Planning and Optimization

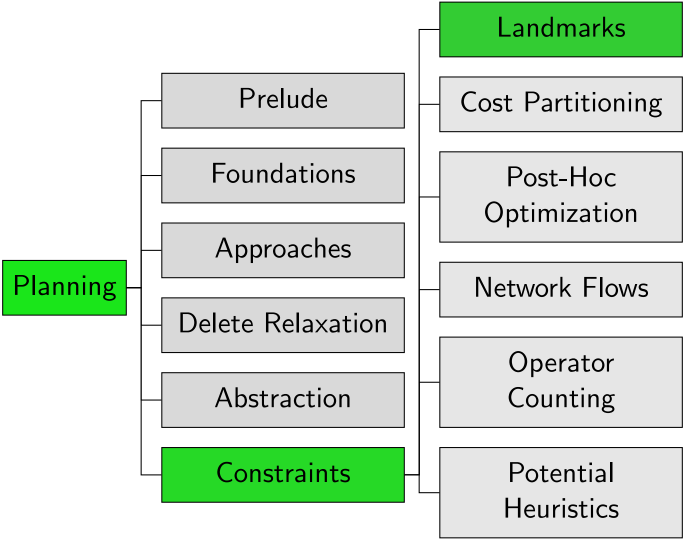
F2. Landmarks: RTG Landmarks

Malte Helmert and Gabriele Röger

Universität Basel

November 26, 2025

Content of the Course



Landmarks

Landmarks

Basic Idea: Something that must happen in every solution

For example

- some operator must be applied (action landmark)
- some atomic proposition must hold (fact landmark)
- some formula must be true (formula landmark)

$\rightarrow$ Derive heuristic estimate from this kind of information.

Landmarks

Basic Idea: Something that must happen in every solution

For example

- some operator must be applied (action landmark)
- some atomic proposition must hold (fact landmark)
- some formula must be true (formula landmark)

→ Derive heuristic estimate from this kind of information.

We mostly consider fact and disjunctive action landmarks.

Reminder: Terminology

Consider sequence of transitions $s^0 \xrightarrow{\ell_1} s^1, \dots, s^{n-1} \xrightarrow{\ell_n} s^n$ such that $s^0 = s$ and $s^n = s'$.

- $s^0, \dots, s^n$ is called (state) path from s to s'
- $\ell_1, \dots, \ell_n$ is called (label) path from s to s'

Disjunctive Action Landmarks

Definition (Disjunctive Action Landmark)

Let s be a state of a propositional or FDR planning task $\Pi = \langle V, I, O, \gamma \rangle$.

A disjunctive action landmark for s is a set of operators $L \subseteq O$ such that every label path from s to a goal state contains an operator from L.
The cost of landmark L is $cost(L) = \min_{o \in L} cost(o)$.

If we talk about landmarks for the initial state, we omit "for I".

Fact and Formula Landmarks

Definition (Formula and Fact Landmark)

Let s be a state of a propositional or FDR planning task $\Pi = \langle V, I, O, \gamma \rangle$.

A formula landmark for s is a formula λ over V such that every state path from s to a goal state contains a state s' with $s' \models \lambda$.

If λ is an atomic proposition then λ is a fact landmark.

If we talk about landmarks for the initial state, we omit "for I".

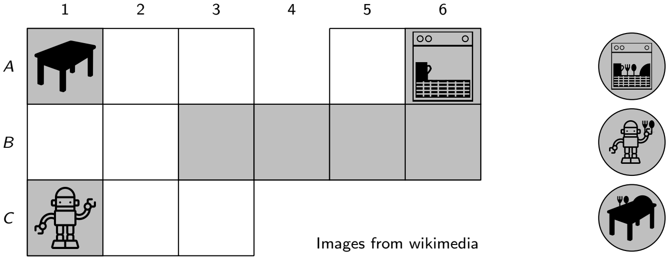
Landmarks: Example

Example

Consider a FDR planning task $\langle V, I, O, \gamma \rangle$ with

- $V = \{\mathit{robot\text{-}at}, \mathit{dishes\text{-}at}\}$ with
  - $\mathrm{dom}(\mathit{robot\text{-}at}) = \{\mathrm{A1}, \dots, \mathrm{C3}, \mathrm{B4}, \mathrm{A5}, \dots, \mathrm{B6}\}$
  - $\mathrm{dom}(\mathit{dishes\text{-}at}) = \{\mathrm{Table}, \mathrm{Robot}, \mathrm{Dishwasher}\}$
- $I = \{\mathit{robot\text{-}at} \mapsto \mathrm{C1}, \mathit{dishes\text{-}at} \mapsto \mathrm{Table}\}$
- operators
  - move-x-y to move from cell x to adjacent cell y
  - pickup dishes, and
  - load dishes into the dishwasher.
- $\gamma = (\mathit{robot\text{-}at} = B6) \land (\mathit{dishes\text{-}at} = \mathrm{Dishwasher})$

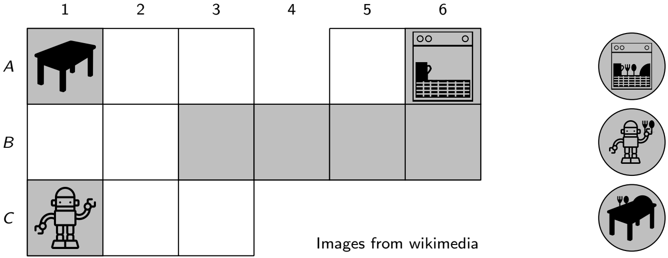
Fact and Formula Landmarks: Example



Each fact in gray is a fact landmark:

- `robot-at` $= x$ for $x \in \{\mathrm{A1, A6, B3, B4, B5, B6, C1}\}$
- `dishes-at` $= x$ for $x \in \{\mathrm{Dishwasher, Robot, Table}\}$

Fact and Formula Landmarks: Example



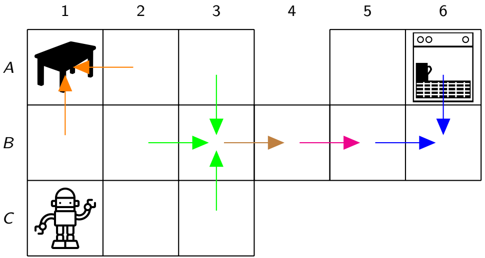
Each fact in gray is a fact landmark:

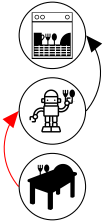
- `robot-at` $= x$ for $x \in \{\text{A1}, \text{A6}, \text{B3}, \text{B4}, \text{B5}, \text{B6}, \text{C1}\}$
- `dishes-at` $= x$ for $x \in \{\text{Dishwasher}, \text{Robot}, \text{Table}\}$

Formula landmarks:

- `dishes-at` $=$ Robot $\wedge$ `robot-at` $=$ B4
- `robot-at` $=$ B1 $\vee$ `robot-at` $=$ A2

Disjunctive Action Landmarks: Example





Actions of same color form disjunctive action landmark:

- {pickup}
- {load}
- {move-B3-B4}
- {move-B4-B5}
- {move-A6-B6, move-B5-B6}
- {move-A3-B3, move-B2-B3, move-C3-B3}
- {move-B1-A1, move-A2-A1}
- ...

Remarks

- Not every landmark is informative. Some examples:
  - The set of all operators is a disjunctive action landmark unless the initial state is already a goal state.
  - Every variable that is initially true is a fact landmark.
  - The goal formula is a formula landmark.
- Every fact landmark v that is initially false induces a disjunctive action landmark consisting of all operators that possibly make v true.

Complexity: Disjunctive Action Landmarks

Theorem

Deciding whether a given operator set is a disjunctive action landmark is as hard as the plan existence problem.

Proof.

Given a propositional planning task $\Pi = \langle V, I, O, \gamma \rangle$, create a new planning task Π' with goal $g \notin V$ as $\Pi' = \langle V \cup \{g\}, I \cup \{g \mapsto \mathbf{F}\}, O \cup \{o_\gamma, o_\top\}, g \rangle$, where

$$o_\gamma = \langle \gamma, g, 0 \rangle, \text{ and}$$
$$o_\top = \langle \top, g, 0 \rangle.$$

If $\gamma = \top$ then Π is trivially solvable. Otherwise Π is solvable iff $\{o_\top\}$ is not a disjunctive action landmark of Π'. $\square$

Complexity: Fact Landmarks

Theorem

Deciding whether a given atomic proposition is a fact landmark is as hard as the plan existence problem.

Proof.

Given a propositional planning task $\Pi = \langle V, I, O, \gamma \rangle$,
let $p, g \notin V$ be new atomic propositions and create a new planning task $\Pi' = \langle V \cup \{p, g\}, I \cup \{p \mapsto \mathbf{F}, g \mapsto \mathbf{F}\}, O \cup \{o, o'\}, g \rangle$, where

$$o = \langle \gamma, g, 0 \rangle, \text{ and}$$
$$o' = \langle \top, g \wedge p, 0 \rangle.$$

Then p is a fact landmark of Π' iff Π is not solvable. □

Complexity: Discussion

- Does this mean that the idea of exploiting landmarks is fruitless?

Complexity: Discussion

- Does this mean that the idea of exploiting landmarks is fruitless?– No!

Complexity: Discussion

- Does this mean that the idea of exploiting landmarks is fruitless?– No!
- We do not need to know all landmarks, so we can use incomplete methods to identify landmarks.
    - The way we generate the landmarks guarantees that they are indeed landmarks.
    - Efficient landmark generation methods do not guarantee to generate all possible landmarks.

Computing Landmarks

How can we come up with landmarks?

Most landmarks are derived from the relaxed task graph:

- RHW landmarks: Richter, Helmert & Westphal. Landmarks Revisited. (AAAI 2008)
- LM-Cut: Helmert & Domshlak. Landmarks, Critical Paths and Abstractions: What's the Difference Anyway? (ICAPS 2009)
- h^m landmarks: Keyder, Richter & Helmert: Sound and Complete Landmarks for And/Or Graphs (ECAI 2010)

Today we will discuss the special case of h^m landmarks for $m = 1$, restricted to STRIPS planning tasks.

Set Representation

Set Representation of STRIPS Planning Tasks

In this (and the following) sections, we only consider STRIPS. For a more convenient notation, we will use a set representation of STRIPS planning task...

Three differences:

- Represent conjunctions of variables as sets of variables.
- Use two sets to represent add and delete effects of operators separately.
- Represent states as sets of the true variables.

STRIPS Operators in Set Representation

- Every STRIPS operator is of the form

$$\langle v_1 \land \dots \land v_p, \quad a_1 \land \dots \land a_q \land \neg d_1 \land \dots \land \neg d_r, c\rangle$$

  where v_i, a_j, d_k are state variables and c is the cost.
- The same operator o in set representation is $\langle pre(o), add(o), del(o), cost(o)\rangle$, where
  - $pre(o) = \{v_1, \dots, v_p\}$ are the preconditions,
  - $add(o) = \{a_1, \dots, a_q\}$ are the add effects,
  - $del(o) = \{d_1, \dots, d_r\}$ are the delete effects, and
  - $cost(o) = c$ is the operator cost.
- Since STRIPS operators must be conflict-free, $add(o) \cap del(o) = \emptyset$

STRIPS Planning Tasks in Set Representation

A STRIPS planning task in set representation is given as a tuple $\langle V, I, O, G \rangle$, where

- V is a finite set of state variables,
- $I \subseteq V$ is the initial state,
- O is a finite set of STRIPS operators in set representation,
- $G \subseteq V$ is the goal.

STRIPS Planning Tasks in Set Representation

A STRIPS planning task in set representation is given as a tuple $\langle V, I, O, G \rangle$, where

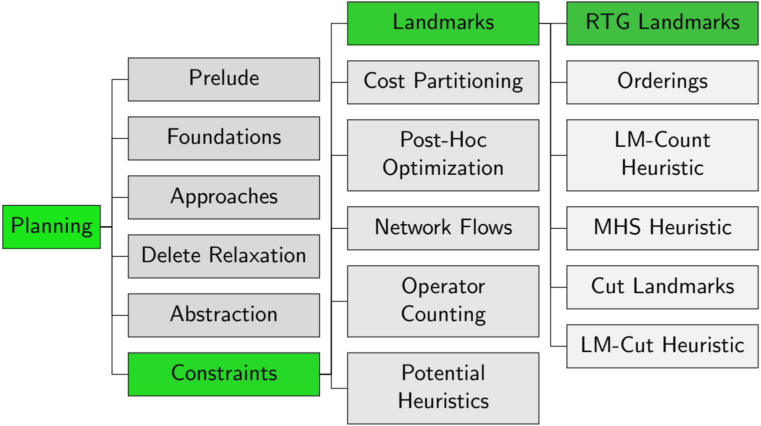
- V is a finite set of state variables,
- $I \subseteq V$ is the initial state,
- O is a finite set of STRIPS operators in set representation,
- $G \subseteq V$ is the goal.

The corresponding planning task in the previous notation is $\langle V, I', O', \gamma \rangle$, where

- $I'(v) = \mathbf{T}$ iff $v \in I$,
- $O' = \{\langle \bigwedge_{v \in pre(o)} v, \bigwedge_{v \in add(o)} v \wedge \bigwedge_{v \in del(o)} \neg v, cost(o) \rangle \mid o \in O\}$,
- $\gamma = \bigwedge_{v \in G} v$.

Landmarks from RTGs

Content of the Course



Incidental Landmarks: Example

Example (Incidental Landmarks)

Consider a STRIPS planning task $\langle V, I, \{o_1, o_2\}, G\rangle$ with

$$\begin{aligned} V &= \{a, b, c, d, e, f\}, \\ I &= \{a, b, e\}, \\ o_1 &= \langle\{a\}, \{c, d, e\}, \{b\}\rangle, \\ o_2 &= \langle\{d, e\}, \{f\}, \{a\}\rangle, \text{ and} \\ G &= \{e, f\}. \end{aligned}$$

Single solution: $\langle o_1, o_2\rangle$

- All variables are fact landmarks.
- Variable b is initially true but irrelevant for the plan.
- Variable c gets true as "side effect" of o_1 but it is not necessary for the goal or to make an operator applicable.

Causal Landmarks (1)

Definition (Causal Formula Landmark)

Let $\Pi = \langle V, I, O, \gamma \rangle$ be a propositional or FDR planning task.

A formula λ over V is a causal formula landmark for I if $\gamma \models \lambda$ or if for all plans $\pi = \langle o_1, \dots, o_n \rangle$ there is an o_i with $pre(o_i) \models \lambda$.

Causal Landmarks (2)

Special case: Fact Landmark for STRIPS task

Definition (Causal Fact Landmark)

Let $\Pi = \langle V, I, O, G\rangle$ be a STRIPS planning task (in set representation).

A variable $v \in V$ is a causal fact landmark for I

- if $v \in G$ or
- if for all plans $\pi = \langle o_1, \dots, o_n\rangle$ there is an o_i with $v \in pre(o_i)$.

Causal Landmarks: Example

Example (Causal Landmarks)

Consider a STRIPS planning task $\langle V, I, \{o_1, o_2\}, G\rangle$ with

$$\begin{aligned} V &= \{a, b, c, d, e, f\}, \\ I &= \{a, b, e\}, \\ o_1 &= \langle\{a\}, \{c, d, e\}, \{b\}\rangle, \\ o_2 &= \langle\{d, e\}, \{f\}, \{a\}\rangle, \text{ and} \\ G &= \{e, f\}. \end{aligned}$$

Single solution: $\langle o_1, o_2\rangle$

- All variables are fact landmarks for the initial state.
- Only a, d, e and f are causal landmarks.

What We Are Doing Next

- Causal landmarks are the desirable landmarks.
- We can use a simplified version of RTGs for STRIPS to compute causal landmarks for STRIPS planning tasks.
- We will define landmarks of AND/OR graphs, ...
- and show how they can be computed.
- Afterwards we establish that these are landmarks of the planning task.

Simplified Relaxed Task Graph

Definition

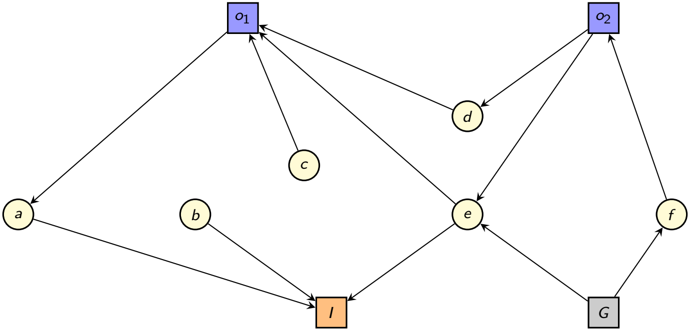
For a STRIPS planning task $\Pi = \langle V, I, O, G \rangle$ (in set representation), the simplified relaxed task graph $sRTG(\Pi^+)$ is the AND/OR graph $\langle N_{\text{and}} \cup N_{\text{or}}, A, type \rangle$ with

- $N_{\text{and}} = \{n_o \mid o \in O\} \cup \{v_I, v_G\}$
  with $type(n) = \wedge$ for all $n \in N_{\text{and}}$,
- $N_{\text{or}} = \{n_v \mid v \in V\}$
  with $type(n) = \vee$ for all $n \in N_{\text{or}}$, and
- $A = \{\langle n_a, n_o \rangle \mid o \in O, a \in add(o)\} \cup$
  $\{\langle n_o, n_p \rangle \mid o \in O, p \in pre(o)\} \cup$
  $\{\langle n_v, n_I \rangle \mid v \in I\} \cup$
  $\{\langle n_G, n_v \rangle \mid v \in G\}$

Like RTG but without extra nodes to support arbitrary conditions.

Simplified RTG: Example

The simplified RTG for our example task is:



Justification

Definition (Justification)

Let $G = \langle N, A, type \rangle$ be an AND/OR graph.

A subgraph $J = \langle N^J, A^J, type^J \rangle$ with $N^J \subseteq N$ and $A^J \subseteq A$ and $type^J = type|_{N^J}$ justifies $n_\star \in N$ iff

- $n_\star \in N^J$,
- $\forall n \in N^J$ with $type(n) = \wedge$:
  $\forall \langle n, n' \rangle \in A : n' \in N^J$ and $\langle n, n' \rangle \in A^J$
- $\forall n \in N^J$ with $type(n) = \vee$:
  $\exists \langle n, n' \rangle \in A : n' \in N^J$ and $\langle n, n' \rangle \in A^J$, and
- J is acyclic.

"Proves" that $n_\star$ is forced true.

Landmarks in AND/OR Graphs

Definition (Landmarks in AND/OR Graphs)

Let $G = \langle N, A, type \rangle$ be an AND/OR graph.

A node $n \in N$ is a landmark for reaching $n_\star \in N$
if $n \in V^J$ for all justifications J for $n_\star$.

But: exponential number of possible justifications

Characterizing Equation System

Theorem

Let $G = \langle N, A, type\rangle$ be an AND/OR graph. Consider the following system of equations:

$$LM(n) = \{n\} \cup \bigcap_{\langle n,n'\rangle \in A} LM(n') \quad type(n) = \vee$$

$$LM(n) = \{n\} \cup \bigcup_{\langle n,n'\rangle \in A} LM(n') \quad type(n) = \wedge$$

The equation system has a unique maximal solution (maximal with regard to set inclusion), and for this solution it holds that

$n' \in LM(n)$ iff n' is a landmark for reaching n in G.

Computation of Maximal Solution

Theorem

Let $G = \langle N, A, type \rangle$ be an AND/OR graph. Consider the following system of equations:

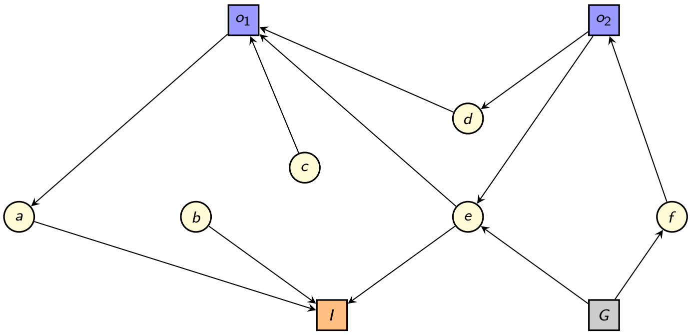
$$LM(n) = \{n\} \cup \bigcap_{\langle n,n' \rangle \in A} LM(n') \quad type(n) = \vee$$

$$LM(n) = \{n\} \cup \bigcup_{\langle n,n' \rangle \in A} LM(n') \quad type(n) = \wedge$$

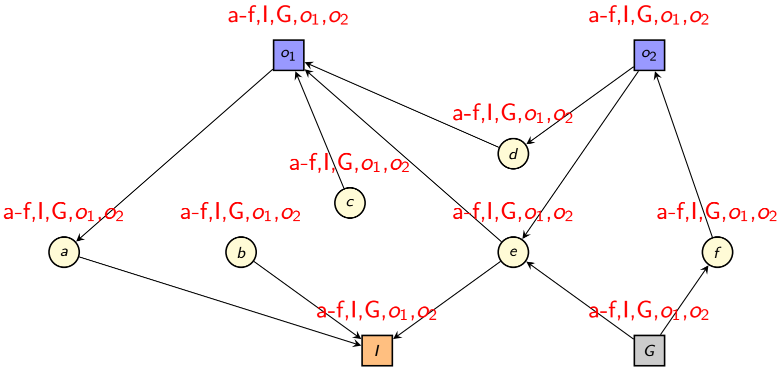
The equation system has a unique maximal solution (maximal with regard to set inclusion).

Computation: Initialize landmark sets as $LM(n) = N$ and apply equations as update rules until fixpoint.

Computation: Example

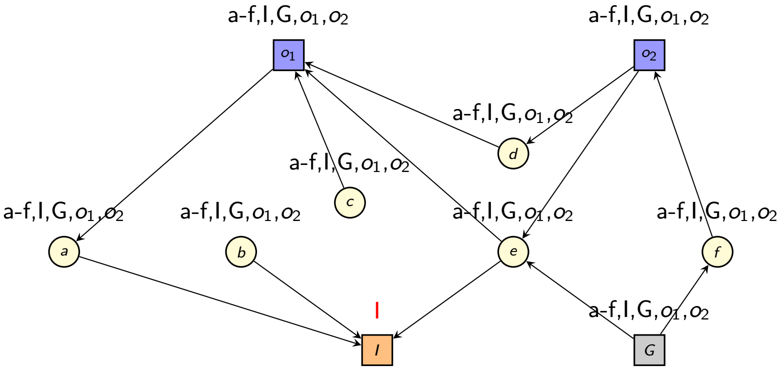


Computation: Example



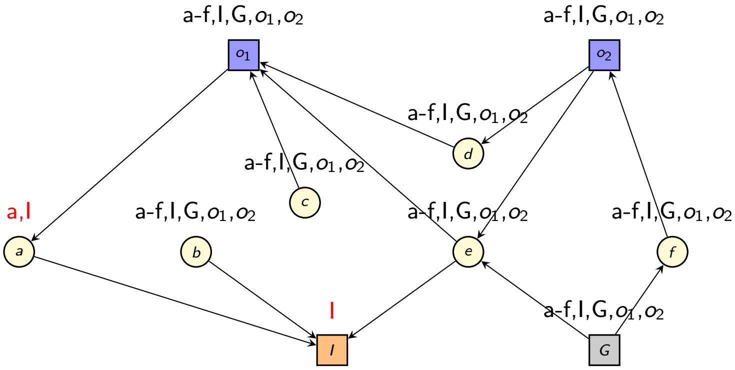
Initialize with all nodes

Computation: Example



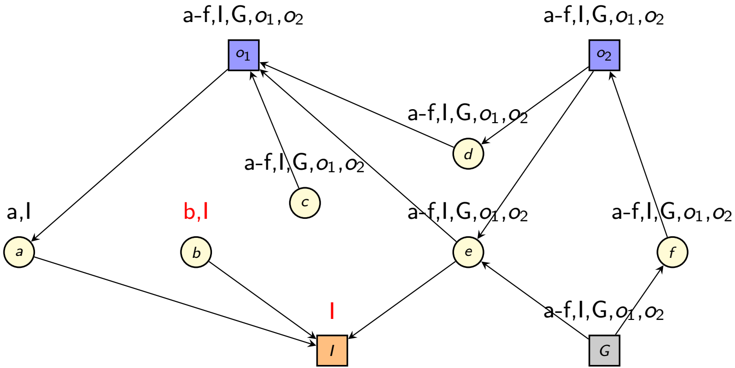
$$LM(I) = \{I\}$$

Computation: Example



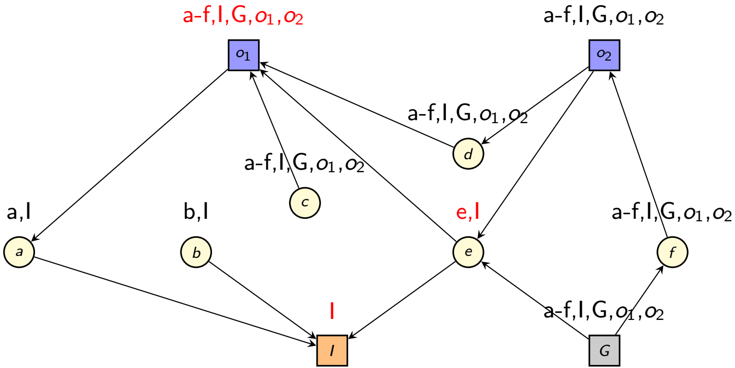
$$LM(a) = \{a\} \cup LM(I)$$

Computation: Example



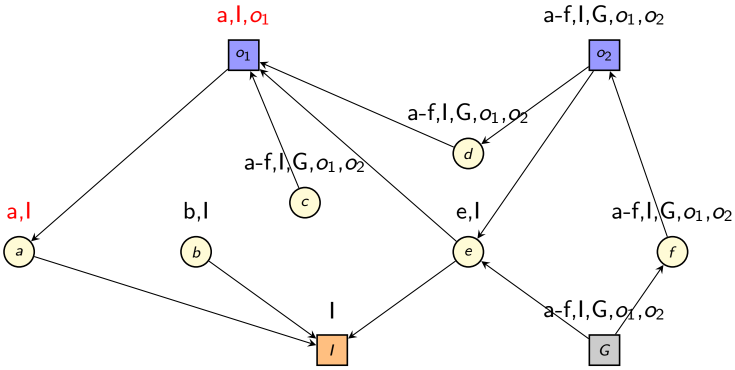
$$LM(b) = \{b\} \cup LM(I)$$

Computation: Example



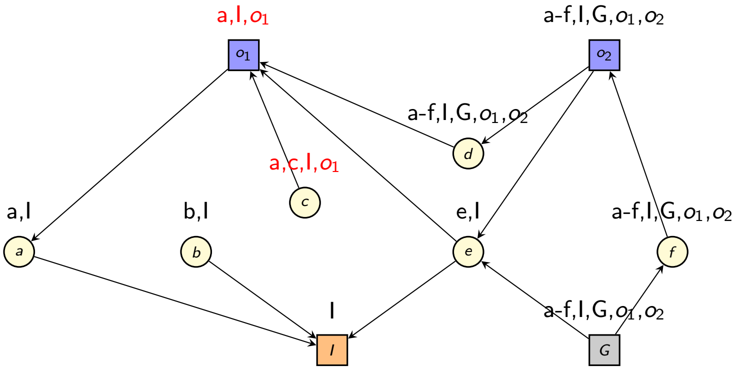
$$LM(e) = \{e\} \cup (LM(I) \cap LM(o_1))$$

Computation: Example



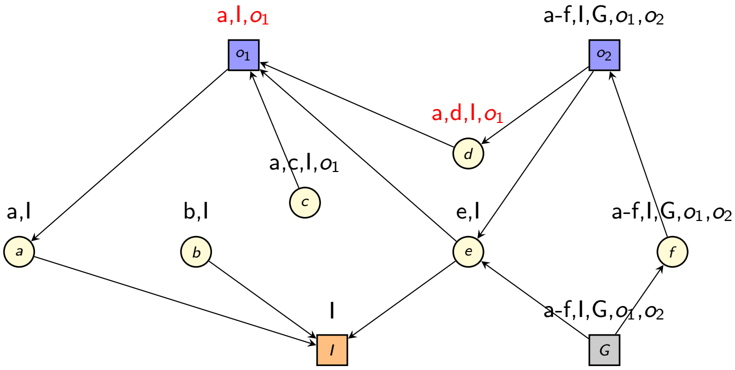
$$LM(o_1) = \{o_1\} \cup LM(a)$$

Computation: Example



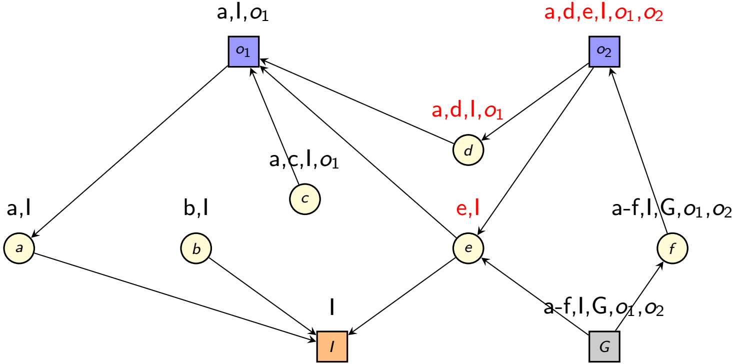
$$LM(c) = \{c\} \cup LM(o_1)$$

Computation: Example



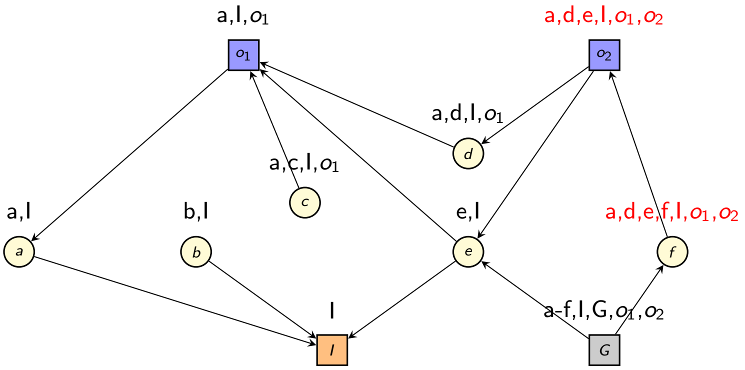
$$LM(d) = \{d\} \cup LM(o_1)$$

Computation: Example



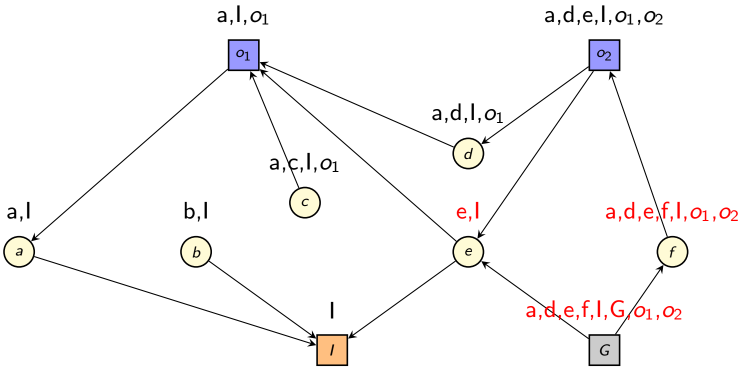
$$LM(o_2) = \{o_2\} \cup LM(d) \cup LM(e)$$

Computation: Example



$$LM(f) = \{f\} \cup LM(o_2)$$

Computation: Example



$$LM(G) = \{G\} \cup LM(e) \cup LM(f)$$

Relation to Planning Task Landmarks

Theorem

Let $\Pi = \langle V, I, O, \gamma \rangle$ be a STRIPS planning task and let $\mathcal{L}$ be the set of landmarks for reaching n_G in $sRTG(\Pi^+)$.

The set $\{v \in V \mid n_v \in \mathcal{L}\}$ is exactly the set of causal fact landmarks in Π^+.

For operators $o \in O$, if $n_o \in \mathcal{L}$ then $\{o\}$ is a disjunctive action landmark in Π^+.
There are no other disjunctive action landmarks of size 1.

(Proofs omitted.)

Computed RTG Landmarks: Example

Example (Computed RTG Landmarks)

Consider a STRIPS planning task $\langle V, I, \{o_1, o_2\}, G\rangle$ with

$$\begin{aligned} V &= \{a, b, c, d, e, f\}, \\ I &= \{a, b, e\}, \\ o_1 &= \langle\{a\}, \{c, d, e\}, \{b\}\rangle, \\ o_2 &= \langle\{d, e\}, \{f\}, \{a\}\rangle, \text{ and} \\ G &= \{e, f\}. \end{aligned}$$

- $LM(n_G) = \{a, d, e, f, I, G, o_1, o_2\}$
- a, d, e, and f are causal fact landmarks of Π^+.
- $\{o_1\}$ and $\{o_2\}$ are disjunctive action landmarks of Π^+.

(Some) Landmarks of Π^+ Are Landmarks of Π

Theorem

Let Π be a STRIPS planning task.

All fact landmarks of Π^+ are fact landmarks of Π and all disjunctive action landmarks of Π^+ are disjunctive action landmarks of Π.

Proof.

Let L be a disjunctive action landmark of Π^+ and π be a plan for Π. Then π is also a plan for Π^+ and, thus, π contains an operator from L.

Let f be a fact landmark of Π^+. If f is already true in the initial state, then it is also a landmark of Π. Otherwise, every plan for Π^+ contains an operator that adds f and the set of all these operators is a disjunctive action landmark of Π^+. Therefore, also each plan of Π contains such an operator, making f a fact landmark of Π. $\square$

Not All Landmarks of Π^+ are Landmarks of Π

Example

Consider STRIPS task $\langle \{a, b, c\}, \emptyset, \{o_1, o_2\}, \{c\} \rangle$ with $o_1 = \langle \{\}, \{a\}, \{\}, 1 \rangle$ and $o_2 = \langle \{a\}, \{c\}, \{a\}, 1 \rangle$.

$a \wedge c$ is a formula landmark of Π^+ but not of Π.

Summary

Summary

- Fact landmark: atomic proposition that is true in each state path to a goal
- Disjunctive action landmark: set L of operators such that every plan uses some operator from L
- We can efficiently compute all causal fact landmarks of a delete-free STRIPS task from the (simplified) RTG.
- Fact landmarks of the delete relaxed task are also landmarks of the original task.