \text{place}(pg)) \tag{G-LLM-C34}$$

$$\forall_{kgk}(\text{key}_g(kgk) \rightarrow \text{key}(kgk)) \tag{G-LLM-C35}$$

$$\forall_{kh}(\text{holding}_g(kh) \rightarrow \text{key}(kh)) \tag{G-LLM-C36}$$

$$\forall_{kk,ss}(\text{key-shape}(kk,ss) \rightarrow kk \neq ss) \tag{G-LLM-C37}$$

$$\forall_{kk}(\text{key}(kk) \rightarrow \text{has}_{key\_shape}(kk)) \tag{G-LLM-C38}$$

$$\forall_{kk}(\text{key}(kk) \rightarrow \text{key}_{or\_held}(kk)) \tag{G-LLM-C39}$$

$$\forall_{kk}(\text{key}_{or\_held}(kk) \rightarrow \text{key}(kk)) \tag{G-LLM-C40}$$

$$\forall_{kx,y1}(\text{key}(kx) \rightarrow \neg(\text{conn}(kx,y1))) \tag{G-LLM-C41}$$

$$\forall_{kz}(\text{key}(kz) \rightarrow \neg(\text{locked}_g(kz))) \tag{G-LLM-C42}$$

$$\forall_{kz}(\text{key}(kz) \rightarrow \neg(\text{open}_g(kz))) \tag{G-LLM-C43}$$

$$\forall_{lgx}(\text{locked}_g(lgx) \rightarrow \text{place}(lgx)) \tag{G-LLM-C44}$$

$$\forall_{lx,sk:\, obj}\exists((\text{lock-shape}(lx,s) \rightarrow \text{key-shape}(k,s))) \tag{G-LLM-C45}$$

$$\forall_{lx,s}(\text{lock-shape}(lx,s) \rightarrow \text{locked}(lx)) \tag{G-LLM-C46}$$

$$\forall_{lxk:\, obj}\exists((\text{locked}(lx) \rightarrow \text{match}(k,lx))) \tag{G-LLM-C47}$$

$$\forall_{lxs:\, obj}\exists((\text{locked}(lx) \rightarrow \text{shape}(s) \wedge \text{lock-shape}(lx,s))) \tag{G-LLM-C48}$$

$$\forall_{lx}(\text{has}_{lock}(lx) \rightarrow \neg(\text{open}(lx))) \tag{G-LLM-C49}$$

$$\forall_{lx}(\text{has}_{lock}(lx) \rightarrow \text{locked}(lx)) \tag{G-LLM-C50}$$

$$\forall_{lx}(\text{locked}(lx) \rightarrow \text{has}_{lock}(lx)) \tag{G-LLM-C51}$$

$$\forall_{ogx}(\text{open}_g(ogx) \rightarrow \text{place}(ogx)) \tag{G-LLM-C52}$$

$$\forall_{pgp}(\text{place}_g(pgp) \rightarrow \text{place}(pgp)) \tag{G-LLM-C53}$$

$$\forall_{px}(\text{place}(px) \rightarrow \neg(\text{shape}(px))) \tag{G-LLM-C54}$$

$$\forall_{q,s}(\text{open}(q) \rightarrow \neg(\text{lock-shape}(q,s))) \tag{G-LLM-C55}$$

$$\forall_{q}\neg(\text{open}(q) \wedge \text{locked}(q)) \tag{G-LLM-C56}$$

$$\underset{r,s}{\forall}(\text{at-robot}(r) \land \text{shape}(s) \rightarrow r \neq s) \tag{G-LLM-C57}$$

$$\underset{r,y}{\forall}(\text{at-robot}(r) \land \text{key}(y) \rightarrow r \neq y) \tag{G-LLM-C58}$$

$$\underset{r}{\forall}(\text{at-robot}(r) \rightarrow \text{open}(r)) \tag{G-LLM-C59}$$

$$\underset{rg}{\forall}(\text{at-robot}_g(rg) \rightarrow \text{open}(rg)) \tag{G-LLM-C60}$$

$$\underset{rg}{\forall}(\text{at-robot}_g(rg) \rightarrow \text{place}(rg)) \tag{G-LLM-C61}$$

$$\underset{rg1,rg2}{\forall}(\text{at-robot}_g(rg1) \land \text{at-robot}_g(rg2) \rightarrow \neg(rg1 \neq rg2)) \tag{G-LLM-C62}$$

$$\underset{s,x}{\forall}(\text{shape}(s) \rightarrow \neg(\text{conn}(s,x))) \tag{G-LLM-C63}$$

$$\underset{s1,p}{\forall}(\text{shape}(s1) \rightarrow \neg(\text{at}(s1,p))) \tag{G-LLM-C64}$$

$$\underset{s2,x2}{\forall}(\text{shape}(s2) \rightarrow \neg(\text{lock-shape}(s2,x2))) \tag{G-LLM-C65}$$

$$\underset{s3,x3}{\forall}(\text{shape}(s3) \rightarrow \neg(\text{key-shape}(s3,x3))) \tag{G-LLM-C66}$$

$$\underset{sg}{\forall}(\text{shape}(sg) \rightarrow \neg(\text{at-robot}_g(sg))) \tag{G-LLM-C67}$$

$$\underset{sg}{\forall}(\text{shape}(sg) \rightarrow \neg(\text{locked}_g(sg))) \tag{G-LLM-C68}$$

$$\underset{sg}{\forall}(\text{shape}(sg) \rightarrow \neg(\text{open}_g(sg))) \tag{G-LLM-C69}$$

$$\underset{sgs}{\forall}(\text{shape}_g(sgs) \rightarrow \text{shape}(sgs)) \tag{G-LLM-C70}$$

$$\underset{u,v}{\forall}(\text{conn}(u,v) \rightarrow u \neq v) \tag{G-LLM-C71}$$

$$\underset{x,s}{\forall}(\text{lock-shape}(x,s) \rightarrow \text{place}(x)) \tag{G-LLM-C72}$$

$$\underset{x,s}{\forall}(\text{lock-shape}(x,s) \rightarrow \text{shape}(s)) \tag{G-LLM-C73}$$

$$\underset{x,s}{\forall}(\text{shape}(s) \rightarrow \neg(\text{conn}(x,s))) \tag{G-LLM-C74}$$

$$\underset{x,s1,s2}{\forall}(\text{lock-shape}(x,s1) \land \text{lock-shape}(x,s2) \rightarrow \neg(s1 \neq s2)) \tag{G-LLM-C75}$$

$$\underset{x,y}{\forall}(\text{at-robot}(x) \land \text{at-robot}(y) \rightarrow \neg(x \neq y)) \tag{G-LLM-C76}$$

$$\underset{x,y}{\forall}(\text{conn}(x,y) \rightarrow \text{conn}(y,x)) \tag{G-LLM-C77}$$

$$\underset{x,y}{\forall}(\text{conn}(x,y) \rightarrow \text{place}(x)) \tag{G-LLM-C78}$$

$$\underset{x,y}{\forall}(\text{conn}(x,y) \rightarrow \text{place}(y)) \tag{G-LLM-C79}$$

$$\underset{x,y}{\forall}(\text{conn}_g(x,y) \rightarrow \text{place}(x)) \tag{G-LLM-C80}$$

$$\underset{x,y}{\forall}(\text{conn}_g(x,y) \rightarrow \text{place}(y)) \tag{G-LLM-C81}$$

$$\underset{x}{\forall}\underset{s:\ obj}{\forall}((\text{place}(x) \rightarrow \neg(\text{key-shape}(x,s)))) \tag{G-LLM-C82}$$

$$\underset{x}{\forall}(\text{at-robot}(x) \rightarrow \text{place}(x)) \tag{G-LLM-C83}$$

$$\underset{x}{\forall}(\text{key}(x) \rightarrow \neg(\text{at-robot}(x))) \tag{G-LLM-C84}$$

$$\underset{x}{\forall}(\text{key}(x) \rightarrow \neg(\text{locked}(x))) \tag{G-LLM-C85}$$

$$\underset{x}{\forall}(\text{key}(x) \rightarrow \neg(\text{open}(x))) \tag{G-LLM-C86}$$

$$\underset{x}{\forall}(\text{key}(x) \rightarrow \neg(\text{place}(x))) \tag{G-LLM-C87}$$

$$\underset{x}{\forall}(\text{locked}(x) \rightarrow \text{place}(x)) \tag{G-LLM-C88}$$

$$\underset{x}{\forall}(\text{open}(x) \rightarrow \text{place}(x)) \tag{G-LLM-C89}$$

$$\forall_{x}(\text{place}(x) \rightarrow \neg(\text{key}(x))) \quad \text{(G-LLM-C90)}$$

$$\forall_{x}(\text{shape}(x) \rightarrow \neg(\text{at-robot}(x))) \quad \text{(G-LLM-C91)}$$

$$\forall_{x}(\text{shape}(x) \rightarrow \neg(\text{key}(x))) \quad \text{(G-LLM-C92)}$$

$$\forall_{x}(\text{shape}(x) \rightarrow \neg(\text{locked}(x))) \quad \text{(G-LLM-C93)}$$

$$\forall_{x}(\text{shape}(x) \rightarrow \neg(\text{open}(x))) \quad \text{(G-LLM-C94)}$$

$$\forall_{x}(\text{shape}(x) \rightarrow \neg(\text{place}(x))) \quad \text{(G-LLM-C95)}$$

$$\forall_{xg,sg}(\text{lock-shape}_{g}(xg, sg) \rightarrow \text{place}(xg)) \quad \text{(G-LLM-C96)}$$

$$\forall_{xg,sg}(\text{lock-shape}_{g}(xg, sg) \rightarrow \text{shape}(sg)) \quad \text{(G-LLM-C97)}$$

$$\forall_{xx,ss}(\text{lock-shape}(xx, ss) \rightarrow xx \neq ss) \quad \text{(G-LLM-C98)}$$

$$\forall_{y2,kx}(\text{key}(kx) \rightarrow \neg(\text{conn}(y2, kx))) \quad \text{(G-LLM-C99)}$$

$$\forall_{z}\neg(\text{conn}(z, z)) \quad \text{(G-LLM-C100)}$$

With LLM-generated descriptions:

- G-LLM-C1: There exists at least one place.
- G-LLM-C2: There exists a robot location.
- G-LLM-C3: Goal connections are between distinct places.
- G-LLM-C4: Undirected connection implies first arg is a place.
- G-LLM-C5: Undirected connection implies second arg is a place.
- G-LLM-C6: Goal connectivity is symmetric if present.
- G-LLM-C7: conn(x,y) implies connected-undirected(x,y).
- G-LLM-C8: If a connection appears in the goal, it is a base connection.
- G-LLM-C9: connected-undirected is symmetric.
- G-LLM-C10: Goal-open and goal-locked are mutex for the same place.
- G-LLM-C11: Derived: has-key-shape(k) implies k is a key.
- G-LLM-C12: Derived: has-lock(x) implies x is a place.
- G-LLM-C13: A key cannot be at a place and held simultaneously.
- G-LLM-C14: Keys and places are distinct in at(k,p).
- G-LLM-C15: Typing: at(k,p) implies k is a key.
- G-LLM-C16: Typing: at(k,p) implies p is a place.
- G-LLM-C17: If a key is held then it is not at any place.
- G-LLM-C18: Each key is at most at one place initially.
- G-LLM-C19: Typing: key-shape(k,s) implies k is a key.
- G-LLM-C20: Typing: key-shape(k,s) implies s is a shape.
- G-LLM-C21: Key-shape is functionally unique per key.
- G-LLM-C22: If arm is empty then no key is held.
- G-LLM-C23: Each key has some associated shape.
- G-LLM-C24: No key appears as the place argument of lock-shape.
- G-LLM-C25: If holding some key then the arm is not empty.
- G-LLM-C26: Holding implies the object is a key.
- G-LLM-C27: Every key is either at some place or held.
- G-LLM-C28: At most one key is held initially.
- G-LLM-C29: Shapes are not used as the second argument of at.

- G-LLM-C30: Goal assigns at most one target place per key.
- G-LLM-C31: If a key must end at a place in the goal, it is not required to be held in the goal.
- G-LLM-C32: Goal at(k,p) uses distinct objects.
- G-LLM-C33: Goal at(k,p) uses a key as first argument.
- G-LLM-C34: Goal at(k,p) uses a place as second argument.
- G-LLM-C35: key-goal implies the object is a key.
- G-LLM-C36: Goal holding(k) refers to keys.
- G-LLM-C37: Key-shape uses distinct key and shape objects.
- G-LLM-C38: Derived: keys have some key-shape.
- G-LLM-C39: Derived: keys are located somewhere or held.
- G-LLM-C40: Derived: only keys can be located or held.
- G-LLM-C41: Keys do not participate in connections as first argument.
- G-LLM-C42: Keys are not marked locked in the goal.
- G-LLM-C43: Keys are not marked open in the goal.
- G-LLM-C44: locked-goal(x) applies to places.
- G-LLM-C45: For each lock-shape, there exists some key of the same shape.
- G-LLM-C46: Any place with a lock-shape is initially locked.
- G-LLM-C47: Derived: every locked place has some matching key.
- G-LLM-C48: Every locked place has some lock-shape and that shape object exists.
- G-LLM-C49: Derived: has-lock places are not open initially.
- G-LLM-C50: Derived: if an object has a lock, it is locked initially.
- G-LLM-C51: Derived: every locked place has a lock.
- G-LLM-C52: open-goal(x) applies to places.
- G-LLM-C53: place-goal implies the object is a place.
- G-LLM-C54: Places are not shapes (disjoint types).
- G-LLM-C55: Open places do not carry lock-shape initially.
- G-LLM-C56: No place is both open and locked initially.
- G-LLM-C57: Robot location differs from any shape object.
- G-LLM-C58: Robot location differs from any key object.
- G-LLM-C59: Robot starts on an open place.
- G-LLM-C60: If a goal specifies robot at x, that x is open initially (likely in this domain).
- G-LLM-C61: Goal robot location (if any) is a place.
- G-LLM-C62: At most one goal robot location.
- G-LLM-C63: Shapes do not participate in outgoing connections.
- G-LLM-C64: Shapes are not used as the first argument of at.
- G-LLM-C65: Shapes are not used as the first argument of lock-shape.
- G-LLM-C66: Shapes are not used as the first argument of key-shape.
- G-LLM-C67: Shapes are not goal robot locations.
- G-LLM-C68: Shapes are not marked locked in the goal.
- G-LLM-C69: Shapes are not marked open in the goal.
- G-LLM-C70: shape-goal implies the object is a shape.
- G-LLM-C71: Connected nodes are distinct.
- G-LLM-C72: Typing: lock-shape(x,s) implies x is a place.
- G-LLM-C73: Typing: lock-shape(x,s) implies s is a shape.
- G-LLM-C74: Shapes do not participate in incoming connections.
- G-LLM-C75: Lock-shape is functionally unique per place.
- G-LLM-C76: Robot is at most at one place initially.
- G-LLM-C77: Connectivity is symmetric.
- G-LLM-C78: Typing: conn(x,y) implies x is a place.

- G-LLM-C79: Typing: conn(x,y) implies y is a place.
- G-LLM-C80: Goal conn(x,y) uses place objects (x).
- G-LLM-C81: Goal conn(x,y) uses place objects (y).
- G-LLM-C82: No place appears as the key argument of key-shape.
- G-LLM-C83: Typing: robot location must be a place.
- G-LLM-C84: Keys are not robot locations.
- G-LLM-C85: Keys cannot be locked.
- G-LLM-C86: Keys cannot be open.
- G-LLM-C87: Keys are not places.
- G-LLM-C88: Only places can be locked.
- G-LLM-C89: Only places can be open.
- G-LLM-C90: Places are not keys.
- G-LLM-C91: Shapes are not robot locations.
- G-LLM-C92: Shapes are not keys.
- G-LLM-C93: Shapes cannot be locked.
- G-LLM-C94: Shapes cannot be open.
- G-LLM-C95: Shapes are not places.
- G-LLM-C96: lock-shape-goal(x,s) uses a place x.
- G-LLM-C97: lock-shape-goal(x,s) uses a shape s.
- G-LLM-C98: Lock-shape uses distinct place and shape objects.
- G-LLM-C99: Keys do not participate in connections as second argument.
- G-LLM-C100: No self-connections.

Which gives the reduced set:

$$\underset{x:\, obj}{\exists}(\text{at-robot}(x)) \tag{G-LLM-1}$$

$$\underset{c,d}{\forall}(\text{conn}_g(c,d) \rightarrow \text{conn}_g(d,c)) \tag{G-LLM-2}$$

$$\underset{cx,cy}{\forall}(\text{conn}_g(cx,cy) \rightarrow \text{conn}(cx,cy)) \tag{G-LLM-3}$$

$$\underset{gx}{\forall}(\text{open}_g(gx) \rightarrow \neg(\text{locked}_g(gx))) \tag{G-LLM-4}$$

$$\underset{k,p}{\forall}(\text{at}(k,p) \rightarrow \text{key}(k)) \tag{G-LLM-5}$$

$$\underset{k,p}{\forall}(\text{at}(k,p) \rightarrow \text{place}(p)) \tag{G-LLM-6}$$

$$\underset{k,p}{\forall}(\text{holding}(k) \rightarrow \neg(\text{at}(k,p))) \tag{G-LLM-7}$$

$$\underset{k,p1,p2}{\forall}(\text{at}(k,p1) \wedge \text{at}(k,p2) \rightarrow \neg(p1 \neq p2)) \tag{G-LLM-8}$$

$$\underset{k,s}{\forall}(\text{key-shape}(k,s) \rightarrow \text{key}(k)) \tag{G-LLM-9}$$

$$\underset{k,s1,s2}{\forall}(\text{key-shape}(k,s1) \wedge \text{key-shape}(k,s2) \rightarrow \neg(s1 \neq s2)) \tag{G-LLM-10}$$

$$\underset{k}{\forall}\underset{s:\, obj}{\exists}((\text{key}(k) \rightarrow \text{shape}(s) \wedge \text{key-shape}(k,s))) \tag{G-LLM-11}$$

$$\underset{k}{\forall}(\text{holding}(k) \rightarrow \neg(\text{arm-empty}())) \tag{G-LLM-12}$$

$$\underset{k}{\forall}(\text{holding}(k) \rightarrow \text{key}(k)) \tag{G-LLM-13}$$

$$\underset{k1,k2}{\forall}(\text{holding}(k1) \wedge \text{holding}(k2) \rightarrow \neg(k1 \neq k2)) \tag{G-LLM-14}$$

$$\underset{kg,p1,p2}{\forall}(\text{at}_g(kg,p1) \wedge \text{at}_g(kg,p2) \rightarrow \neg(p1 \neq p2)) \tag{G-LLM-15}$$

$$\underset{kg,pg}{\forall}(\text{at}_g(kg, pg) \rightarrow \neg(\text{holding}_g(kg))) \quad \text{(G-LLM-16)}$$

$$\underset{kg,pg}{\forall}(\text{at}_g(kg, pg) \rightarrow \text{key}(kg)) \quad \text{(G-LLM-17)}$$

$$\underset{kg,pg}{\forall}(\text{at}_g(kg, pg) \rightarrow \text{place}(pg)) \quad \text{(G-LLM-18)}$$

$$\underset{kgk}{\forall}(\text{key}_g(kgk) \rightarrow \text{key}(kgk)) \quad \text{(G-LLM-19)}$$

$$\underset{kh}{\forall}(\text{holding}_g(kh) \rightarrow \text{key}(kh)) \quad \text{(G-LLM-20)}$$

$$\underset{kk}{\forall}(\text{key}(kk) \rightarrow \text{key}_{or\_held}(kk)) \quad \text{(G-LLM-21)}$$

$$\underset{lgx}{\forall}(\text{locked}_g(lgx) \rightarrow \text{place}(lgx)) \quad \text{(G-LLM-22)}$$

$$\underset{lx,s}{\forall}(\text{lock-shape}(lx, s) \rightarrow \text{locked}(lx)) \quad \text{(G-LLM-23)}$$

$$\underset{lxk:\ obj}{\forall\ \exists}((\text{locked}(lx) \rightarrow \text{match}(k, lx))) \quad \text{(G-LLM-24)}$$

$$\underset{ogx}{\forall}(\text{open}_g(ogx) \rightarrow \text{place}(ogx)) \quad \text{(G-LLM-25)}$$

$$\underset{pgp}{\forall}(\text{place}_g(pgp) \rightarrow \text{place}(pgp)) \quad \text{(G-LLM-26)}$$

$$\underset{q}{\forall}\neg(\text{open}(q) \wedge \text{locked}(q)) \quad \text{(G-LLM-27)}$$

$$\underset{r}{\forall}(\text{at-robot}(r) \rightarrow \text{open}(r)) \quad \text{(G-LLM-28)}$$

$$\underset{rg}{\forall}(\text{at-robot}_g(rg) \rightarrow \text{open}(rg)) \quad \text{(G-LLM-29)}$$

$$\underset{rg1,rg2}{\forall}(\text{at-robot}_g(rg1) \wedge \text{at-robot}_g(rg2) \rightarrow \neg(rg1 \neq rg2)) \quad \text{(G-LLM-30)}$$

$$\underset{sgs}{\forall}(\text{shape}_g(sgs) \rightarrow \text{shape}(sgs)) \quad \text{(G-LLM-31)}$$

$$\underset{x,s1,s2}{\forall}(\text{lock-shape}(x, s1) \wedge \text{lock-shape}(x, s2) \rightarrow \neg(s1 \neq s2)) \quad \text{(G-LLM-32)}$$

$$\underset{x,y}{\forall}(\text{at-robot}(x) \wedge \text{at-robot}(y) \rightarrow \neg(x \neq y)) \quad \text{(G-LLM-33)}$$

$$\underset{x,y}{\forall}(\text{conn}(x, y) \rightarrow \text{conn}(y, x)) \quad \text{(G-LLM-34)}$$

$$\underset{x,y}{\forall}(\text{conn}(x, y) \rightarrow \text{place}(y)) \quad \text{(G-LLM-35)}$$

$$\underset{x}{\forall}(\text{locked}(x) \rightarrow \text{place}(x)) \quad \text{(G-LLM-36)}$$

$$\underset{x}{\forall}(\text{open}(x) \rightarrow \text{place}(x)) \quad \text{(G-LLM-37)}$$

$$\underset{x}{\forall}(\text{place}(x) \rightarrow \neg(\text{key}(x))) \quad \text{(G-LLM-38)}$$

$$\underset{x}{\forall}(\text{shape}(x) \rightarrow \neg(\text{key}(x))) \quad \text{(G-LLM-39)}$$

$$\underset{x}{\forall}(\text{shape}(x) \rightarrow \neg(\text{place}(x))) \quad \text{(G-LLM-40)}$$

$$\underset{xg,sg}{\forall}(\text{lock-shape}_g(xg, sg) \rightarrow \text{place}(xg)) \quad \text{(G-LLM-41)}$$

$$\underset{xg,sg}{\forall}(\text{lock-shape}_g(xg, sg) \rightarrow \text{shape}(sg)) \quad \text{(G-LLM-42)}$$

$$\underset{z}{\forall}\neg(\text{conn}(z, z)) \quad \text{(G-LLM-43)}$$

With LLM-generated descriptions:

- G-LLM-1: There exists a robot location.
- G-LLM-2: Goal connectivity is symmetric if present.
- G-LLM-3: If a connection appears in the goal, it is a base connection.
- G-LLM-4: Goal-open and goal-locked are mutex for the same place.

- G-LLM-5: Typing: at(k,p) implies k is a key.
- G-LLM-6: Typing: at(k,p) implies p is a place.
- G-LLM-7: If a key is held then it is not at any place.
- G-LLM-8: Each key is at most at one place initially.
- G-LLM-9: Typing: key-shape(k,s) implies k is a key.
- G-LLM-10: Key-shape is functionally unique per key.
- G-LLM-11: Each key has some associated shape.
- G-LLM-12: If holding some key then the arm is not empty.
- G-LLM-13: Holding implies the object is a key.
- G-LLM-14: At most one key is held initially.
- G-LLM-15: Goal assigns at most one target place per key.
- G-LLM-16: If a key must end at a place in the goal, it is not required to be held in the goal.
- G-LLM-17: Goal at(k,p) uses a key as first argument.
- G-LLM-18: Goal at(k,p) uses a place as second argument.
- G-LLM-19: key-goal implies the object is a key.
- G-LLM-20: Goal holding(k) refers to keys.
- G-LLM-21: Derived: keys are located somewhere or held.
- G-LLM-22: locked-goal(x) applies to places.
- G-LLM-23: Any place with a lock-shape is initially locked.
- G-LLM-24: Derived: every locked place has some matching key.
- G-LLM-25: open-goal(x) applies to places.
- G-LLM-26: place-goal implies the object is a place.
- G-LLM-27: No place is both open and locked initially.
- G-LLM-28: Robot starts on an open place.
- G-LLM-29: If a goal specifies robot at x, that x is open initially (likely in this domain).
- G-LLM-30: At most one goal robot location.
- G-LLM-31: shape-goal implies the object is a shape.
- G-LLM-32: Lock-shape is functionally unique per place.
- G-LLM-33: Robot is at most at one place initially.
- G-LLM-34: Connectivity is symmetric.
- G-LLM-35: Typing: conn(x,y) implies y is a place.
- G-LLM-36: Only places can be locked.
- G-LLM-37: Only places can be open.
- G-LLM-38: Places are not keys.
- G-LLM-39: Shapes are not keys.
- G-LLM-40: Shapes are not places.
- G-LLM-41: lock-shape-goal(x,s) uses a place x.
- G-LLM-42: lock-shape-goal(x,s) uses a shape s.
- G-LLM-43: No self-connections.