On the Relation between Star-Topology Decoupling and Petri Net Unfolding

Daniel Gnad and **Jörg Hoffmann**
Saarland University
Saarland Informatics Campus
Saarbrücken, Germany
{gnad,hoffmann}@cs.uni-saarland.de

Abstract

Petri net unfolding expands concurrent sub-threads of a transition system separately. In AI Planning, star-topology decoupling (STD) finds a partitioning of state variables into components whose dependencies take a star shape, and expands leaf-component state spaces separately. Thus both techniques rely on the separate expansion of state-space composites. How do they relate? We show that, provided compatible search orderings, STD state space size dominates that of unfolding if every component contains a single state variable, and unfolding dominates STD in the absence of prevail conditions (non-deleted action preconditions). In all other cases, exponential state space size advantages are possible on either side. Thus the sources of exponential advantages of STD are exactly a) state space size in the presence of prevail conditions (our results), and b) decidability of reachability in time linear in state space size vs. **NP**-hard for unfolding (known results).

Introduction

Petri net unfolding is a well-known partial-order reduction method (e. g. McMillan (1992), Esparza, Römer, and Vogler (2002), Baldan et al. (2012)). It maintains concurrent threads separately. Instead of building the forward state space and trying to prune permutative parts as in other methods (e. g. Valmari (1989), Godefroid and Wolper (1991), Wehrle et al. (2013), Wehrle and Helmert (2014)), the state variables are not multiplied with each other in the first place. The unfolding process incrementally adds transitions to an acyclic graph, when the transition's input "places" (precondition facts) can be reached jointly. A new output place is then added for each effect. The outcome structure is an acyclic Petri net, a *complete prefix*, that preserves reachability exactly relative to the input Petri net.

In classical planning, transition systems are described by finite-domain state variables, where actions have conjunctive preconditions and effects over these. This can be translated into Petri nets (Hickmott et al. 2007; Bonet et al. 2008). Each place in an unfolding then corresponds to a state-variable value, and the complete prefix is an acyclic fact-action dependency structure that captures reachability.

Star-topology decoupling (STD) (Gnad and Hoffmann 2018) finds a partitioning of the state variables into components, where all cross-component dependencies involve a *center* component C, so that the other components $\mathcal{L}$ can be viewed as *leaves*. The search explores action sequences affecting the center. At each search node, the reachable leaf states are expanded for each leaf $L \in \mathcal{L}$ individually.

Unfolding and STD are related. Consider a *singleton-component* topology, where each component contains a single state variable. The component states then are exactly the places in the unfolding, and, like unfolding, STD expands these separately. So how do the techniques relate exactly?

A significant known separation is the complexity of deciding whether a conjunctive condition is reachable. Given an unfolding prefix, this test is **NP**-complete (McMillan 1992) as the transition histories supporting two places may be in conflict. In contrast, given a decoupled state space prefix, the test can be done in time linear in prefix size: thanks to the star-topology organization, there are no conflicts. But what about the sizes of the fully expanded prefixes, i. e., the respective complete representation of reachability?

Gnad and Hoffmann (2018) have shown that the STD state space can be exponentially smaller in the presence of prevail conditions (non-deleted preconditions, whose treatment in Petri nets leads to blow-ups); and that the complete unfolding prefix can be exponentially smaller in the presence of non-singleton components. Here, we show corresponding dominance results: without prevail conditions, unfolding size dominates STD size; with only singleton components, STD size dominates unfolding size. Both hold subject to *compatible* search orders, that prefer expansions on leaves over ones on the center whenever possible, and that are identical on the ordering of center-action sequences. For incompatible search orders, exponential advantages are possible in either direction. Overall, we obtain a complete classification along the three dimensions of prevail conditions (yes/no), non-singleton components (yes/no), and incompatible search orderings (yes/no).

Background

We provide an overview suited to understand our results at a high level. More technical details and notations, as needed for the full proofs, are given in the appendix.

We use finite-domain state variables (Bäckström and Nebel 1995; Helmert 2006). A planning *task* is a tuple $\Pi = \langle \mathcal{V}, \mathcal{A}, I, G \rangle$. Here, $\mathcal{V}$ is a set of *variables*, each associated

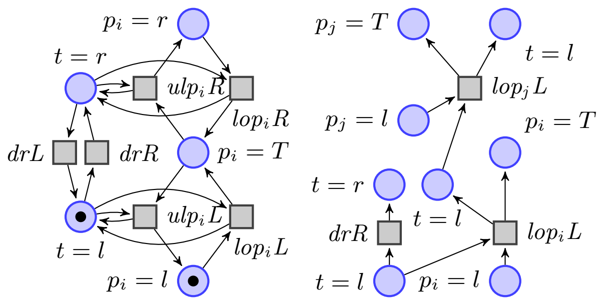

Figure 1: Part of the petri-net encoding of our running example (left), and an incomplete prefix of its unfolding (right).

with a finite *domain* $\mathcal{D}(v)$. (Partial) variable assignments are identified with sets of variable/value pairs, called *facts*, denoted $(var = val)$. A *state* is a complete assignment to $\mathcal{V}$. I is the *initial state*, and the *goal* G is a partial assignment to $\mathcal{V}$. $\mathcal{A}$ is a finite set of *actions*, each a pair $\langle \mathsf{pre}(a), \mathsf{eff}(a) \rangle$ of partial assignments to $\mathcal{V}$, called *precondition* and *effect* of a.

For a partial assignment p, $vars(p)$ is the subset of variables on which p is defined. For $V \subseteq vars(p)$, $p[V]$ denotes the assignment to V made by p. We say that p *satisfies* a condition q, $p \models q$, if $vars(q) \subseteq vars(p)$, and $p[v] = q[v]$ for all $v \in vars(q)$. An action a is *applicable* in a (partial) state s if $s \models \mathsf{pre}(a)[vars(s)]$. If so, the outcome of applying a in s is denoted $s[\![a]\!]$, where $s[\![a]\!][v] = \mathsf{eff}(a)[v]$ for $v \in vars(\mathsf{eff}(a)) \cap vars(s)$, and is $s[\![a]\!][v] = s[v]$ elsewhere.

For convenience in the encoding as Petri nets, we assume that there are no effect-only variables, $v \in vars(\mathsf{eff}(a)) \setminus vars(\mathsf{pre}(a))$. This is WLOG as such variables can be compiled away with a linear size increase (Pommerening and Helmert 2015). What will be important, however, are *prevail* conditions, $v \in vars(\mathsf{pre}(a)) \setminus vars(\mathsf{eff}(a))$.

For illustration, we use a simple logistics example that highlights some key differences between STD and unfolding. $\mathcal{V} = \{t, p_1, \dots, p_n\}$ where t encodes the position of a truck on a map with two locations l, r; and each p_i encodes the position of a package. We have $D_t = \{l, r\}$ and $D_{p_i} = \{l, r, T\}$ where T stands for being in the truck. In I, all variables have value l. Actions drive, e. g. drR with precondition $\{(t, l)\}$ and effect $\{(t, r)\}$; or load a package, e. g. $lop_1 L$ with precondition $\{(t, l), (p_1, l)\}$ and effect $\{(p_1, T)\}$; or unload a package accordingly. Note that load and unload actions have a prevail condition on the truck.

Petri-Net Unfolding

Planning tasks can be encoded in *Petri nets* $\Sigma = \langle P, T, F, M_0 \rangle$, which are digraphs whose nodes are the *places* P and *transitions* T of Σ. When encoding a planning task Π (Hickmott et al. 2007; Bonet et al. 2008), the places P correspond to the facts of Π, and the transitions $t \in T$ correspond to the actions $a \in \mathcal{A}$. The *flow relation* F connects precondition places as input to transitions, and effects as their outcome, e. g. $(p, t) \in F$ means that t has precondition p. A state s of Π (a set of facts) becomes a *marking* M in Σ (a set of places). $M_0 = I$ is the initial marking. A transition t can *fire* (i. e., is applicable) in a marking M if $\mathsf{pre}(t) \subseteq M$. The resulting marking is $M' = (M \setminus \mathsf{pre}(t)) \cup \mathsf{eff}(t)$.

Figure 1 (left) illustrates the Petri net encoding of our running example. Places are blue circles, transitions gray boxes. A marking M places a token in every place $p \in M$.

Petri nets do not natively support prevail conditions. The input places of transitions are "consumed" when a transition fires. Prevail conditions can be encoded by re-adding a token to the respective place. In our example, the encoding of (un)load actions consumes the current truck position, and adds it back in the effect. This incurs a blow-up in the unfolding (illustrated below), as a distinction is introduced between the truck position before vs. after (un)load actions.

The outcome of the unfolding process for a Petri net Σ is a triple $\mathit{Unf}_\Pi = \langle B, E, G \rangle$ that captures all markings reachable from M_0 in Σ. The *conditions* $b \in B$ and *events* $e \in E$ of Unf_Π are labeled with the places $p \in P$ and transitions $t \in T$ in Σ. G extends the flow relation F according to these labels. The unfolding process ensures that G is acyclic. A *configuration* is a set $C \subseteq E$ of events, partially ordered by G, that includes all its predecessor events and that can be sequenced to an executable transition sequence.

The unfolding is generated as follows. Initially, for every $p \in M_0$, there is a *condition* b in the unfolding. Then, the unfolding incrementally extends Unf_Π by appending possible events e, adding a new condition to Unf_Π for every $b \in \mathsf{eff}(e)$. An event e can fire at a configuration C if the outcome $C \cup \{e\}$ is again a configuration (i. e., can still yield an executable transition sequence), and e is not a *cut-off* event. An event e is *cut-off* if its marking M, obtained when executing the configuration supporting e, equals the marking M' of an already generated configuration. The unfolding terminates if no more events can fire; it then represents all markings reachable in Σ. We define the size of an unfolding $|\mathit{Unf}_\Pi| := |B|$ as the number of its conditions. We assume a *search order* $\ll$, an order over configurations, constraining the firing order: at each point in the unfolding process, the $\ll$-minimal possible event e is added.

Figure 1 (right) illustrates some unfolding steps. Observe the exponential blow-up due to missing support for prevail conditions: any load event can choose to either use the initial occurrence of $(t = l)$, or any other occurrence generated by previous load events, so that the number of possibilities multiplies over the number of load events and thus packages.

Star-Topology Decoupling

Star-topology decoupling (STD) generates component states in a similar way as unfolding generates conditions. Yet, whereas conditions are facts, STD partitions the state variables into *factors*, and the atomic entities are assignments to one such factor. STD finds a partitioning $\mathcal{F}$ of state variables so that the interaction across factors takes the form of a star topology, where every cross-factor interaction involves a single *center factor* $C \in \mathcal{F}$, so that no direct interactions exist between the other factors, called *leaf factors* $\mathcal{L} := \mathcal{F} \setminus \{C\}$. We call assignments to C *center states*, and assignments to any $L \in \mathcal{L}$ *leaf states*. Both are *factor states*. In our example, one can set $C := \{t\}$ and $L_i := \{p_i\}$.

Decoupled search branches over *center actions* $\mathcal{A}^C$, i. e., actions that *affect* – that have an effect on – C. It enumerates the reachable leaf states separately for each leaf factor.

$\mathsf{center}(I^{\mathcal{F}}) = l \quad S^{\mathcal{L}}(I^{\mathcal{F}}) = \{(p_i = l) \xrightarrow{lop_i L} (p_i = T), \dots\}$

$\downarrow drR$

$\mathsf{center}(s^{\mathcal{F}}) = r \quad S^{\mathcal{L}}(s^{\mathcal{F}}) = \{(p_i = l), (p_i = T) \xrightarrow{ulp_i R} (p_i = r), \dots\}$

$\downarrow drL$

$\mathsf{center}(t^{\mathcal{F}}) = l \quad S^{\mathcal{L}}(t^{\mathcal{F}}) = \{(p_i = l), (p_i = T), (p_i = r), \dots\}$

Figure 2: The complete decoupled state space of our example. One decoupled state per row. Center states and transitions highlighted in blue. Transition within leaf state-sets illustrate how new leaf states are reached.

The *leaf actions* $\mathcal{A}^L$ are those which affect an $L \in \mathcal{L}$. A *decoupled state* $s^{\mathcal{F}}$ is a pair $s^{\mathcal{F}} = (\mathsf{center}(s^{\mathcal{F}}), S^{\mathcal{L}}(s^{\mathcal{F}}))$ of center state $\mathsf{center}(s^{\mathcal{F}})$ and set of leaf states $S^{\mathcal{L}}(s^{\mathcal{F}})$. We say that $s^{\mathcal{F}}$ satisfies a condition p, denoted $s^{\mathcal{F}} \models p$, if $\mathsf{center}(s^{\mathcal{F}}) \models p[C]$ and for every $L \in \mathcal{L}$ there exists an $s^L \in S^{\mathcal{L}}(s^{\mathcal{F}})$ such that $s^L \models p[L]$.

The decoupled initial state $I^{\mathcal{F}}$ has $\mathsf{center}(I^{\mathcal{F}}) = I[C]$. For each leaf L, first $I[L]$ is included into $S^{\mathcal{L}}(I^{\mathcal{F}})$; then the reachable leaf states are added into $S^{\mathcal{L}}(I^{\mathcal{F}})$, i. e., those s^L reachable from $I[L]$ via leaf actions $a \in \mathcal{A}^L \setminus \mathcal{A}^C$ whose center precondition is satisfied by $\mathsf{center}(I^{\mathcal{F}})$, i. e., $I[C] \models \mathsf{pre}(a)[C]$. In our example, see Figure 2, we first have the initial state facts $(p_i = l)$. The leaf states $\{(p_i = T)\} \in S^{\mathcal{L}}(I^{\mathcal{F}})$ are then reached via $lop_i L$ actions.

Applying a center action $a \in \mathcal{A}^C$ to a decoupled state $s^{\mathcal{F}}$ generates a successor $t^{\mathcal{F}}$ as follows. First, $\mathsf{center}(t^{\mathcal{F}}) = \mathsf{center}(s^{\mathcal{F}})[\![a]\!]$ and $S^{\mathcal{L}}(t^{\mathcal{F}}) = \{s^L[\![a]\!] \mid s^L \in S^{\mathcal{L}}(s^{\mathcal{F}}) \wedge s^L \models \mathsf{pre}(a)[L]\}$. Then, $S^{\mathcal{L}}(t^{\mathcal{F}})$ is augmented by those leaf states reachable via leaf actions whose center precondition is satisfied by $\mathsf{center}(t^{\mathcal{F}})$. In our example, applying $a = drR$ to $I^{\mathcal{F}}$, all $s^L \in S^{\mathcal{L}}(I^{\mathcal{F}})$ satisfy $\mathsf{pre}(a)[L]$ (which is empty), and additionally the leaf states $(p_i = r)$ become reachable.

The *decoupled state space* $\Theta_{\Pi}^{\mathcal{F}}$ incrementally expands center actions as described. Like for unfolding, we encode the expansion order in an ordering relation $\ll$, here an order over center-action paths, where each step considers the $\ll$-minimal possible expansion. An outcome state $t^{\mathcal{F}}$ is pruned (is a cut-off) if its *hypercube* $[t^{\mathcal{F}}]$ – the set of states formed from $\mathsf{center}(t^{\mathcal{F}})$ and leaf states in $S^{\mathcal{L}}(t^{\mathcal{F}})$ – is contained in the union of $[s^{\mathcal{F}}]$ for the previously generated decoupled states $s^{\mathcal{F}}$. Checking whether this is the case is co-**NP**-complete (Gnad and Hoffmann 2018). A cheap sufficient criterion is based on testing $[t^{\mathcal{F}}] \subseteq [s^{\mathcal{F}}]$ against individual previous $s^{\mathcal{F}}$. For all but one of the results we prove here (Theorem 5), that criterion is sufficient.

We define the size of $\Theta_{\Pi}^{\mathcal{F}}$ as the number of facts $|\Theta_{\Pi}^{\mathcal{F}}| := \sum_{s^{\mathcal{F}} \in \Theta_{\Pi}^{\mathcal{F}}} (|C| + \sum_{s^L \in S^{\mathcal{L}}(s^{\mathcal{F}})} |s^L|)$. We denote the number of decoupled states in $\Theta_{\Pi}^{\mathcal{F}}$ by $\#\Theta_{\Pi}^{\mathcal{F}}$.

Results Overview

We consider the following three dimensions:

(i) Presence or absence of prevail conditions.

(ii) Presence or absence of multi-variable components.

(iii) Compatibility, or lack thereof, of the search orders $\ll$.

Dimension (i) concerns the planning tasks Π. We denote the class of Π *without prevail conditions* by "-P", and the class of all (arbitrary) Π, not making that restriction, by "+P". For dimension (ii), we denote by -M the restriction where the factorization $\mathcal{F}$ *may not contain multi-variable components*, and by +M the class of all $\mathcal{F}$ not imposing this limitation.

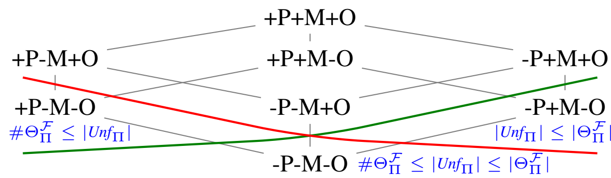

Figure 3: Subsumption hierarchy and results overview. Above the green line, STD can yield exponentially smaller representation size, below that line it cannot. Above the red line, unfolding can yield exponentially smaller representation size, below it cannot.

Regarding dimension (iii), note that we are not interested in heuristic search here; the target is to build a complete representation of reachability (the decoupled state space in STD, a complete prefix in unfolding). Still, the order of expansions can significantly impact representation size, potentially incurring or avoiding exponential blow-ups as we shall see. We capture this in terms of the search orders $\ll$. We say that a pair of search orders $(\ll_U, \ll_D)$ for unfolding respectively STD is *compatible* if *(O1)* $\ll_U$ always orders new leaf events before new center events, and *(O2)* $\ll_D$ and $\ll_U$ agree on center paths, i. e., denoting by $Q|_C$ the restriction of a configuration Q to center events, $Q_1|_C \ll_U Q_2|_C$ iff $\pi^C(Q_1) \ll_D \pi^C(Q_2)$ for all valid sequencings of (the partially ordered) $Q_1|_C$ and $Q_2|_C$ into center paths $\pi^C(Q_1)$ and $\pi^C(Q_2)$. In other words, the only degree of freedom in $\ll_U$, over $\ll_D$, is the relative ordering of leaf events. We denote that restriction by -O, and the unrestricted case by +O. Note that (O1) mimics the factor-state generation order in STD, where after adding a center action, all reachable leaf states are added prior to considering the next center action.

Figure 3 gives an overview of the hierarchy of sub-classes induced by dimensions (i) – (iii), and the associated reachability representation size results. In this hierarchy, exponential separations are inherited upwards, to more permissive classes, as separating example families get preserved; while domination properties are inherited downwards, to more restricted classes, as the required prerequisites are preserved.

The next section shows our separation theorems. We show that for the class +P-M-O there exist planning task families where STD results in exponentially smaller reachability representations. We show that, within -P+M-O, unfolding size can be exponentially smaller. For incompatible orders -P-M+O, we show separations in both directions.

Afterwards, we show our domination theorems. Within +P-M-O, the number of decoupled states is always at most as large as the unfolding, $\#\Theta_{\Pi}^{\mathcal{F}} \leq |Unf_{\Pi}|$. On the other hand, within -P+M-O, the unfolding is at most as large as the decoupled state space, $|Unf_{\Pi}| \leq |\Theta_{\Pi}^{\mathcal{F}}|$. As $\#\Theta_{\Pi}^{\mathcal{F}}$ and $|\Theta_{\Pi}^{\mathcal{F}}|$ are polynomially related given -M, with downward inheritance in particular we get that, in the most restricted class -P-M-O, STD size and unfolding size are polynomially related.

Separation Theorems

We show exponential separations between STD and unfolding. The planning task families Π^n in the following theorems have size linear in n.

Theorem 1 *There exists a family of tasks Π^n in +P, with factorings in -M and search orders in -O, where $|\Theta^{\mathcal{F}}_{\Pi^n}|$ is polynomial in n while $|Unf_{\Pi^n}|$ is exponential in n.*

Our running example is such a family Π^n. There are only 3 decoupled states, independently of n; the number of factor states is linear in n. The number of conditions in the unfolding is exponential in n, because all possible combinations of, e. g., $lop_i L$ actions are enumerated in the initial state.

The so-called *place-replication* method in Petri nets encodes prevail conditions differently, with copies of the prevail places (Baldan et al. 2012). In our example, though, this incurs the same blow-up. Contextual Petri nets (Baldan et al. 2012) have built-in support for prevail conditions ("read arcs"), yet contextual unfoldings must keep track of "event histories" which again incur the same blow-up.

Theorem 2 *There exists a family of tasks Π^n in -P, with factorings in +M and search orders in -O, where $\#\Theta^{\mathcal{F}}_{\Pi^n}$ is exponential in n while $|Unf_{\Pi^n}|$ is polynomial in n.*

Such example families can be constructed through permutability (*concurrency*, in Petri net parlance) within factors. For example, scaling the number of trucks in logistics, if all truck variables are in the center, then STD enumerates all possible interleavings of truck drives. Unfolding expands the trucks separately, avoiding that blow-up.

Theorem 3 *There exists a family of tasks Π^n in -P, with factorings in -M and search orders in +O, where $|\Theta^{\mathcal{F}}_{\Pi^n}|$ is polynomial in n while $|Unf_{\Pi^n}|$ is exponential in n.*

Theorem 4 *There exists a family of tasks Π^n in -P, with factorings in -M and search orders in +O, where $\#\Theta^{\mathcal{F}}_{\Pi^n}$ is exponential in n while $|Unf_{\Pi^n}|$ is polynomial in n.*

For both theorems, we construct example families and search orders where one technique enters a part of the search space that is exponential in n, while the other technique takes a "short-cut", entering a part of the search space that allows to capture complete reachability with polynomial representation size. For Theorem 3, the task family is constructed so that the unfolding search has a detrimental priority to expand center actions – even though leaf actions could be expanded, violating (O1) – missing the "short-cut" offered by leaf actions after a single center action has been applied. For Theorem 4, vice versa, complying with (O1) may lead to an exponential disadvantage, due to generating the leaf preconditions of center actions causing a blow-up.

Domination Theorems

We now show domination results between the size of the decoupled state space and that of the unfolding.

Intuitively, in singleton-component factorings, there is no concurrency within factors, so unfolding can only exploit the concurrency inherent in the factoring. Formally:

Theorem 5 *For Π in +P, factorings in -M, and search orders in -O, with hypercube pruning $\#\Theta^{\mathcal{F}}_{\Pi} \leq |Unf_{\Pi}|$.*

The proof consists of Lemmas 2 and 3 in the appendix. Lemma 2 considers non-pruned/non-cut versions of STD and unfolding, i. e., the infinite structures that arise without pruning. It shows that the action and factor-state occurrences in $\Theta^{\mathcal{F}}_{\Pi}$ can be injectively mapped to corresponding events and conditions in Unf_{Π}. This is because, in singleton-component factorings, factor states are singleton facts (variable/value pairs), corresponding exactly to the places and conditions in the Petri net formulation. Whenever an action occurence in $\Theta^{\mathcal{F}}_{\Pi}$ generates new factor states, a corresponding event in Unf_{Π} generates corresponding conditions.

Lemma 3 shows that, if $s^{\mathcal{F}}$ is not pruned by hypercube pruning in $\Theta^{\mathcal{F}}_{\Pi}$, then it contains a non-cut-off event in Unf_{Π}. Namely, say action occurrence a in $s^{\mathcal{F}}$ generates a state not contained in any previous hypercube, and say a is mapped to e as per Lemma 2. Then e is not a cut-off: with compatibility of $\ll$, the only additional conditions generated in Unf_{Π} are duplicates of prevail conditions, and the only additional events are duplicates of actions consuming these conditions.

Our next result is perhaps more surprising. The exponential advantage of STD disappears without prevail conditions:

Theorem 6 *For Π in -P, factorings in +M, and search orders in -O, $|Unf_{\Pi}| \leq |\Theta^{\mathcal{F}}_{\Pi}|$.*

The proof (in the appendix) consists of two parts. The first part considers the non-pruned versions of STD and unfolding, and shows that the factor-state occurrences in $\Theta^{\mathcal{F}}_{\Pi}$ can be surjectively mapped to corresponding *factor co-sets* in Unf_{Π}: jointly reachable conditions over the variables of a factor. During the construction of Unf_{Π} and $\Theta^{\mathcal{F}}_{\Pi}$, for every new event e in Unf_{Π}, every new factor co-set supported by e is matched by corresponding new factor states in $\Theta^{\mathcal{F}}_{\Pi}$. The crucial part of the argument is that, in the absence of prevail conditions, all new conditions b generated by e correspond to a factor-state change in the planning task, matched by the generation of a new factor state in $\Theta^{\mathcal{F}}_{\Pi}$.

The second part of the proof observes that, for any event e, corresponding new factor state occurrences p in $\Theta^{\mathcal{F}}_{\Pi}$ map to factor co-sets including the new conditions added by e. The factor co-sets mapped to are different for every event. Further, at any point in the construction, with compatibility of $\ll$, the current $\Theta^{\mathcal{F}}_{\Pi}$ prefix cannot represent states not represented in the Unf_{Π} prefix. Thus, if e is a non-cut-off event, at least one decoupled state $s^{\mathcal{F}}$ containing the new factor-state occurrences p is not pruned by hypercube pruning.

Corollary 1 *For Π in -P, factorings in -M, and search orders in -O, with hypercube pruning $\#\Theta^{\mathcal{F}}_{\Pi} \leq |Unf_{\Pi}| \leq |\Theta^{\mathcal{F}}_{\Pi}|$.*

Conclusion

Our results completely characterize the possibility of exponential size differences, or lack thereof, between STD and unfolding as a function of three major dimensions. A major question for the future is whether, guided by these results, the strengths of STD could be combined with those of unfolding. One could use unfolding inside the STD factors, which should dominate both algorithms in search space size, at the expense of worst-case exponential reachability tests within factors. Other thinkable combinations include special-case handling of prevail conditions, for star-shape dependencies, within unfolding techniques.

Appendix: Technical Background Details

We spell out the concepts previously only outlined, and we give additional notations as needed in our proofs.

Petri-Net Unfolding

Our definitions loosely follow Bonet et al. (2014). A *net* N is a tuple $N = \langle P, T, F \rangle$, where P and T are sets of *places* and *transitions*. $F \subseteq (P \times T) \cup (T \times P)$ is the *flow relation*. For $z \in P \cup T$, we denote $\mathsf{pre}(z) := \{y \mid (y, z) \in F\}$ and $\mathsf{eff}(z) := \{y \mid (z, y) \in F\}$. For $Z \subset P \cup T$, we denote $\mathsf{pre}(Z) := \bigcup_{z \in Z} \mathsf{pre}(z)$ and $\mathsf{eff}(Z) := \bigcup_{z \in Z} \mathsf{eff}(z)$. A set of places $M \subseteq P$ is called a *marking*. A Petri net $\Sigma = \langle N, M_0 \rangle$ is a pair of a net $N = \langle P, T, F \rangle$ and *initial marking* $M_0 \subseteq P$. By $\preceq$, we denote the reflexive transitive closure of the flow relation F. Two nodes $y, y' \in P \cup T$ are *in conflict*, denoted $y \# y'$, if there exist distinct $t, t' \in T$ s.t. $\mathsf{pre}(t) \cap \mathsf{pre}(t') \neq \emptyset$, $t \preceq y$, and $t' \preceq y'$. Two nodes $y, y' \in P \cup T$ are *concurrent*, denoted $y \parallel y'$, if neither $y \# y'$ nor $y \preceq y'$ nor $y' \preceq y$.

The unfolding procedure builds a *branching process*, which is an *occurrence net* labeled with the places and transitions in Σ. An occurrence net $ON = \langle B, E, G \rangle$ is a net where B and E are called *conditions* and *events*, corresponding to places and transitions in a net. Occurrence nets have the following properties: they are acyclic, i. e., $\preceq$ is a partial order; for every $b \in B : |\mathsf{pre}(b)| \leq 1$; for every $y \in B \cup E$, $\neg(y \# y)$ and there are finitely many y' s.t. $y' \prec y$, where $\prec$ is the transitive closure of G. $\prec$ is called the *causality relation*, and an event f with $f \prec e$ is called a causal predecessor of e. $Min(ON)$ is the set of $\prec$-minimal elements of $B \cup E$. A branching process Δ of a Petri net Σ is a pair $\Delta = \langle ON, \phi \rangle$ of an occurrence net ON and a homomorphism $\phi : B \cup E \to P \cup T$ specifying the labels.

A set of conditions D is called a *co-set* if for all $d \neq d' \in D : d \parallel d'$. A set of events $C \subseteq E$ is *causally closed* if for every $e \in C$, $f \prec e$ implies $f \in C$. A *configuration* C is a finite set of events that is causally closed and free of conflicts ($\forall e, f \in C : \neg(e \# f)$). By $[e] := \{f \mid f \preceq e\}$ we denote the *local configuration* of an event $e \in E$. For a configuration C, $Mark(C) := \phi((Min(ON) \cup \mathsf{eff}(C)) \setminus \mathsf{pre}(C))$ is a reachable marking of Σ. Intuitively, a configuration corresponds to a partially ordered plan.

An event e is a *cut-off* if there exists a configuration C in Δ such that $Mark(C) = Mark([e])$. An event $e \in E$ labeled with a transition t is a possible *extension* of a configuration C in Δ if $C \cup \{e\}$ is a configuration, and there exists a co-set D in Δ such that no event in $\mathsf{pre}(D)$ is a cut-off, $|D| = |\mathsf{pre}(t)|$, $\phi(D) = \mathsf{pre}(t)$, and Δ contains no event e' with $\mathsf{pre}(e') = D$ where $\phi(e') = t$. We then say that e *fires* in C.

The *unfolding* process for Σ incrementally builds a branching process called a *complete prefix*, denoted Unf_Σ. The process starts from $Min(ON)$, and adds possible extensions while ones exist. The extensions e are added according to an order $\ll$ over their local configurations $[e]$. In each step, the $\ll$-minimal event e is considered. If e is not a cut-off, then new instances of $\mathsf{eff}(\phi(e))$ are added to Unf_Σ. Upon termination, all reachable markings of Σ are represented by a configuration in Unf_Σ (McMillan 1992).

If $\ll$ is a well-founded order and satisfies certain conditions (see Def. 3 in Bonet et al. (2014)), then the number of non-cut-off events in Unf_Σ is upper-bounded by the number of reachable markings in Σ. We will consider such $\ll$ throughout. We define the size of Unf_Σ as $|Unf_\Sigma| := |B|$.

A planning task $\Pi = \langle \mathcal{V}, \mathcal{A}, I, G \rangle$ can be encoded as a Petri net $\Sigma(\Pi) = \langle \langle P, T, F \rangle, M_0 \rangle$. Facts are encoded as places. Actions a are encoded as transitions t with $\mathsf{pre}(t) = \mathsf{pre}(a)$ and $\mathsf{eff}(t) = \mathsf{eff}(a)$, adding redundant effects $\mathsf{eff}(a)[v] = \mathsf{pre}(a)[v]$ for prevail conditions. We assume this encoding throughout, and refer to its unfolding as the *unfolding of* Π, denoted Unf_Π. We identify facts with places, actions with transitions, and (partial) states with markings.

Star-Topology Decoupling (STD)

Given a planning task Π, a variable partitioning $\mathcal{F}$ is a *star factoring* if $|\mathcal{F}| > 1$ and there exists $C \in \mathcal{F}$ such that, for every action a where $vars(\mathsf{eff}(a)) \cap C = \emptyset$, there exists $F \in \mathcal{F}$ with $vars(\mathsf{eff}(a)) \subseteq F$ and $vars(\mathsf{pre}(a)) \subseteq F \cup C$.

The set of actions affecting a leaf $L \in \mathcal{L} := \mathcal{F} \setminus \{C\}$ is denoted $\mathcal{A}^L$, the set of all leaf actions is denoted $\mathcal{A}^\mathcal{L}$. We refer to sequences $\pi^C = \langle a_1^C, \dots, a_n^C \rangle$ of center actions $a_i^C \in \mathcal{A}^C$ as *center paths*, and sequences $\pi^L = \langle a_1^L, \dots, a_n^L \rangle$ of leaf actions $a_i^L \in \mathcal{A}^L$ as *leaf paths*. The set of states of a leaf L is denoted S^L, the set of all leaf states is denoted $S^\mathcal{L}$.

A *decoupled state space* given Π and $\mathcal{F}$ is a labeled transition system $\Theta_\Pi^\mathcal{F} = \langle S^\mathcal{F}, \mathcal{A}^C, T^\mathcal{F}, I^\mathcal{F} \rangle$, built by starting from $I^\mathcal{F}$ and incrementally adding non-pruned transitions and outcome states $t^\mathcal{F}$. $S^\mathcal{F}$ is the set of decoupled states. The center actions $a^C \in \mathcal{A}^C$ label the transitions $T^\mathcal{F}$. We have $\langle s^\mathcal{F}, a^C, t^\mathcal{F} \rangle \in T^\mathcal{F}$ iff $s^\mathcal{F}, t^\mathcal{F} \in S^\mathcal{F}$, $s^\mathcal{F} \models \mathsf{pre}(a^C)$, and $s^\mathcal{F}[\![a^C]\!] = t^\mathcal{F}$. Here, the outcome $s^\mathcal{F}[\![a^C]\!]$ of applying a^C to $s^\mathcal{F}$ is defined by $\mathsf{center}(t^\mathcal{F}) := \mathsf{center}(s^\mathcal{F})[\![a^C]\!]$ and $S^\mathcal{L}(t^\mathcal{F}) := \bigcup_{i=0}^\infty S^\mathcal{L}(t^\mathcal{F})^i$ where $S^\mathcal{L}(t^\mathcal{F})^0 := \{s^L[\![a^C]\!] \mid \exists L \in \mathcal{L}, s^L \in S^\mathcal{L}(s^\mathcal{F}) \cap S^L : s^L \models \mathsf{pre}(a^C)[L]\}$ and $S^\mathcal{L}(t^\mathcal{F})^{i+1} := \{s^L[\![a^L]\!] \mid \exists L \in \mathcal{L}, s^L \in S^\mathcal{L}(t^\mathcal{F})^i \cap S^L, a^L \in \mathcal{A}^L \setminus \mathcal{A}^C : \mathsf{center}(t^\mathcal{F}) \models \mathsf{pre}(a^L)[C], s^L \models \mathsf{pre}(a^L)[L]\} \setminus \bigcup_{j=0}^i S^\mathcal{L}(t^\mathcal{F})^j$. The initial decoupled state $I^\mathcal{F}$ is defined similarly by $\mathsf{center}(I^\mathcal{F}) := I[C]$ and $S^\mathcal{L}(I^\mathcal{F}) := \bigcup_{i=0}^\infty S^\mathcal{L}(I^\mathcal{F})^i$ where $S^\mathcal{L}(I^\mathcal{F})^0 := \{I[L] \mid L \in \mathcal{L}\}$. The center path on which a decoupled state $s^\mathcal{F}$ is reached from $I^\mathcal{F}$ in $\Theta_\Pi^\mathcal{F}$ is denoted $\pi^C(s^\mathcal{F})$.

Essentially, state transitions in $\Theta_\Pi^\mathcal{F}$ advance the center state by a^C, and advance the set of reached leaf states using those leaf actions enabled by the new center state. This corresponds to an unfolding (sub-)process over factor states that adds one center event and iteratively adds all leaf events enabled by that center event.

Appendix: Proofs

We give the full proofs of our theorems, covering first the separation results then the domination results.

Separation Theorems

Theorem 1 *There exists a family of tasks Π^n in +P, with factorings in -M and search orders in -O, where $|\Theta^{\mathcal{F}}_{\Pi^n}|$ is polynomial in n while $|Unf_{\Pi^n}|$ is exponential in n.*

Proof: One family as claimed is our illustrative running example, $\Pi^n = \langle \mathcal{V}^n, \mathcal{A}^n, I^n, G^n \rangle$ defined as follows. $\mathcal{V}^n = \{t, p_1, \dots, p_n\}$ where $\mathcal{D}(t) = \{l, r\}$ and $\mathcal{D}(p_i) = \{l, r, T\}$. The initial state is $I^n = \{t = l, p_1 = l, \dots, p_n = l\}$. The goal does not matter here. The actions are $\mathcal{A}^n = \{drive(x, y) \mid (x, y) \in \{(l, r), (r, l)\}\} \cup \{load(i, z), unload(i, z) \mid 1 \leq i \leq n, z \in \{l, r\}\}$ where $\mathsf{pre}(drive(x, y)) = \{t = x\}$, $\mathsf{eff}(drive(x, y)) = \{t = y\}$, $\mathsf{pre}(load(i, z)) = \{t = z, p_i = z\}$, $\mathsf{eff}(load(i, z)) = \{p_i = T\}$, and $\mathsf{pre}(unload(i, z)) = \{t = z, p_i = T\}$, $\mathsf{eff}(unload(i, z)) = \{p_i = z\}$.

Assume the factoring $\mathcal{F}$ with center $C = \{t\}$ and leaves $\mathcal{L} = \{\{p_1\}, \dots, \{p_n\}\}$. The number of decoupled states is $\#\Theta^{\mathcal{F}}_{\Pi^n} = 3$ as illustrated in Figure 2: After applying $drive(l, r)$ and $drive(r, l)$, all leaf states are reached. $\Theta^{\mathcal{F}}_{\Pi}$ contains $|\Theta^{\mathcal{F}}_{\Pi}| = 3 + 2n + 3n + 3n = 8n + 3$ factor states.

The size of the unfolding prefix $|Unf_{\Sigma}|$, however, is exponential in n. Any $load(i, l)$ event that fires in the initial state consumes an instance of the condition $(t = l)$, and produces a new instance of that condition. As the consumed instance can be any instance produced beforehand, the number of instances in the Petri net doubles in each step. □

Theorem 2 *There exists a family of tasks Π^n in -P, with factorings in +M and search orders in -O, where $\#\Theta^{\mathcal{F}}_{\Pi^n}$ is exponential in n while $|Unf_{\Pi^n}|$ is polynomial in n.*

We prove the following stronger claim:

Lemma 1 *There exists a family of tasks Π^n in -P, with factorings in +M and search orders in -O, where $\#\Theta^{\mathcal{F}}_{\Pi^n}$ is exponential in n for every family of star factorings $\mathcal{F}^n$, while $|Unf_{\Pi^n}|$ is polynomial in n.*

Proof: Consider $\Pi^n = \langle \mathcal{V}^n, \mathcal{A}^n, I^n, G^n \rangle$ as follows. $\mathcal{V}^n = \{v_1, \dots, v_n\}$, where $\mathcal{D}(v_i) = \{0, 1, 2\}$ for $1 \leq i \leq n$. The initial state is $I^n = \{v_1 = 0, \dots, v_n = 0\}$. The actions are $\mathcal{A}^n = \{a_0, a^{12}_i, a^{12}_{ij} \mid 1 \leq i, j \leq n\}$ where $\mathsf{pre}(a_0) = \{v_1 = 0, \dots, v_n = 0\}$ and $\mathsf{eff}(a_0) = \{v_1 = 1, \dots, v_n = 1\}$; $\mathsf{pre}(a^{12}_i) = \{v_i = 1\}$ and $\mathsf{eff}(a^{12}_i) = \{v_i = 2\}$; $\mathsf{pre}(a^{12}_{ij}) = \{v_i = 0, v_j = 1\}$ and $\mathsf{eff}(a^{12}_{ij}) = \{v_i = 2, v_j = 2\}$.

The unfolding prefix Unf_{Σ} has size $|Unf_{\Sigma}| = 3n$, with a single condition b for every reachable fact. $\#\Theta^{\mathcal{F}}_{\Pi^n}$ is exponential in n as claimed. Observe that the a^{12}_{ij} actions have an unreachable precondition, yet their presence means that, in any star factoring, there can be at most one leaf: if there were two leaves F_i and F_j containing v_i and v_j respectively, then the action a^{12}_{ij} would incur a direct dependency across F_i and F_j, in contradiction. Thus, for any family $\mathcal{F}^n = \{C^n, L^n\}$ of star factorings (where L^n may not be present for some values of n), $\max(|C^n|, |L^n|) \in \Omega(n)$. So $\#\Theta^{\mathcal{F}}_{\Pi^n}$ is exponential in n since it has to enumerate all applications of a^{12}_i actions for a linear number of variables v_i. □

Theorem 3 *There exists a family of tasks Π^n in -P, with factorings in -M and search orders in +O, where $|\Theta^{\mathcal{F}}_{\Pi^n}|$ is polynomial in n while $|Unf_{\Pi^n}|$ is exponential in n.*

Proof: We construct a task family $\Pi^n = \langle \mathcal{V}^n, \mathcal{A}^n, I^n, G^n \rangle$ as follows. The variables are $\mathcal{V}^n = \{c, l_1, \dots, l_n\}$, where $\mathcal{D}(c) = \{0, 1\}$ and $\mathcal{D}(l_i) = \{0, 1, 2\}$. The initial state is $I^n = \{c = 0, l_1 = 0, \dots, l_n = 0\}$. The actions are $\mathcal{A}^n = \{a^C_{01all2}, a^C_{10}, a^C_{01i01}, a^L_{i20}, a^L_{i21} \mid 1 \leq i \leq n\}$. The action preconditions and effects are: $\mathsf{pre}(a^C_{01all2}) = \{c = 0, l_1 = 0, \dots, l_n = 0\}$ and $\mathsf{eff}(a^C_{01all2}) = \{c = 1, l_1 = 2, \dots, l_n = 2\}$; $\mathsf{pre}(a^C_{10}) = \{c = 1\}$ and $\mathsf{eff}(a^C_{10}) = \{c = 0\}$; $\mathsf{pre}(a^C_{01i01}) = \{c = 0, l_i = 0\}$ and $\mathsf{eff}(a^C_{01i01}) = \{c = 1, l_i = 1\}$; $\mathsf{pre}(a^L_{i20}) = \{l_i = 2\}$ and $\mathsf{eff}(a^L_{i20}) = \{l_i = 0\}$; $\mathsf{pre}(a^L_{i21}) = \{l_i = 2\}$ and $\mathsf{eff}(a^L_{i21}) = \{l_i = 1\}$.

Assume the factoring $\mathcal{F}$ with center $C = \{c\}$ and leaves $\mathcal{L} = \{\{l_1\}, \dots, \{l_n\}\}$. After applying a^C_{01all2}, exploration of the leaf actions a^L_{i20} and a^L_{i21} reaches all variable values and thus a compact representation of reachability. We construct the search orders $\ll$ so that STD finds this compact representation, but unfolding does not.

We postpone configurations containing leaf events until no more center-only configurations are available (thus violating constraint (O1) of compatible orders); and we constrain the order on center actions to start with the sequence $\langle a^C_{01all2}, a^C_{10} \rangle$. Precisely: if C_l contains an event e labeled by $\phi(e) = a \in \mathcal{A}^{\mathcal{L}} \setminus \mathcal{A}^C$, but C does not contain such an event, then $C \ll C_l$; denoting $C_1 = \{a^C_{01all2}\}$ and $C_2 = \{a^C_{01all2}, a^C_{10}\}$, we set $C_1 \ll C_2 \ll C \in Unf_{\Sigma} \setminus \{C_1, C_2\}$. Inside these constraints, $\ll$ can be arbitrary.

With this search order, STD first generates $s^{\mathcal{F}} = I^{\mathcal{F}}[\![a^C_{01all2}]\!]$, where application of the leaf actions a^L_{i20} and a^L_{i21} reaches all values of the leaf variables. Then STD generates $t^{\mathcal{F}} = s^{\mathcal{F}}[\![a^C_{10}]\!]$. After that, the process stops: $s^{\mathcal{F}}$ covers everything with center state $c = 1$, $t^{\mathcal{F}}$ covers everything with center state $c = 0$. The decoupled state space has $\#\Theta^{\mathcal{F}}_{\Pi^n} = 3$ states, and thus polynomial size.

The unfolding prefix Unf_{Σ}, however, has size exponential in n. The unfolding starts with the center events a^C_{01all2} and a^C_{10}. Thereafter, given $\ll$, it prefers to explore the center events a^C_{01i01} rather than the leaf events a^L_{i2x}. The unfolding thus has to set each leaf variable separately to 1, using a^C_{01i01}. Every step a^C_{01i01} sets c to 1, and must be followed by a^C_{10} setting c back to 0. In doing so, a^C_{01i01} consumes an instance of the condition $c = 0$, and a^C_{10} generates a new instance of that condition. As the consumed instance can be any instance produced beforehand, the number of instances in the Petri net doubles in each step. □

Theorem 4 *There exists a family of tasks Π^n in -P, with factorings in -M and search orders in +O, where $\#\Theta^{\mathcal{F}}_{\Pi^n}$ is exponential in n while $|Unf_{\Pi^n}|$ is polynomial in n.*

Proof: We adapt the task Π^n used in the proof of Theorem 3. We add a new variable l with domain $\{0, 1\}$ and initial value 0. We include a new action a^L_{01} with precondition $\{l = 0\}$ and effect $\{l = 1\}$. We add the fact $l = 1$ into the preconditions of all actions a^C_{01i01}, and we add $l = 0$ into the effects of these actions. In this modified task, to enter the exponential part of the search space, the leaf action a^L_{01} must be applied first. STD always applies leaf actions first. If we violate (O1) however, unfolding can avoid this.

Precisely, we constrain $\ll$ to order configurations containing (an event labeled) a^C_{01all2} behind all configurations containing any of a^C_{01i01}; and to order configurations containing a^L_{01} behind all other configurations. STD then expands the leaf action a^L_{01} at $I^{\mathcal{F}}$, enabling the a^C_{01i01} actions, thus forcing the search into exploring the search sub-space using these actions. This sub-space contains a different decoupled state for every subset of leaf states so is exponentially large. Yet unfolding prefers to do anything other than adding a^L_{01}, so initially adds a^C_{01all2} and then expands the a^L_{i20} and a^L_{i21} actions, which together with a single a^C_{10} event and a single a^L_{01} event represent all reachable markings. $\square$

Domination Theorems

We first analyze the case of singleton components, then that where there are no prevail conditions. Each analysis is decomposed into two steps, first showing a correspondence across hypothetical *non-pruned* infinite structures, then showing that this correspondence persists in the actual structures. The non-pruned Unf_Π expands cut-off events. The non-pruned $\Theta^{\mathcal{F}}_\Pi$ does not prune decoupled states, and within each decoupled state does not do duplicate checking across leaf-factor states. Note that these structures can be built incrementally by choosing applicable center and leaf expansions non-deterministically.

By $\hat{a}$ we denote an *occurrence* of an action a in $\Theta^{\mathcal{F}}_\Pi$, i. e., center action $a \in \mathcal{A}^C$ inducing a new decoupled state $s^{\mathcal{F}}$, or a leaf action $a \in \mathcal{A}^{\mathcal{L}}$ inducing a leaf state in a decoupled state $s^{\mathcal{F}}$. By $\hat{p}$ we denote an occurrence of a factor state p, i. e., a center state or a reached leaf state in a decoupled state.

Theorem 5 *For Π in +P, factorings in -M, and search orders in -O, with hypercube pruning $\#\Theta^{\mathcal{F}}_\Pi \leq |Unf_\Pi|$.*

The proof shows how to embed $\Theta^{\mathcal{F}}_\Pi$ into Unf_Π. Theorem 5 follows directly from the following two Lemmas.

Lemma 2 *Let Π be a task in +P, and $\mathcal{F}$ a factoring in -M. Let $\Theta^{\mathcal{F}}_\Pi$ and Unf_Π be non-pruned. Then there is an injective mapping f from action and factor-state occurrences $\hat{a}$ and $\hat{p}$ in $\Theta^{\mathcal{F}}_\Pi$ to events and conditions in Unf_Π, where $\phi(f(\hat{a})) = a$ and $\phi(f(\hat{p})) = p$ and, if $\hat{p}$ is generated by $\hat{a}$ in $\Theta^{\mathcal{F}}_\Pi$, then $f(\hat{p})$ is added by $f(\hat{a})$ in Unf_Π.*

Proof: Let $\hat{p}$ be a factor-state occurrence in $s^{\mathcal{F}}$. The proof is by induction over the length of the shortest action sequence $\pi = \langle a_1, \dots, a_n \rangle$ to a state $s \in [s^{\mathcal{F}}]$ where $s \models p$ and, if $n > 0$, $p \subseteq \mathsf{eff}(a_n)$. Such a π exists for every $\hat{p}$.

The base case $n = 0$ captures exactly those $\hat{p}$ not generated by an action. These $\hat{p}$ form the *non-expanded initial decoupled state*, denoted $I^{\mathcal{F}}_0$, where no leaf action has yet been applied, i. e., the factor state occurrencess $\hat{p}$ are $I[C]$ and $I[L]$ for every $L \in \mathcal{L}$. As $\mathcal{F}$ is singleton-component, these $\hat{p}$ are simply the initial-state facts p. By definition of Unf_Π, for every such p there is $b \in Unf_\Pi$ where $\phi(b) = p$. We can define the desired mapping f_0 accordingly.

For the inductive case, we have $\langle a_1, \dots, a_n \rangle$ and $p \subseteq \mathsf{eff}(a_n)$. By induction hypothesis, the factor states generated along $\pi_{n-1} := \langle a_1, \dots, a_{n-1} \rangle$ can be mapped by an injective function f_{n-1} to corresponding conditions in Unf_Π, generated by a corresponding configuration C_{n-1}. Therefore, a new event e_n with $\phi(e_n) = a_n$ is a possible extension of C_{n-1}. In $\Theta^{\mathcal{F}}_\Pi$, $\hat{a_n}$ generates a new occurrence of factor state $p := \mathsf{eff}(a)[F]$ for every factor F it affects. As $\mathcal{F}$ is singleton-component, each p is a fact ($v = \mathsf{eff}(a)[v]$). In the unfolding, e_n generates a new condition b in Unf_Σ for every $p \in \mathsf{eff}(a)$, where $\phi(b) = p$. We can thus extend f_{n-1} to a mapping f_n as desired. $\square$

Lemma 3 *Let Π be a task in +P, and $\mathcal{F}$ a factoring in -M. Let $\Theta^{\mathcal{F}}_\Pi$ and Unf_Π be generated using compatible orders $\ll$, with hypercube pruning for $\Theta^{\mathcal{F}}_\Pi$. Then decoupled states in $\Theta^{\mathcal{F}}_\Pi$ can be injectively mapped to non-cut-off events in Unf_Π.*

Proof: Consider first $\Theta^{\mathcal{F}}_\Pi$ and Unf_Π generated using identical orders, iteratively as follows. In each expansion step, first add a new action occurrence $\hat{a}$ and its new factor state occurrences $\hat{p}$ to $\Theta^{\mathcal{F}}_\Pi$. Then add all corresponding non-cut-off events e, and conditions c, to Unf_Π as per Lemma 2.

Let $s^{\mathcal{F}}$ in $\Theta^{\mathcal{F}}_\Pi$ be arbitrary. Denote by D the prefix of $\Theta^{\mathcal{F}}_\Pi$ generated prior to $s^{\mathcal{F}}$, and by U the prefix of Unf_Π. As $s^{\mathcal{F}}$ is not pruned by hypercube pruning, there exists $s \in [s^{\mathcal{F}}]$ not contained in $[t^{\mathcal{F}}]$ for any $t^{\mathcal{F}}$ in D. Let $\hat{a}$ be an action occurrence in $s^{\mathcal{F}}$ that generates s, i. e., either the center action application leading to $s^{\mathcal{F}}$ or a leaf action application setting a leaf L to $s[L]$. Then the corresponding event e is a non-cut-off event in Unf_Π, because U cannot contain reachable markings (states) not contained in D. Clearly, e is different for every $s^{\mathcal{F}}$, allowing an injective mapping.

Consider now $\Theta^{\mathcal{F}}_\Pi$ and Unf_Π generated using compatible orders. The only additional degree of freedom then is the relative ordering of leaf event expansion within decoupled states. This does not affect the above arguments. $\square$

Theorem 6 *For Π in -P, factorings in +M, and search orders in -O, $|Unf_\Pi| \leq |\Theta^{\mathcal{F}}_\Pi|$.*

The proof shows how to surjectively map factor states in $\Theta^{\mathcal{F}}_\Pi$ to *factor co-sets* in Unf_Π, showing that the number of factor states is at least as high as that of factor co-sets.

We use the following notations. A factor co-set P is a co-set where $vars(\phi(P)) = F$ for some $F \in \mathcal{F}$. We write $P[F]$ to indicate the factor F concerned, and given an arbitrary co-set Q we write $Q[F]$ for the restriction of Q to conditions over the variables F. We write $[Q]$ for the configuration supporting Q, and we write $[Q]^C$ for the restriction of $[Q]$ to center events. We write $p[F]$ to indicate that a factor state p is over factor F. We say that a sequence π^C of center actions *extends* a partial order C over center actions if there is a sub-sequence of π^C that is a sequencing of C.

Proof: The proof has two parts: first, we consider the non-pruned $\Theta^{\mathcal{F}}_\Pi$ and Unf_Π; then we analyze cut-off events vs. hypercube pruning.

For the first part, we prove that *(*) there is a surjective mapping g where (a) for every $\hat{p}$, $\phi(g(\hat{p})) = p$; (b) for every* *co-set Q, there is at least one $s^{\mathcal{F}}$ where $\pi^C(s^{\mathcal{F}})$ extends $[Q]^C$; and (c) for every such $s^{\mathcal{F}}$ and every F, there is $\hat{p}[F]$ in $s^{\mathcal{F}}$ where $g(\hat{p}[F]) \supseteq Q[F]$.*

We prove (*) by structural induction over an incremental construction of $\Theta_\Pi^{\mathcal{F}}$ alongside the construction of Unf_Π. D and U denote the current prefix of $\Theta_\Pi^{\mathcal{F}}$ and Unf_Π respectively, during the construction.

The induction base case is simple: U is then the set $Min(ON)$ of $\prec$-minimal elements of $B \cup E$. This contains exactly one condition b for every state variable v, with $\phi(b) = I[v]$. The factor co-sets $P[F]$ here match exactly the factor states $I[F]$ for $F \in \mathcal{F}$. We construct D as the non-expanded initial decoupled state $I_0^{\mathcal{F}}$. Defining $g(I[F]) := P[F]$, we obviously get (a) – (c).

For the inductive case, say that U' results from U by adding event e. We denote $a := \phi(e)$. By IH, we have a mapping g from D to U satisfying (*). We show how to extend D and g to suitable D' and g' respectively.

We construct D' by, for every $s^{\mathcal{F}}$ where $\pi^C(s^{\mathcal{F}})$ extends $[\mathsf{pre}(e)]^C$, *extending $s^{\mathcal{F}}$ with a*, as follows. If a is a leaf action, then (i) we apply a to every factor state p in $s^{\mathcal{F}}$ where $p \models \mathsf{pre}(a)$. If a is a center action and $s^{\mathcal{F}} \models \mathsf{pre}(a)$, then we apply a to $s^{\mathcal{F}}$, resulting in a new successor $t^{\mathcal{F}}$. In the latter, (ii) we add the updated center state; (iii) for every leaf factor L affected by a, and for every $s^L \in S^L \cap S^{\mathcal{L}}(s^{\mathcal{F}})$ where $s^L \models \mathsf{pre}(a)[L]$, we add s^L updated with $\mathsf{eff}(a)[L]$; (iv) for every (leaf) factor L not affected by a, we add to $t^{\mathcal{F}}$ occurrences of actions $a^L \in \mathcal{A}^L \setminus \mathcal{A}^C$ reaching all of $S^L \cap S^{\mathcal{L}}(s^{\mathcal{F}})$. The latter is possible because, without prevail conditions, no such a^L has preconditions on the center.

Observe that this construction of D builds several decoupled states in a parallel manner, in difference to the actual construction of (pruned) $\Theta_\Pi^{\mathcal{F}}$ during search. However, the construction of D complies with the unfolding search order.

Regarding the construction of g': For (i) – (iii), let $\hat{p'}[F]$ be a new factor state occurrence added to D' by an occurrence $\hat{a}$ of a, and let $\hat{p}[F]$ be the factor state occurrence that $\hat{a}$ is applied to. By IH, $P[F] := g(\hat{p}[F])$ is a factor co-set and $P[F] \supseteq \mathsf{pre}(e)[F]$. Let $P'[F] := (P[F] \setminus \mathsf{pre}(e)[F]) \cup \mathsf{eff}(e)[F]$. Then $P'[F]$ is a co-set in U'. We set $g'(\hat{p'}[F]) := P'[F]$. For (iv), i. e., a factor state occurrence $\hat{p'}[L]$ of $p'[L] \in S^L \cap S^{\mathcal{L}}(s^{\mathcal{F}})$ added to D', we define $g'(\hat{p'}[L]){:=}g(\hat{p}[L])$, where $\hat{p}[L]$ is $p'[L]$'s occurrence in $s^{\mathcal{F}}$.

We next show that g' has the desired properties (*) on D' and U'. Obviously, (a) is given by construction.

To see that g' is surjective, note that any new factor coset $P'[F]$ in U' must result from a factor co-set $P[F]$ in U through $P'[F] := (P[F] \setminus \mathsf{pre}(e)[F]) \cup \mathsf{eff}(e)[F]$ where $\mathsf{eff}(e)[F] \neq \emptyset$ and thus $\mathsf{pre}(e)[F] \neq \emptyset$. Let $Q := P[F] \cup \mathsf{pre}(e)$. Then Q is a co-set in U as otherwise $P'[F]$ could not be a co-set in U'. By IH (b), there is at least one $s^{\mathcal{F}}$ in D where $\pi^C(s^{\mathcal{F}})$ extends $[Q]^C$. By IH (c), for every F there is $\hat{p}[F]$ in $s^{\mathcal{F}}$ where $g(\hat{p}[F]) \supseteq Q[F] = P[F]$, which implies with IH (a) that $g(\hat{p}[F]) = P[F]$.

As $Q \supseteq \mathsf{pre}(e)$, we have that $\pi^C(s^{\mathcal{F}})$ extends $[\mathsf{pre}(e)]^C$. Thus $s^{\mathcal{F}}$ has been extended with $a = \phi(e)$. If a is a leaf action, then, because there are no prevail conditions and thus no Petri net outputs of e on the center, F must be the respective leaf factor L. We have $p[L] \models \mathsf{pre}(a)$, so a was applied to $p[L]$ by (i), generating the outcome state $\phi(P'[L])$ which is mapped by g' to $P'[F]$ as desired. If a is a center action, then, because there are no prevail conditions and thus no Petri net outputs of e on factors not affected by a, F must be either (ii) the center or (iii) a leaf factor L affected by a. In both cases, a was applied to $p[F]$, generating the outcome state $\phi(P'[F])$ which is mapped by g' to $P'[F]$ as desired.

Let now Q' be any new co-set in U'. We must show that (b) and (c) hold for Q'. Observe first that Q' must result from $Q := (Q' \setminus \mathsf{eff}(e)) \cup \mathsf{pre}(e)$ in U, and that Q is a co-set in U.

Regarding (b): By IH (b), there is at least one $s^{\mathcal{F}}$ where $\pi^C(s^{\mathcal{F}})$ extends $[Q]^C$. If a is a leaf action, there is nothing to show as, then, $[Q]^C = [Q']^C$. Say that a is a center action. By construction, $s^{\mathcal{F}}$ has been extended with a, producing a new successor $t^{\mathcal{F}}$. Clearly, $\pi^C(t^{\mathcal{F}})$ extends $[Q']^C$.

Regarding (c): Let $t^{\mathcal{F}}$ in D', where $\pi^C(t^{\mathcal{F}})$ extends $[Q']^C$, be arbitrary. First, say that a is a leaf action. Then D contains $s^{\mathcal{F}}$ with $\pi^C(s^{\mathcal{F}}) = \pi^C(t^{\mathcal{F}})$, namely the same decoupled state but yet with less leaf states. Let F be arbitrary. By IH (c), there is $\hat{p}[F]$ in $s^{\mathcal{F}}$ where $g(\hat{p}[F]) \supseteq Q[F]$. Say that a affects L. If $F \neq L$, then, as there are no prevail conditions and thus no outputs of e on any factor other than L, $Q[F] = Q'[F]$ and we are done. Say that $F = L$. Then, as $Q \supseteq \mathsf{pre}(e)$, we have $p[L] \models \mathsf{pre}(a)$ so a was applied to $p[L]$ by (i). The outcome state $p'[L]$ is mapped by g' to a co-set $P'[L]$ in U', where $P'[L] \supseteq Q'[L]$ as needed.

Finally, say that a is a center action. Then $t^{\mathcal{F}}$ was generated by extending $s^{\mathcal{F}}$ in D with a. Let F be arbitrary. By IH (c), there is $\hat{p}[F]$ in $s^{\mathcal{F}}$ where $g(\hat{p}[F]) \supseteq Q[F]$. If a affects F, then similar to the above we have $p[F] \models \mathsf{pre}(a)[F]$, so a was applied to $p[F]$ by either (ii) or (iii), and the outcome state $p'[F]$ in $t^{\mathcal{F}}$ is mapped by g' to a co-set $P'[F] \supseteq Q'[F]$ in U' as needed. If a does not affect F, then as above $Q[F] = Q'[F]$. In that case, due to construction (iv), $t^{\mathcal{F}}$ contains a new occurrence of $p[F]$, mapped by g' to $g(\hat{p}[F])$ which concludes the argument.

For the second part of the proof, consider now the pruned versions of $\Theta_\Pi^{\mathcal{F}}$ and Unf_Π, built using compatible orders $\ll$. Assume that e is a non-cut-off event in Unf_Π. Consider the construction step where e is added, and denote D, D' and U, U' as above. Consider the decoupled states $s^{\mathcal{F}}$ extended with $a := \phi(e)$ by the above construction. With compatibility of $\ll$, there is at least one such $s^{\mathcal{F}}$ in D'. For every such $s^{\mathcal{F}}$, and for every factor F affected by a, there is a factor-state occurrence $\hat{p}[F]$ in $s^{\mathcal{F}}$ mapped to a factor co-set $P[F] := g(\hat{p}[F])$ where $P[F] \supseteq \mathsf{eff}(e)[F]$ and in particular $|P[F]| \geq |\mathsf{eff}(e)[F]|$.

Observe that, for any other event e', the factor-state occurrences $\hat{p'}[F]$ identified in the same manner must map to different factor co-sets $P'[F] \neq P[F]$, simply because every construction step of kinds (i) – (iii) maps to factor co-sets including newly generated conditions. Therefore, to prove the main claim it now suffices to show that at least one $s^{\mathcal{F}}$ as above is not pruned by hypercube pruning.

Clearly, $\phi(Mark([e])) \in [s^{\mathcal{F}}]$ for any such $s^{\mathcal{F}}$. Consider the first $s^{\mathcal{F}}$ generated in the construction of $\Theta_\Pi^{\mathcal{F}}$. As, by construction, D cannot represent states not represented by U, $\phi(Mark([e]))$ is not covered yet and $s^{\mathcal{F}}$ is not pruned. $\square$

Acknowledgments

Daniel Gnad was supported by the German Research Foundation (DFG), under grant HO 2169/6-1, "Star-Topology Decoupled State Space Search". Jörg Hoffmann's research group has received support by DFG grant 389792660 as part of TRR 248 (see perspicuous-computing.science).

References

Bäckström, C., and Nebel, B. 1995. Complexity results for SAS$^+$ planning. *Computational Intelligence* 11(4):625–655.

Baldan, P.; Bruni, A.; Corradini, A.; König, B.; Rodríguez, C.; and Schwoon, S. 2012. Efficient unfolding of contextual Petri nets. *Theoretical Computer Science* 449:2–22.

Bonet, B.; Haslum, P.; Hickmott, S. L.; and Thiébaux, S. 2008. Directed unfolding of petri nets. *Transactions on Petri Nets and Other Models of Concurrency* 1:172–198.

Bonet, B.; Haslum, P.; Khomenko, V.; Thiébaux, S.; and Vogler, W. 2014. Recent advances in unfolding technique. *Theoretical Computer Science* 551:84–101.

Brafman, R.; Domshlak, C.; Haslum, P.; and Zilberstein, S., eds. 2015. *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS'15)*. AAAI Press.

Esparza, J.; Römer, S.; and Vogler, W. 2002. An improvement of mcmillan's unfolding algorithm. *Formal Methods in System Design* 20(3):285–310.

Gnad, D., and Hoffmann, J. 2018. Star-topology decoupled state space search. *Artificial Intelligence* 257:24 – 60.

Godefroid, P., and Wolper, P. 1991. Using partial orders for the efficient verification of deadlock freedom and safety properties. In *Proceedings of the 3rd International Workshop on Computer Aided Verification (CAV'91)*, 332–342.

Helmert, M. 2006. The Fast Downward planning system. *Journal of Artificial Intelligence Research* 26:191–246.

Hickmott, S. L.; Rintanen, J.; Thiébaux, S.; and White, L. B. 2007. Planning via petri net unfolding. In Veloso, M., ed., *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, 1904–1911. Hyderabad, India: Morgan Kaufmann.

McMillan, K. L. 1992. Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. In von Bochmann, G., and Probst, D. K., eds., *Proceedings of the 4th International Workshop on Computer Aided Verification (CAV'92)*, Lecture Notes in Computer Science, 164–177. Springer.

Pommerening, F., and Helmert, M. 2015. A normal form for classical planning tasks. In Brafman et al. (2015), 188–192.

Valmari, A. 1989. Stubborn sets for reduced state space generation. In *Proceedings of the 10th International Conference on Applications and Theory of Petri Nets*, 491–515.

Wehrle, M., and Helmert, M. 2014. Efficient stubborn sets: Generalized algorithms and selection strategies. In Chien, S.; Do, M.; Fern, A.; and Ruml, W., eds., *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS'14)*. AAAI Press.

Wehrle, M.; Helmert, M.; Alkhazraji, Y.; and Mattmüller, R. 2013. The relative pruning power of strong stubborn sets and expansion core. In Borrajo, D.; Fratini, S.; Kambhampati, S.; and Oddi, A., eds., *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS'13)*. Rome, Italy: AAAI Press.