AI Planning

19. Dominance Pruning

On States that are Worse than Other States

Álvaro Torralba, Cosmina Croitoru

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Winter Term 2018/2019

Thanks to Prof. Jörg Hoffmann for slide sources

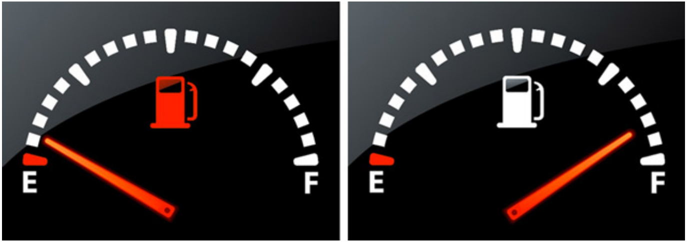
Agenda

Introduction

Reminder: → **Chapter 14**

State Pruning: Reduces search effort by cross-state comparisons. Prunes states whose exploration can be shown to be unnecessary.

When is a state unnecessary? → By comparing it to another.



A Non-Original Running Example

- Initial state I: $t(R), p_1(L), p_2(L), \mathit{fuel}(F3)$.
- Goal G: $p_1(R), p_2(R)$.
- Actions A: $move, l_L^i, l_R^i, u_L^i, u_R^i, refuel$.

Question!

Intuitively, what do we want to capture here? Which states are "better than" others?

Our Agenda for This Chapter

2. **Basics of Dominance Relations:** Formal definition of dominance based on simulation relations.
3. **Simulation Relations:** Formal definition of dominance based on simulation relations.
4. **Compositional Approach:** Basic technique to find dominance relations based on a compositional approach.
5. **NOOP Simulation:** More elaborate method to find a dominance relation for a given planning task.

Dominance Relation

Definition (Relation). *A (binary) relation R on set S is a subset of $S \times S$, i.e., a set of pairs of states. The interpretation of this subset is that it contains all the pairs for which the relation is true.*

Definition (Dominance Relation). *A dominance relation of a transition system Θ is a binary relation $\preceq$ on S such that $s \preceq t$ (t dominates s) iff t is at least as good as s, i.e., $h^*(s) \geq h^*(t)$.*

Properties:

→ Reflexive: $s \preceq s$ for all s.

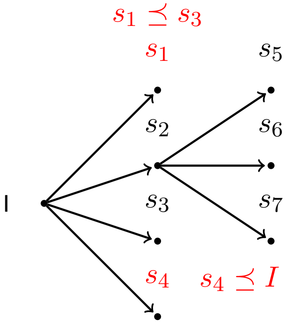
→ Transitive. $s \preceq t$ and $t \preceq u$ then $s \preceq u$.

→ Note that $h^*(s) \geq h^*(t)$ is talking about real cost, not heuristic. Heuristics are just estimations so $h(s) \geq h(t)$ does not guarantee anything.

Dominance Pruning

A dominance relation of a transition system Θ is a relation on pairs of states $\preceq: S \otimes S \rightarrow \{F, T\}$ such that $s \preceq t$ (t dominates s) iff t is at least as good as s, i.e., $h^*(s) \geq h^*(t)$.

Admissible pruning: If $g(t) \leq g(s)$ and $s \preceq t$ then s can be pruned.

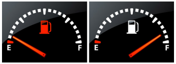


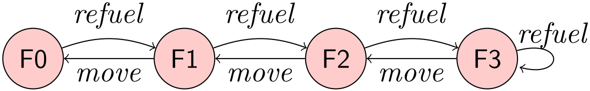
Challenges:

1. How to find good dominance relations? $\rightarrow$ **This Chapter**
2. How to efficiently check dominance?
   We do not go into detail here. In practice, you can use efficient data-structures to compare against all previously known states (see Remarks).

Simulation Relation [Milner (1971)]

Definition (Simulation Relation). *A binary relation* $\preceq \subseteq S \times S$ *is a simulation for* Θ *if, whenever* $s \preceq t$, *for every transition* $s \xrightarrow{l} s'$ *there exists* $t \xrightarrow{l} t'$ *s.t.* $s' \preceq t'$. *We call* $\preceq$ *goal-respecting for* Θ *if, whenever* $s \preceq t$, $s \in S_G$ *implies that* $t \in S_G$. *A simulation relation* $\preceq$ *is the coarsest iff for every other simulation relation* $\preceq'$, $\preceq' \subseteq \preceq$.





What is the coarsest simulation relation?:

Simulation Relation is a Dominance Relation

Theorem (Simulation Relations are Dominance Relations). *Let Θ be the state of a planning task. Then, if $\preceq$ is a goal-respecting simulation relation of Θ, $\preceq$ is a dominance relation.*

Proof. We need to show that $s \preceq t$ implies that $h^*(s) \geq h^*(t)$. Let $\pi = \{a_1, \ldots, a_k\}$ be an optimal plan for s. Proof by induction in the length of π.

Base case, $|\pi| = 0$. s is a goal-state, therefore t is also a goal because $\preceq$ is goal-respecting.

Inductive case, $\pi = \{a_1, \ldots, a_k\}$. Then, $s \xrightarrow{a_1} s'$. By simulation, we know that $t \xrightarrow{a_1} t'$ s.t. $s' \preceq t'$. Since the optimal plan for s' is necessarily shorter than π, we know that $h^*(s') \geq h^*(t')$. Thus, $h^*(s) \geq h^*(t)$.

In short, if $\preceq$ is a simulation relation, $s \preceq t$ implies that any plan for s is a valid plan for t.

Coarsest Goal-respecting Simulation

Theorem (Unique Coarsest Goal-respecting Simulation). *Let Θ be a LTS. Then, a unique coarsest goal-respecting simulation always exists and can be computed in time polynomial in the size of Θ.*

Proof Intuition:

$\rightarrow$ Intuition for existence: The identity relation is always a simulation relation.

$\rightarrow$ Intuition for uniqueness of the coarsest simulation: If $\preceq$ and $\preceq'$ are simulation relations then $\preceq \cup \preceq'$ is also a simulation relation.

$\rightarrow$ Intuition for polynomial time: The definition of simulation is co-inductive and monotonic: removing elements from $\preceq$ cannot cause others to be added. Details for an algorithm in the next section.

Compute a Simulation Relation

function ComputeSimulationRelation *(Θ)*

$\preceq \leftarrow \{(s, t)$ such that $t \in S_G \vee s \notin S_G\}$

while $\exists\ s \preceq t$ s.t., $(s \xrightarrow{l} s'$ and $\nexists t'$ s.t. $s' \preceq t'$ and $t \xrightarrow{l} t')$ **do**

$\preceq \leftarrow \preceq \setminus \langle s, t \rangle$

return $\preceq$

Theorem. ComputeSimulationRelation computes the coarsest goal-respecting simulation.

Proof. It always terminates because at each iteration it removes a pair $\langle s, t \rangle$ from $\preceq$.

The result is a simulation because that is the termination condition of the while loop.

The result is goal-respecting because $\preceq$ is initialized with a goal-respecting relation and no pair $\langle s, t \rangle$ is introduced afterwards.

The result is the coarsest goal-respecting simulation because we start assuming that every goal-respecting s, t belongs to the relation. If $\langle s, t \rangle$ is removed from $\preceq$ then it cannot possibly belong to any simulation relation, i.e., removing pairs cannot help for any state to simulate another.

Cost-Simulation

Simulation: Any plan for s is also a valid plan for t:

$$s \xrightarrow{a_1} s' \xrightarrow{a_2} \dots \xrightarrow{a_k} s_G$$
$$t \xrightarrow{a_1} t' \xrightarrow{a_2} \dots \xrightarrow{a_k} t_G$$

We do not care that the same plan applies, just about its cost.

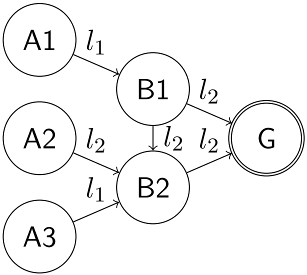
Definition (Cost-Simulation Relation). *A relation $\preceq \subseteq S \times S$ is a cost-simulation for Θ if, whenever $s \preceq t$, $s \in S_G$ implies that $t \in S_G$, and for every transition $s \xrightarrow{l} s'$ there exists $t \xrightarrow{l'} t'$ s.t. $s' \preceq t'$ and $c(l') \leq c(l)$.*

Cost-simulation For any plan for s there is one for t

Theorem A cost-simulation is always a dominance relation.

Proof The same proof as for simulation relation applies!

Questionaire



Question!

Which of the following hold in the coarsest simulation relation?

(A): $B_2 \preceq G$ (B): $B_1 \preceq B_2$

(C): $A_2 \preceq A_3$ (D): $A_3 \preceq A_1$

Compositional Approach

Problem: Computing a simulation relation is polynomial in the size of Θ but this is exponential in the size of our planning task!

Compositional approach:

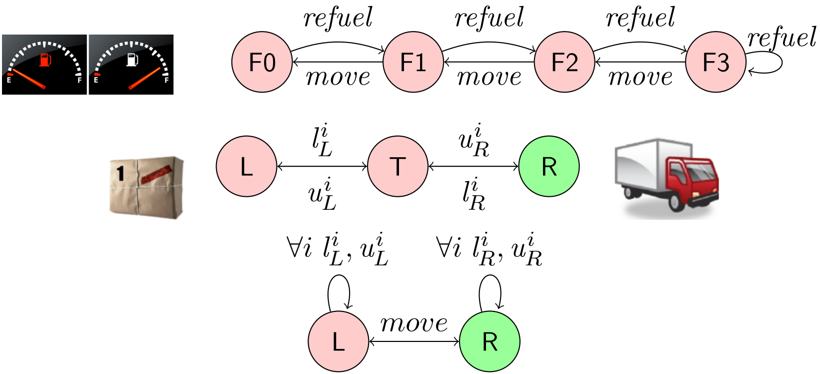
- Consider a partition of the problem: $\Theta_1, \dots, \Theta_k$.
- Compute a simulation relation in each partition: $\preceq_1, \dots, \preceq_k$
- A state dominates another iff it dominates in every aspect: $s \preceq t$ iff $s_i \preceq_i t_i$ for all i.

Definition (Partition of the problem). *Let Π be an FDR planning task and $\mathcal{T} = \{\Theta_1, \dots, \Theta_k\}$ a set of LTSs. $\mathcal{T}$ is a partition of Π iff $\Theta_1 \otimes \cdots \otimes \Theta_k = \Theta_\Pi$.*

For example, the atomic projections are a partition of the planning task. In general, the partition can be the projections of the planning task onto a partition of the variables **(cf. Chapters 12 and 13)**.

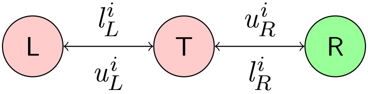
Computation of Compositional Approach

- Initial state I: $t(R), p_1(L), p_2(L), fuel(F3)$.
- Goal G: $p_1(R), p_2(R)$.
- Actions A: $move, l^i_L, l^i_R, u^i_L, u^i_R$.



(*) Transitions that don't affect a transition system induce a self-loop in

Computing Simulation Relations: Example Package



Question!

Compute the coarsest goal-respecting simulation relation on the package partition above. Do the following hold?

(A): $L \preceq T$

(B): $T \preceq L$

(C): $T \preceq R$

(D): $L \preceq R$

Combining the Partitions

- Initial state I: $t(R), p_1(L), p_2(L), fuel(F3)$.
- Goal G: $p_1(R), p_2(R)$.
- Actions A: $move, l^i_L, l^i_R, u^i_L, u^i_R$.

$F0 \preceq_f F1 \preceq_f F2 \preceq_f F3$

$\preceq_t, \preceq_{p1}, \preceq_{p2}$ = Identity

A state dominates another iff it dominates in every aspect:

$$s \preceq t \text{ iff } s_i \preceq_i t_i \text{ for all } i.$$

For example:

- $LTR2 \preceq LTR3$ because $L \preceq_{p1} L$, $T \preceq_{p2} T$, $R \preceq_t R$, and $2 \preceq_f 3$.
- $LLL3 \not\preceq LLR3$ because $L \not\preceq_t R$.

Compositional Approach Produces a Simulation

Theorem (Compositional Approach). *Let Π be an FDR planning task and $\mathcal{T} = \{\Theta_1, \dots, \Theta_k\}$ a partition of Π. Let $\{\preceq_1, \dots, \preceq_k\}$ be a set of relations s.t. $\preceq_i$ is a simulation of Θ_i. Define $\preceq$ as a relation on Θ s.t. $s \preceq t$ iff $s_i \preceq_i t_i$ for all Θ_i. Then, $\preceq$ is a simulation of Θ_Π.*

Proof Sketch (for reference). We show the claim for the case with two parts $\{\Theta_1, \Theta_2\}$ and induction takes care of the rest.
$\preceq_{12} = \{s_1 s_2, t_1 t_2$ s.t. $s_1 \preceq_1 t_1, s_2 \preceq_2 t_2\}$ is a simulation of $\Theta_{12} = \Theta_1 \otimes \Theta_2$ because for any $s_1 s_2 \preceq_{12} t_1 t_2$ and $s_1 s_2 \xrightarrow{l} s'_1 s'_2$, exists $t_1 t_2 \xrightarrow{l} t'_1 t'_2$ where $s'_1 s'_2 \preceq t'_1 t'_2$. Since a transition $s_1 s_2 \xrightarrow{l} s'_1 s'_2$ exists in Θ_{12}, by the definition of synchronized product **(cf. Chapter 13)**, $s_1 \xrightarrow{l} s'_1$ exists in Θ_1 and $s_2 \xrightarrow{l} s'_2$ exists in Θ_2.
By the definition of $\preceq_{12}$, $s_1 s_2 \preceq_{12} t_1 t_2$ implies that $s_1 \preceq_1 t_1$ and $s_2 \preceq_2 t_2$. Since $\preceq_1$ is a simulation, $s_1 \preceq_1 t_1$ and $s_1 \xrightarrow{l} s'_1$ then exists a transition $t_1 \xrightarrow{l} t'_1$ with $s'_1 \preceq_1 t'_1$ (and $t_2 \xrightarrow{l} t'_2$ for similar reasons).

Then, because of the definition of synchronized product, since $t_1 \xrightarrow{l} t'_1$ in Θ_1 and $t_2 \xrightarrow{l} t'_2$ in Θ_2 use the same label, exists $t_1 t_2 \xrightarrow{l} t'_1 t'_2$ in Θ_{12}. $s'_1 s'_2 \preceq_{12} t'_1 t'_2$ since $s'_1 \preceq_1 t'_1$ and $s'_2 \preceq_2 t'_2$.

$\rightarrow$Note that a cost-simulation in every partition does not produce a cost-simulation of the entire state space because $t_1 \xrightarrow{l} t'_1$ and $t_2 \xrightarrow{t}{}'_2$ must use the same label.

Recap

We want to compute a dominance relation for our planning task, Π.

We followed a compositional approach:

- Consider a partition of the problem: $\Theta_1, \dots, \Theta_k$.
- Compute a simulation relation in each partition: $\preceq_1, \dots, \preceq_k$
- A state dominates another iff it dominates in every aspect: $s \preceq t$ iff $s_i \preceq_i t_i$ for all i.
- $\preceq$ is a simulation relation of Θ

Problem: Simulation relations are too restrictive:

$s \preceq t$ implies that any plan for s is a valid plan for t.

All we need for dominance is $h^*(s) \geq h^*(t)$ but the plans for s and t could be completely different!

Using Label Equivalence

Cost-simulation For any plan for s there is

Definition (Label Equivalence). *Labels l and l' are equivalent in Θ iff $c(l) = c(l')$ and they have the same transitions:*

$$s \xrightarrow{l} s' \in \Theta \Leftrightarrow s \xrightarrow{l'} s' \in \Theta$$

$\rightarrow$ When computing simulation for Θ_i, labels that are equivalent in all other transition systems are interchangeable. The result is still a cost-simulation.

$\rightarrow$ In our simplified example, we used "refuel" as a label for different actions ($refuel_{i,j}$ for $(i, j) \in \{(0, 1), (1, 2), (2, 3), (3, 3)\}$). All those labels are equivalent in all variables except fuel.

Cost-Simulation with NOOP

Cost-simulation For any plan for s there is

Definition (NOOP). *Let Π be an FDR planning task. We define Π_{noop} as Π plus a NOOP action, of cost 0, empty preconditions and effects.*

→The *NOOP* action induces a 0-cost self-loop transition in every state of the state space: $s \xrightarrow{noop} s$, but preserves $h^*(s)$ for all s.

Cost-simulation with NOOP: For any plan for s there is one for t that

Theorem A cost-simulation on Θ_{noop} is a dominance relation for Θ.

NOOP Dominance

We need to capture when a label is important in the rest of the problem:

- Refuel is only important for the fuel variable.
- Loading/Unloading a package is only important for the package variable.

Definition (NOOP Dominance). *Label l is dominated by $noop$ in Θ given $\preceq$ if for every $s \xrightarrow{l} s' \in \Theta$ $s' \preceq s$.*

In words, if NOOP dominates label l in a partition, it means that l is not needed in any path in Θ_j. Not doing anything is as good as applying the action!

NOOP Simulation

Definition (NOOP Simulation).
A set $\mathcal{R} = \{\preceq_1, \dots, \preceq_k\}$ of binary relations $\preceq_i \subseteq S_i \times S_i$ is a NOOP simulation for $\{\Theta^1, \dots, \Theta^k\}$ if, whenever $s \preceq_i t$:

- $s \in S_i^G$ *implies that* $t \in S_i^G$, *and*
- *For every* $s \xrightarrow{l} s'$ *in* Θ^i,
  there exists $t \xrightarrow{l'} t'$ *in* Θ^i *s.t.:*
  1. $s' \preceq_i t'$,
  2. *for all* $j \neq i$, l' *is equivalent to* l *in* Θ^j

  or:
  1. $s' \preceq_i t$, *and*
  2. *for all* $j \neq i$, $NOOP$ *dominates* l *in* Θ^j

$\rightarrow$ NOOP simulation extends the definition of goal-respecting simulation by considering at the same time the relations on all partitions.

NOOP Simulation: Theoretical Results

Theorem *A coarsest label-dominance simulation always exists and can be computed in polynomial time.*

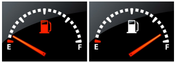
function ComputeNOOPSimulation $(\Theta_1, \dots, \Theta_k)$
For all $i \in [1, \dots, k]$, set $\preceq_i := \{(s,t) \mid s,t \in S_i, s \notin S_G^i$ or $t \in S_G^i\}$
while exists (i,s,t) s.t. not **Ok**(i,s,t) **do**
Select one such triple (i,s,t)
Set $\preceq_i := \preceq_i \setminus \{(s,t)\}$
return $\mathcal{R} := \{\preceq_1, \dots, \preceq_k\}$

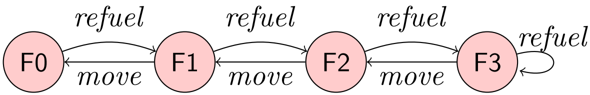
Theorem *Combination of* $\{\preceq_1, \dots, \preceq_k\}$ *is a cost-simulation for the planning task* $\Theta_1 \otimes \cdots \otimes \Theta_k$

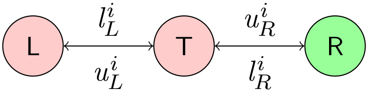
Proof intuition Like the proof for the composition of simulation relations but using different labels. NOOP dominance ensures that the transition exists in the final transition system.

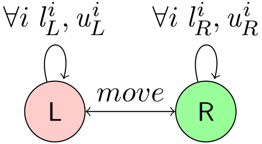
Computation of NOOP Simulation

- Initial state I: $t(R), p_1(L), p_2(L), fuel(F3)$.
- Goal G: $p_1(R), p_2(R)$.
- Actions A: $move, l^i_L, l^i_R, u^i_L, u^i_R, refuel$.









NOOP Simulation Algorithm in: (Blackboard)

Summary

- A dominance relation of a transition system Θ is a relation on pairs of states $\preceq: S \otimes S \rightarrow \{F, T\}$ such that $s \preceq t$ (t dominates s) iff t is at least as good as s, i.e., $h^*(s) \geq h^*(t)$.
- A binary relation $\preceq \subseteq S \times S$ is a goal-respecting simulation for Θ if, whenever $s \preceq t$, for every transition $s \xrightarrow{l} s'$ there exists $t \xrightarrow{l} t'$ s.t. $s' \preceq t'$ and $s \in S_G$ implies that $t \in S_G$.
- Compositional approach:
  - Consider a partition of the problem: $\Theta_1, \ldots, \Theta_k$.
  - Compute a simulation relation in each partition: $\preceq_1, \ldots, \preceq_k$
  - A state dominates another iff it dominates in every aspect: $s \preceq t$ iff $s_i \preceq_i t_i$ for all i.
- NOOP Simulation: Uses label dominance and computes $\preceq_i$ simultaneously in all partitions in order to find coarser relations.

Remarks

The simulation relation algorithm that we see in the lecture is just a simple algorithm for illustration purposes. In the model-checking literature there are more elaborate algorithms with lower computational complexity [Henzinger *et al.* (1995); Gentilini *et al.* (2003); Cécé (2013)].

There are multiple ways of implementing the check of whether given a state s there exists t in open or closed such that $s \preceq t$. A way to do that is to use efficient data structures to represent the sets of dominated states with a given g. Binary Decision Diagrams (BDDs) [Bryant (1986)] are an efficient data structure for these purposes.

→For the programming exercises, one should just compare each state against its parent, and prune any children dominated by their parent.

Remarks: Generalized Version

Definition (Label Dominance). *Label l' dominates l in Θ given $\preceq$ if for every $s \xrightarrow{l} s' \in \Theta$ there exists $s \xrightarrow{l'} t'$ s.t. $s' \preceq t'$.*

In words, if label l' dominates label l in a partition, it means that anything we can do with l, we can do with l'.

Definition (Label-Dominance Simulation).
A set $\mathcal{R} = \{\preceq_1, \dots, \preceq_k\}$ of binary relations $\preceq_i \subseteq S_i \times S_i$ is a label-dominance simulation for $\{\Theta^1, \dots, \Theta^k\}$ if, whenever $s \preceq_i t$:

- *$s \in S_i^G$ implies that $t \in S_i^G$*
- *For every $s \xrightarrow{l} s'$ in Θ^i, there exists $t \xrightarrow{l'} t'$ in Θ^i s.t.:*
  1. *$s' \preceq_i t'$,*
  2. *$c(l') \leq c(l)$, and*
  3. *for all $j \neq i$, l' dominates l in Θ^j given $\preceq_j$*

References I

Randal E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, 35(8):677–691, 1986.

Gérard Cécé. Three simulation algorithms for labelled transition systems. *CoRR*, abs/1301.1638, 2013.

Raffaella Gentilini, Carla Piazza, and Alberto Policriti. From bisimulation to simulation: Coarsest partition problems. *Journal of Automated Reasoning*, 31(1):73–103, 2003.

Monika Rauch Henzinger, Thomas A. Henzinger, and Peter W. Kopke. Computing simulations on finite and infinite graphs. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, pages 453–462. IEEE Computer Society, 1995.

Robin Milner. An algebraic definition of simulation between programs. In *Proceedings of the 2nd International Joint Conference on Artificial Intelligence (IJCAI'71)*, pages 481–489, London, UK, September 1971. William Kaufmann.