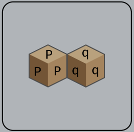
Abstraction Heuristics for Classical Planning Task with Conditional Effects

Martín Pozo
Jendrik Seipp

>>> Abstractions



L

R

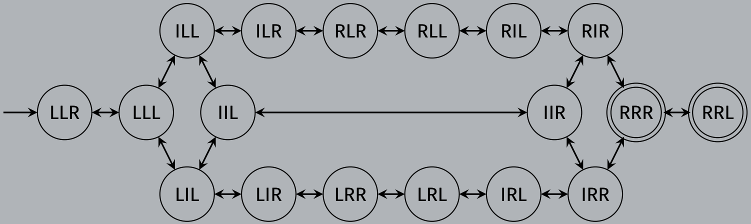
$V = \{p, q, t\}$

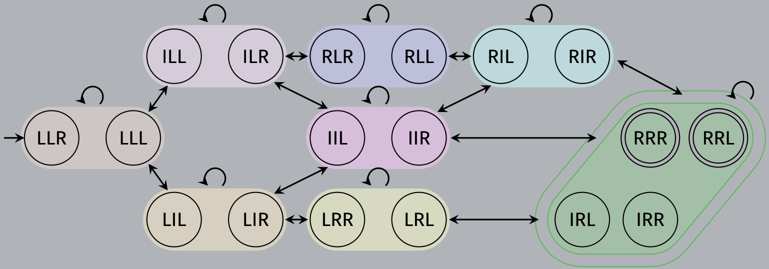
$D_t = \{L, R\}, D_p = D_q = \{L, R, I\}$

$I = \{p \mapsto L, q \mapsto L, t \mapsto R\}$

$G = \{p \mapsto R, q \mapsto R\}$

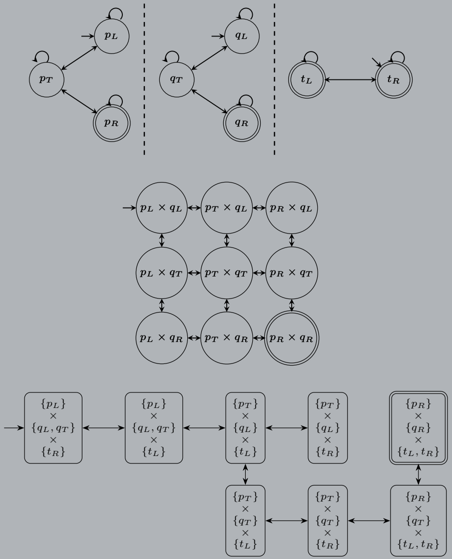
Operators: drive, pick, drop





>>> Abstractions

- Projections completely abstract one or more variables
  - The full domain of abstracted variables is true at the same time in all states
  - Atomic projections abstract all variables except one
  - Typically many projections are combined in a PDB
- Merge-and-shrink abstractions start from atomic projections and apply transformations on them
  - Merge two abstractions
  - Shrink an abstraction
  - Label reduction
  - Pruning
- In Cartesian Abstractions each state is mapped to a Cartesian set of states
  - Fine grained and efficient



>>> CEGAR (Counterexample-Guided Abstraction Refinement)

- Start from the trivial abstraction
  - A single state consistent with all concrete states

- Refine it in a loop until reaching a termination condition
  - Typically a memory or time limit

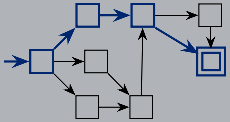
1. While not termination condition:

   1.1. Find an optimal abstract plan
   - If not found $\Rightarrow$ unsolvable task

   1.2. Execute it in the concrete state space until finding a flaw
   - If no flaws found $\Rightarrow$ solution found

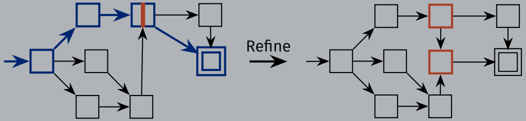
   1.3. Split the abstract state into 2 states

1.1.



1.2.

1.3.



>>> Conditional effects: why?

- Compact representation for complex tasks
- Each effect now contains a number of facts as conditions
  - $o = \langle pre(o), \mathit{eff}(o), cost(o) \rangle \in O$, where each $e \in \mathit{eff}(o) =$
    - $\langle cond s(e) \equiv$ partial state, $atom(e) \equiv$ atom$\rangle$
- Compiling them away $\rightarrow$ exponential growth

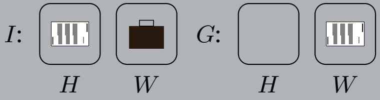
>>> Conditional effects: Briefcase (Pednault, 1988)

$V = \{v_B, v_D, v_I\}$

$D_{v_B} = D_{v_D} = \{H, W\}, D_{v_I} = \{\bot, \top\}$

$I = \{v_B \mapsto W, v_D \mapsto H, v_I \mapsto \bot\}$

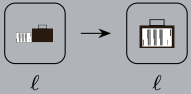
$G = \{v_D \mapsto W\}$



Operators:

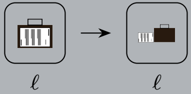
$$store(\ell) = \left\langle \begin{array}{c} \{v_B{=}\ell, v_D{\mapsto}\ell, v_I{\mapsto}\bot\}, \\ \{\langle\{\}, v_I{=}\top\rangle\}, \\ 1 \end{array} \right\rangle$$

$store(\ell)$:



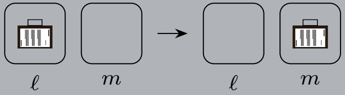
$$takeout(\ell) {\mapsto} \left\langle \begin{array}{c} \{v_B{\mapsto}\ell, v_D{\mapsto}\ell, v_I{\mapsto}\top\}, \\ \{\langle\{\}, v_I{\mapsto}\bot\rangle\}, \\ 1 \end{array} \right\rangle$$

$takeout(\ell)$:



$$move(\ell, m) = \left\langle \begin{array}{c} \{v_B{\mapsto}\ell\}, \\ \{\langle\{v_I{\mapsto}\top\}, v_D{\mapsto}m\rangle, \langle\{\}, v_B{\mapsto}m\rangle\}, \\ 1 \end{array} \right\rangle$$

$move(\ell, m)$:



>>> Projections and merge-and-shrink abstractions

- Non-induced abstractions (over-approximations)
  - Some abstract transitions have no correspondence in the concrete space
  - Making induced abstractions for conditional effects is too expensive

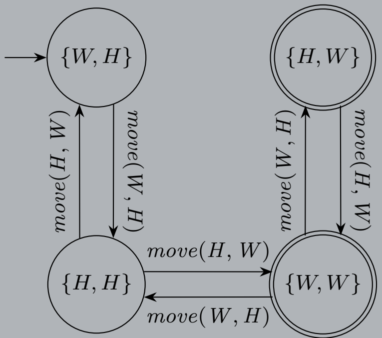
Projections

- Outgoing transitions algorithm:
  1. If preconditions are not satisfied in the projection, no transition
  2. Else $\{a \xrightarrow{o} b \mid b \in \bigtimes_{v \in P}$ all possible post values for $v\}$

Merge-and-shrink abstractions

- Initial atomic projections are computed as above

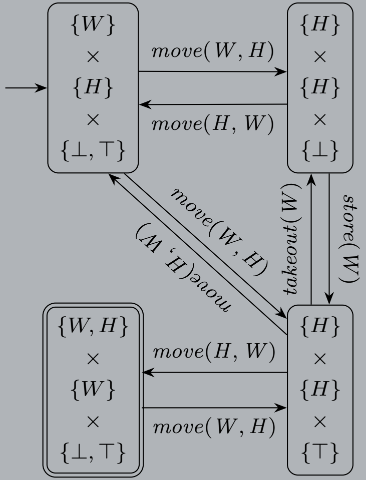
Projection with states $\{v_B, v_D\}$ (v_I abstracted)



>>> Cartesian abstractions: rewiring transitions

- Also non-induced abstractions

- Outgoing transitions algorithm for a child Cartesian state a after a split:
  1. If preconditions are not satisfied in a, no transition
  2. Else
     2.1. $post \leftarrow a \cap \mathcal{C}(pre(o))$
     2.2. For all effects with effect atom x **possibly** satisfied in a, $post[v] \leftarrow post[v] \cup \{x\}$
     2.3. For all effects with effect atom x **always** satisfied in a, $post[v] \leftarrow \{x\}$
     2.4. $\{a \xrightarrow{o} b \mid b \in S^\alpha, b \cap post \neq \varnothing\}$



>>> Cartesian abstractions: progression flaws

- Only deviation flaws $\langle s, c\rangle$ are affected, where $c = a \cap regr(b, o)$

>>> Cartesian abstractions: progression flaws

- But regression is **not Cartesian** with conditional effects!

>>> Cartesian abstractions: progression flaws

- Only deviation flaws $\langle s, c\rangle$ are affected, where $c = a \cap regr(b, o)$
- Cartesian over-approximation

$$regr(b,o)[v] = \begin{cases} \{pre(o)[v]\} & \text{if } v \in vars(pre(o)) \\ \mathcal{D}_v & \textbf{if } \langle C, v \mapsto x\rangle \in eff(o), x \in b[v] \\ b[v] \cup \{x\} & \textbf{if } \langle \{v \mapsto x, ...\}, w \mapsto y\rangle \in eff(o), y \in b[w] \\ b[v] & \text{otherwise} \end{cases} \quad (1)$$

- Multiple causes for a deviation may happen
  - Refined one by one in each iteration of the loop
  - As done for tasks without conditional effects for multiple non-satisfied preconditions

>>> Cartesian abstractions: regression flaws

- Search for flaws in regression from the goals of the plan
  - Regression applied as the above Cartesian over-approximation
  - Superset of the actual regression $\Rightarrow$ fewer flaws found
  - Mitigated by searching for a progression flaw as a fallback when no regression flaw is found

>>> Experiments

Domain	#Tasks	h^{LMC}	Sym	$h^{\text{Cart}}_{\text{fact}}$	$h^{\text{PDBs}}_{\text{fact}}$	$\mathcal{C}^{\text{fBS}}_{\text{B}}$
General Conds.	543	296	316	–	–	**341**
Briefcase	50	9	9	–	–	**17**
Caldera	20	10	10	–	–	**17**
CalderaSplit	20	8	**11**	–	–	9
Citycar	20	16	**18**	–	–	17
FSC Domains	57	**20**	**20**	–	–	19
GED Domains	26	15	**20**	–	–	**20**
Miconic	150	142	**150**	–	–	147
Nurikabe	20	12	11	–	–	**14**
Rubik's Cube	20	7	6	–	–	**10**
Settlers	20	8	9	–	–	**12**
Spider	20	11	8	–	–	**18**
TO Domains	120	38	**44**	–	–	41

Domain	#Tasks	h^{LMC}	Sym	$h^{\text{Cart}}_{\text{fact}}$	$h^{\text{PDBs}}_{\text{fact}}$	$\mathcal{C}^{\text{fBS}}_{\text{B}}$
CNOT Domains	992	631	688	–	–	**733**
CNOT	219	196	210	–	–	**214**
CNOT Hard	526	189	237	–	–	**273**
CNOT Map	247	**246**	241	–	–	**246**
Factored Tasks	428	135	207	182	**247**	196
Cavediving	17	**4**	**4**	**4**	**4**	**4**
Matrix Mult.	11	**7**	**7**	**7**	**7**	**7**
Burnt Pancakes	100	30	49	40	**53**	45
Pancakes	100	35	52	44	**59**	51
Rubik's Cube 2	100	37	50	47	**66**	51
Topspin	100	22	45	40	**58**	38
Total	1963	1062	1211	182	247	**1270**

>>> Conclusions

- Support of tasks with conditional effects
  - Projections
  - Merge-and-shrink abstractions
  - Cartesian abstractions
    - Progression flaws
    - Regression flaws
    - Sequence flaws
- Combining projections and Cartesian abstractions via online SCP solves more tasks than symbolic search
- Projections for factored tasks are still better suited for these tasks