Course "Automated Planning: Theory and Practice"
Chapter 11: Landmarks Heuristics

Teacher: Marco Roveri - marco.roveri@unitn.it
M.S. Course: Artificial Intelligence Systems (LM)
A.A.: 2023-2024
Where: DISI, University of Trento
URL: https://bit.ly/3zOkGk8

Last updated: Thursday 2nd November, 2023

Terms of Use and Copyright

The material (text, figures) in these slides is authored by Jonas Kvarnström and Marco Roveri.

Landmarks

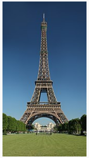
"a geographic feature used by explorers and others
to find their way back or through an area"

A landmark is a recognizable natural or artificial feature used for navigation, a feature that stands out from its near environment and is often visible from long distances. [12]

Natural

https://it.wikipedia.org/wiki/Cervino

Artificial



https://it.wikipedia.org/wiki/Torre_Eiffel

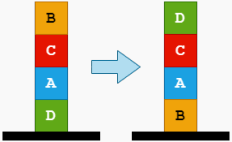
Landmarks in planning

- Something you must achieve or use in every solution to a problem instance!

Assume we are considering a state s...

Fact Landmarks for s

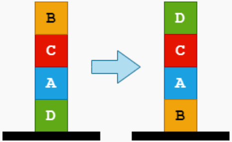
- A fact that must be true at some point in every solution starting from s



`(clear A), (holding C), (clear B), ...`

Formula Landmarks for s

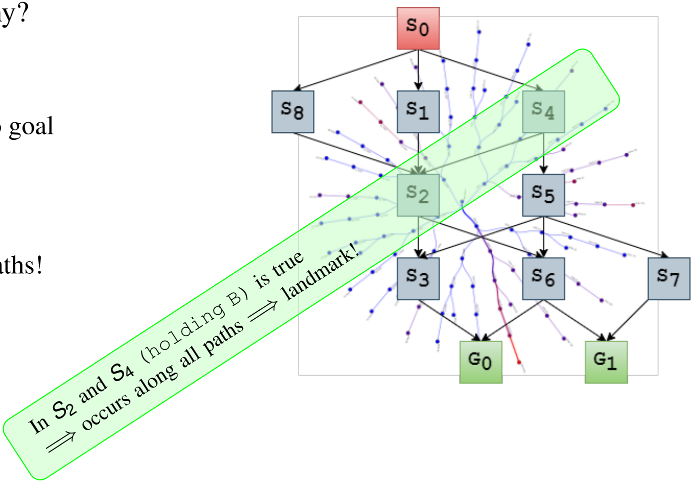
- A formula that must be true at some point in every solution starting from s



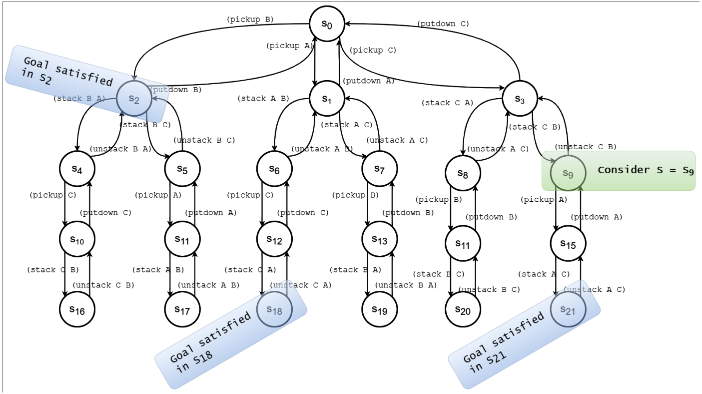
`((clear A)` $\wedge$ `(handempty))` $\vee$ `(clear A) ...`

Landmarks in planning (cont.)

- Facts and formulas, not states! Why?
  - Usually many paths lead from s to goal states!
  - Few states are shared among all paths!
  - Many facts occur along all paths!
    - $\Longrightarrow$ Landmarks!



Landmarks in planning (cont.)



- Is there a landmark state s_{lm} we must pass to reach some goal from s_9?
- No! But we may have to pass different states satisfying the same fact f_{lm}!

Landmarks in planning: Misunderstandings

- Not "we must reach (pass through) the landmark state"!
- Instead "we must reach some state that satisfies the fact/formula landmark"!

- A landmark fact is not "a fact that is true in every solution"!
- A solution is a plan! Facts are true in states!
- A landmark fact is "a fact that is true in some state along every path from the initial state to any goal state"!

- Not "a landmark fact is a state that ...
- A fact is not a state!
  A state consists of many facts!

- Can you be "close" to a landmark?
- You can be in a state s that is close to another state s' satisfying the landmark.
- Problem: How to know?
  - Distance is "number of edges" or "cost of reaching", not $\Delta x / \Delta y$. And the graph may not even be expanded yet!

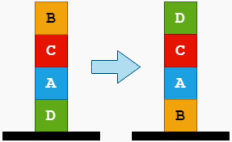
Landmarks in Planning (cont.)

- Something you must pass *by/through* in *every solution* to a specific planning problem

Assume we are considering a state s...

Fact Landmarks for s

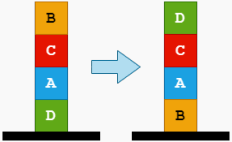
- A fact that must be true at some point in every solution starting from s



`(clear A), (holding C), (clear B), ...`

Action Landmarks for s

- An action that must be used in every solution starting from s



`(unstack B C) (putdown B) (stack D C)`

- ... so the effects of action landmarks are *fact landmarks* and so are their *preconds*

  (except those facts that are already true in s)

Landmarks in Planning (cont.)

- Generalization
  - Disjunctive action landmark $\{ a_1, a_2, a_3 \}$ for state s
    - Every solution starting in state s and reaching a goal must use *at least one* of these actions

Finding Landmarks: General Technique

Current planning problem P

Initial state does not include atom A

Modified planning problem P'

Removed *all actions that add atom* A

If this problem P' is unsolvable ...

Test:
Delete relaxation of P' is unsolvable
or $h_m(s_0) = \infty$, or...
$\Longrightarrow$ P' is unsolvable

... then every solution to P
must use one of the removed actions

$\Longrightarrow$ Action set is a disj.act.landmark

$\Longrightarrow$ Atom A is a fact landmark

- Unsolvable when removing a set of actions
  $\Longrightarrow$ some action in the set must be used $\Longrightarrow$ disjunctive action landmark

Finding Landmarks: General Technique (cont.)

- This is a very general technique
- It is applicable to *any* planning problem, *any* atom!

- General techniques tend to be widely applicable but slow!

Verifying Landmarks

- How difficult is to verify that an action is an action landmark, in the general case?
  - Suppose we can determine if any STRIPS problem P has a solution.
  - We add to the problem a new action:
```
(:action cheat
   :parameters ()
   :precondition ()
   :effect (<goal-formula>))
```
  - If `(cheat)` is an action landmark, then it is *needed* in order to solve the problem $\Longrightarrow$ the original problem was *unsolvable*
- $\Longrightarrow$ As difficult as solving the planning problem (PSPACE-complete) Porteous et al. [15]

Verifying Landmarks (cont.)

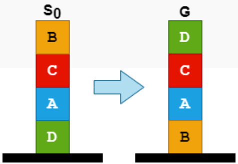
- How difficult is to verify that a fact is a fact landmark, in the general case?
  - Suppose we can determine if any STRIPS problem P has a solution.
  - We add a new fact `(cheated)` (false in the initial state)
  - We add to the problem a new action:

```
(:action cheat
   :parameters ()
   :precondition ()
   :effect (and (cheated) <goal-formula>))
```

  - If `(cheated)` is a fact landmark, then `(cheat)` was necessary $\Longrightarrow$ the original problem was *unsolvable*
- $\Longrightarrow$ Again, as difficult as solving the planning problem Porteous et al. [15]

But of course there are special cases...

FINDING LANDMARKS: MEANS-ENDS ANALYSIS



- Discover landmarks using means-ends analysis

Unachieved goal facts are (obviously) fact landmarks: `(clear D)`, `(on D C)`, `(on A B)`

fact-landmarks $\leftarrow G \setminus s$

Suppose `(on D C)` is a landmark, `(on D C)` is not true in the current state s
$\Longrightarrow$ we must *cause* `(on D C)` with an action
$\Longrightarrow$ compute *achievers* = { `(stack D C)` }

do {
  for each $p \in$ fact-landmarks {
    // Create *disjunctive* action landmark
    achievers $\leftarrow \{a \in A | p \in eff(a)\}$

All achievers require these candidates = { `(holding D)`, `(handempty)`, `(clear C)`,...}

candidates $\leftarrow \bigcap_{a \in achievers} pre(a)$

`(handempty)` is already true, but new = { `(holding D)`, `(clear C)`,...} are *not*!

new $\leftarrow$ candidates $\setminus s$
fact-landmarks $\leftarrow$ fact-landmarks $\cup$ new

May be we can find more landmarks related to achieving *those*!

  }
} **until** no more fact-landmarks found

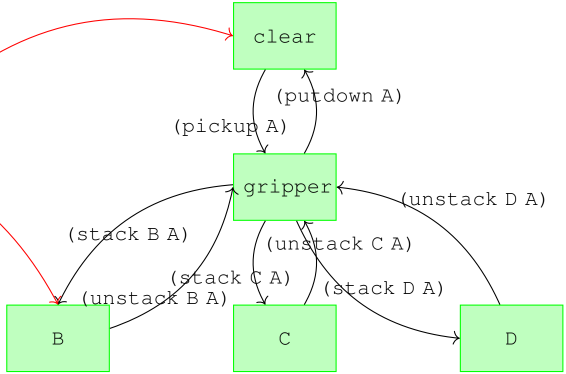
Finding Landmarks: Actions, Backward

- Effects of disjunctive action landmarks:
  - All shared effects must also take place regardless of the "chosen" action, similarly to shared *preconditions* on previous slide
  - Given a disjunctive action landmark, every fact in $\cap_{a \in landmark}\{eff(a)\} \setminus s$ is a fact landmark for s

Finding Landmarks: Domain Transition Graphs

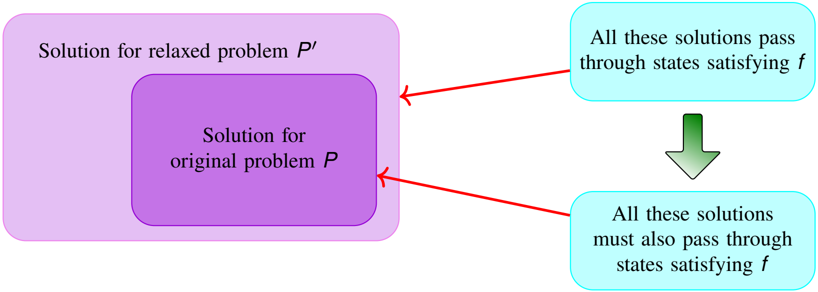
- A directed graph for a state variable x whose vertex corresponds to possible values the state variable can take (i.e. its domain D_x), and there is an arc (v, v') between two possible values v, v' if there is an action o such that $v \in \mathit{eff}^-(o)$ and $v' \in \mathit{eff}^+(o)$.

- Use of the DTG
  - Current state: `aboveA = clear`
  - In goal: `aboveA = B`
- Then `aboveA = gripper` is a fact landmark...
  - ... and `(pickup A)` + `(stack B A)` are action landmarks



Landmarks and Relaxation

- Assume a problem P, and a relaxed problem P'
  - Suppose f is a fact landmark for P'



  - Then f is a fact landmark for the original problem as well!!
  - Similar for action landmarks, etc..

Finding Landmarks: ...

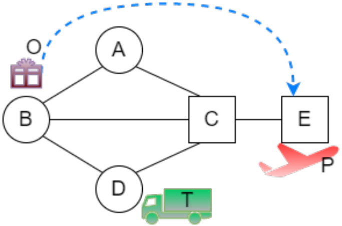
- Many other techniques exists...
  - It is beyond the scope of the course
  - Possible reference: Hoffmann et al. [8], Richter et al. [16], Zhu and Givan [17], Gregory et al. [7], Keyder et al. [11], Karpas and Domshlak [9], Domshlak et al. [2], Domshlak et al. [3], Mirkis and Domshlak [14]

Landmark Ordering

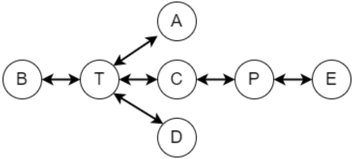
- Landmarks can be (partially) ordered according to the order in which they must be achieved
- Sometimes we can find or approximate necessary orderings
  - We must achieve `(holding A)` then `(holding B)` ...
- Some landmarks and orderings can be discovered automatically

Landmark Ordering: Example Problem [10]

- Domain:
  - Truck T transports object O with road network A/B/C/D
  - Airplane P transports object O between airports C/E
- Goal
  - Object at E



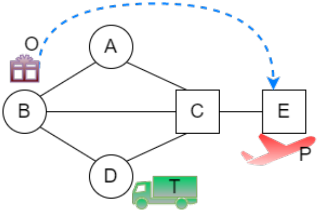
- Domain transition graph for `location-of-object`:

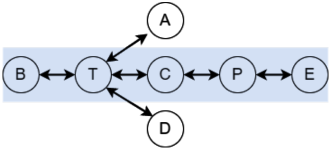


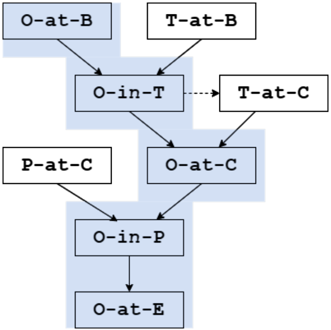
- Note: Every edge in the road network corresponds to a path through T in the DTG!

Landmark Ordering: Inference Karpas and Richter [10]

- One way of inferring the order of landmarks:
  - Directly from the DTG!







Landmark Ordering: Types of Landmark Orderings

- Two kind of landmark ordering
  - Sound landmark orderings are guaranteed to hold - they do not prune the solution space
  - Unsound landmark orderings are additional constraints on plans - they may prune the solution space
- It is even possible that no plan exists that respects the unsound orderings
- However, unsound orderings are likely to hold and may save effort in planning

Landmark Ordering: Sound Landmark Orderings

- *Natural ordering* $A \rightarrow B$, iff A true some time before B
- *Necessary ordering* $A \rightarrow_n B$, iff A always true one step before B becomes true
- *Greedy-necessary ordering* $A \rightarrow_{gn} B$, iff A true one step before B becomes true for the first time

- Note that: $A \rightarrow_n B \Rightarrow A \rightarrow_{gn} B \Rightarrow A \rightarrow B$

Encoding in LTL with past operators (Lichtenstein et al. [13])

- $\overline{A} \rightarrow \overline{B}$, iff $\overline{A}$ true some time before $\overline{B}$ i.e., $\overline{B} \rightarrow \mathbf{Y}\,\mathbf{O}\,\overline{A}$ which means that given a path π, $\pi, i \models \overline{B} \rightarrow \mathbf{Y}\,\mathbf{O}\,\overline{A}$ that is if $\overline{B}$ holds at time instant i, then there shall be a $j < i$ such that $\overline{A}$ holds.
- $\overline{A} \rightarrow_n \overline{B}$, iff $\overline{A}$ always true one step before $\overline{B}$ becomes true, i.e., $(\overline{B} \wedge \mathbf{Y}\,\neg\overline{B}) \rightarrow \mathbf{Y}\,\mathbf{H}\,\overline{A}$ which means that given a path π, $\pi, i \models (\overline{B} \wedge \mathbf{Y}\,\neg\overline{B}) \rightarrow \mathbf{Y}\,\mathbf{H}\,\overline{A}$ that is if $\overline{B}$ holds at time instant i and $\neg\overline{B}$ holds at $i-1$ ($\overline{B}$ becomes true), then at $\forall j \leq i-1$ $\overline{A}$ holds.
- $\overline{A} \rightarrow_{gn} \overline{B}$, iff $\overline{A}$ true one step before $\overline{B}$ becomes true for the first time, i.e., $(\overline{B} \wedge \mathbf{Y}\,\mathbf{H}\,\neg\overline{B}) \rightarrow \mathbf{Y}\,\overline{A}$ which means that given a path π, $\pi, i \models (\overline{B} \wedge \mathbf{Y}\,\mathbf{H}\,\neg\overline{B}) \rightarrow \mathbf{Y}\,\overline{A}$, that is if at time i $\overline{B}$ holds and all $j < i$ are such that $\neg\overline{B}$ holds ($\overline{B}$ becomes true for the first time), then at step $j = i-1$ $\overline{A}$ holds.

From the above considerations it is easy to see: $A \rightarrow_n B \Rightarrow A \rightarrow_{gn} B \Rightarrow A \rightarrow B$

The used past temporal operators are $\mathbf{Y}$ = Yesterday, $\mathbf{H}$ = Historically, $\mathbf{O}$ = Once in the past, they are the past correspondence of $\mathbf{X}$ = in the next step, $\mathbf{G}$ = Globally, and $\mathbf{F}$ = Eventually. See Lichtenstein et al. [13] for details on LTL with past operators!

Landmark Ordering: Reasonable Landmark Orderings

- Not sound - not guaranteed to hold in all plans
- Reasonable ordering $A \rightarrow_r B$, iff given B was achieved before A, any plan must delete B on the way to A, and re-achieve/achieve B after or at the same time as A

$$B \rightsquigarrow \neg B \rightsquigarrow A \rightsquigarrow B \Rightarrow A \rightarrow_r B$$

- Initial state landmarks can be reasonably ordered after other landmarks (e. g., if they must be made false and true again)
- This can never happen with sound orderings

Landmark Ordering: Complexity

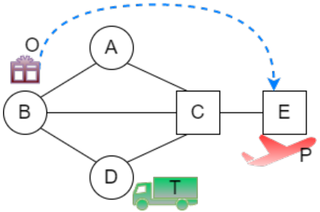
- Everything is PSPACE-complete
- Deciding if a given fact is a landmark is PSPACE-complete
  - Proof Sketch: it's the same as deciding if the problem without actions that achieve this fact is unsolvable
- Deciding if there is a natural / necessary / greedy-necessary / reasonable ordering between two landmarks is PSPACE-complete

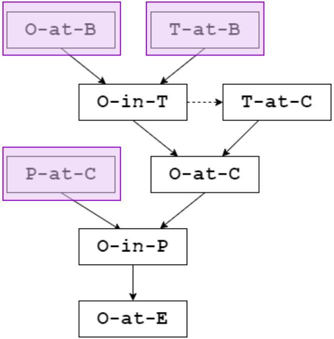
Ordered Landmarks as Subgoals

- Landmarks can be used as subgoals for a base planner
- The landmarks which could be achieved in the next iteration are passed as a disjunctive goal to a base planner
- After a landmark is achieved, repeat

Ordered Landmarks as Subgoals (cont.)

- Try to plan for each landmark separately in the inferred order

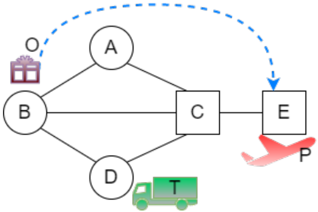


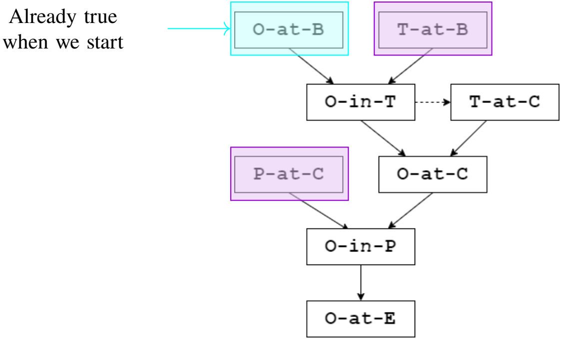


- Three landmarks could be "first" (all predecessors achieved)
- Current goal: O-at-B $\vee$ T-at-B $\vee$ P-at-C (disjunctive)!

Ordered Landmarks as Subgoals (cont.)

- Try to plan for each landmark separately in the inferred order

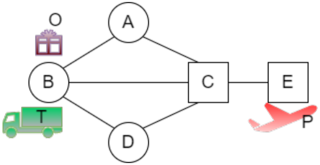




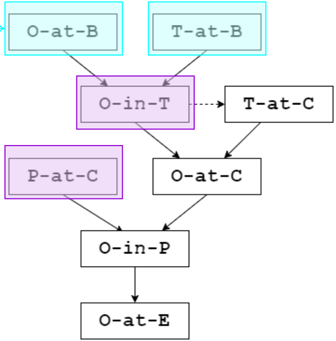
- Two landmarks could be "first" (all predecessors achieved)
- Current goal: `T-at-B` ∨ `P-at-C` (disjunctive)!

Ordered Landmarks as Subgoals (cont.)

- Suppose we begin by achieving `T-at-B`: simple planning problem, results in a single action – `(drive T B)`



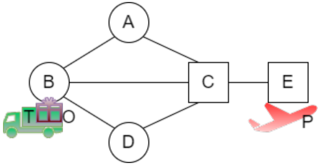
Already true when we start

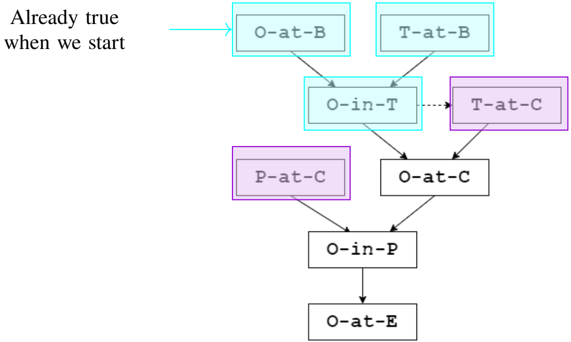


- Current goal: `O-in-T` ∨ `P-at-C`

Ordered Landmarks as Subgoals (cont.)

- Suppose we continue by achieving `O-at-B`: simple planning problem, results in a single action – `(load-truck O T B)`

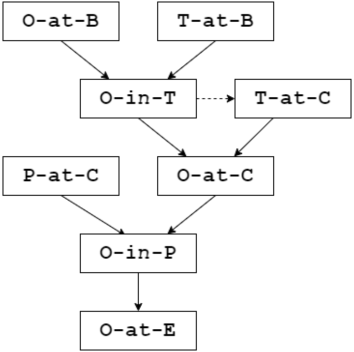




- Current goal: `P-at-C` $\vee$ `T-at-C`

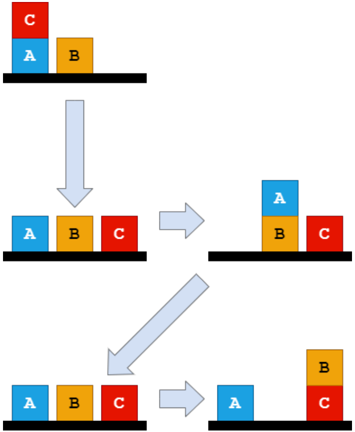
Landmarks as Subgoals (cont.)

- Sometimes very helpful, but:
  - There are still choices to be made - backtrack points!
  - Optimizing one part of the overall goal at a time:
    - Can't see the whole picture
    - Can miss opportunities:
      Cheapest solution here $\Longrightarrow$ more expressive solution later!
    - Can be incomplete:
      Cheapest solution here $\Longrightarrow$ impossible to solve later!



The Problem with Separating Subgoals

- The Sussman Anomaly (Gerald Sussman)
  - Goal: `(on A B), (on B C)`
  - Now: `(on C A), (clear C), (clear B), (ontable A), (ontable B)`
- Idea: Achieve one at a time!
  - First, plan only for `(on A B)`
  - Then, plan only for `(on B C)`
- Achieve first subgoal `(on A B)`
  - `(unstack C A), (putdown C), (pickup A), (stack A B)`
- Achieve second subgoal `(on B C)`
  - `(unstack A B), (putdown A), (pickup B), (stack B C)` $\Longrightarrow$ original goal destroyed!

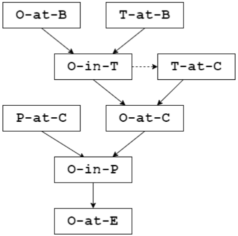


Using Landmarks as Subgoals - Pros and Cons

- Pros:
  - Planning is very fast - the base planner needs to plan to a lesser depth
- Cons:
  - Can lead to much longer plans
  - Not complete in the presence of dead-ends

Landmark Heuristics

- The LAMA state space planner counts landmarks:
  - Landmarks that need to be achieved after reaching state s through action sequence π
  - $L(s, \pi) =$



$$L \setminus Accepted(s, \pi) \quad \cup \quad ReqAgain(s, \pi)$$

All discovered landmarks, minus those that are *accepted* as achieved (have become true after predecessors are achieved!)

Plus those we can show will have to be *re-achieved*

(Example: Landmarks that were reached, are no longer true, but are required by the goal)

- $h(s) = |L(s, \pi)|$
- Not admissible: One action may achieve multiple landmarks!

Accepted Landmarks

- In LAMA, a landmark A is first accepted by path π in state s if
    - all predecessors of A in the landmark graph have been accepted, and
    - A becomes true in s
- Once a landmark has been accepted, it remains accepted

Required Again Landmarks

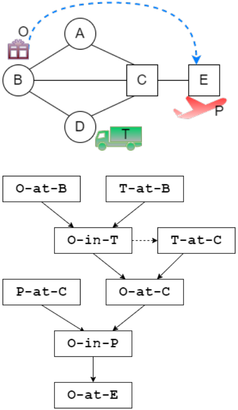
- A landmark A is required again by path π in state s if:
  - false-goal A is false in s and is a goal, or
  - open-prerequisite A is false in s and is a greedy-necessary predecessor of some landmark B that is not accepted
- It's also possible to use approach described in Buffet and Hoffmann [1]
  - doomed-goal A is true in s and is a goal, but one of its greedy-necessary successors was not accepted, and is inconsistent with A
- Unsound rule:
  - required-ancestor is the transitive closure of open-prerequisite

Landmark Heuristics (cont.)

- To achieve admissible heuristic estimates:
  - Idea: The cost of each action is *divided* across the landmarks it achieves

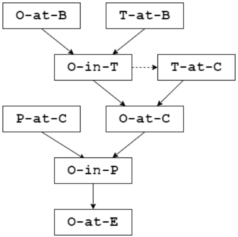
- Simplified example
  - Suppose there is a `goto-and-pick` action of cost 10, that achieves both `T-at-B` and `O-in-T`
  - Suppose *no other action* can achieve these landmarks
  - One can then let (for example) $cost(\texttt{T-at-B}) = 3$ and $cost(\texttt{O-in-T}) = 7$

- The sum of the cost of remaining landmarks is then an admissible heuristic
  - Must decide how to split costs across landmarks
  - Optimal split can be computed polinomially, but it is still expensive



Landmarks: Modified Problem

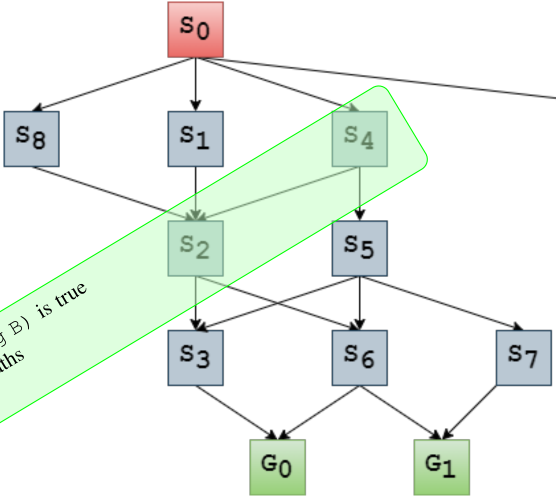
- Landmarks are, in essence, implicit goals
  - We can make these explicit by reformulating the planning problem
- Landmarks as a basis for a modified planning problem Domshlak et al. [2]
  - Add new facts "`achieved=landmark-n`"
    - Example: `object-has-been-in-plane`
  - An action achieving a landmark makes the corresponding facts true
    - `(load object plane)` $\Longrightarrow$ `object-has-been-in-plane` := `true`
  - The goal requires all such facts to be true!
    - `(:goal object-has-been-in-plane ...)`
  - $\Longrightarrow$ Any *other* heuristic can be applied to the modified problem!
    - $h_{PDB}(s)$ - pattern databases: What is the cost of achieving `object-has-been-in-plane`?



Why Modifying Problems?

- Landmarks and orderings are implicit, encoding them into the problem makes them explicit
- Allows other heuristics to use landmark information
  - Example: structural pattern heuristic on the enriched problem accounts not only for explicit goals Domshlak et al. [2]
  - In fact, the landmark count heuristic can be seen as the goal count heuristic on the landmark enriched problem
- Caveat – since current landmark discovery procedures are based on delete-relaxation, this adds no information to delete-relaxation based heuristics (discussed later).

Landmarks: More Misunderstandings



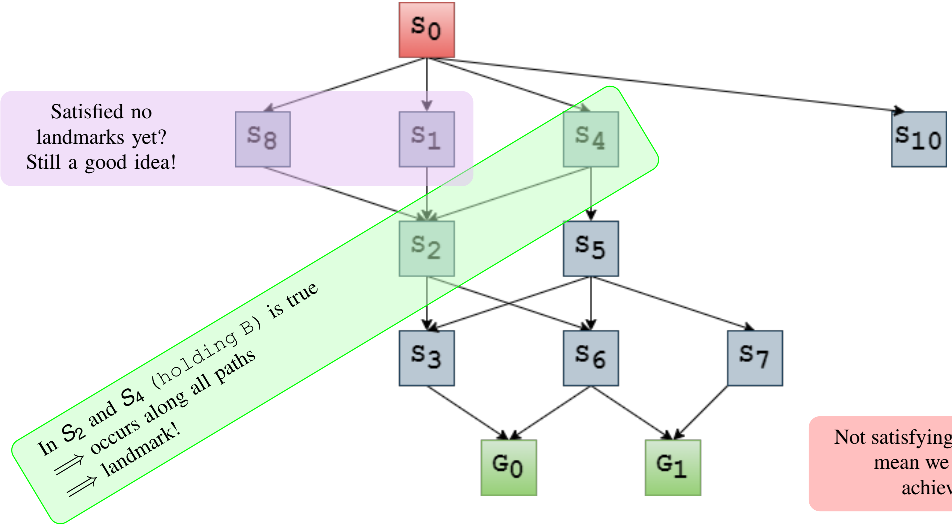
In S_2 and S_4 `(holding B)` is true
$\implies$ occurs along all paths
$\implies$ landmark!

Also happens to satisfy `(holding B)`

Might still be a bad idea!

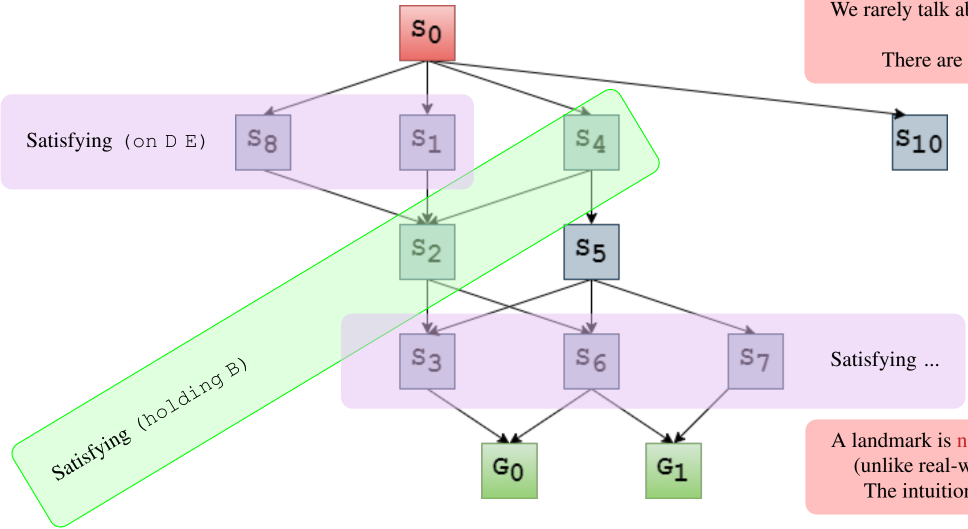
Satisfying a landmark does not mean we must be close to achieving the goal!

Landmarks: More Misunderstandings (cont.)



Not satisfying a landmark does not mean we can't be close to achieving the goal!

Landmarks: More Misunderstandings (cont.)



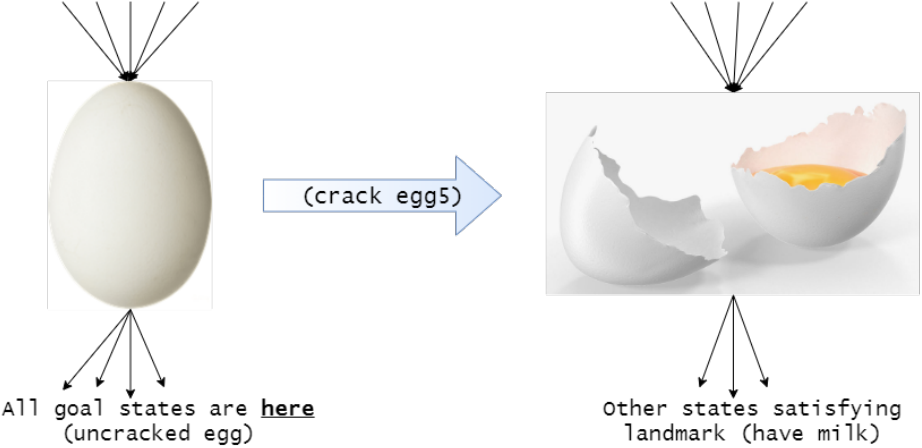
We rarely talk about "the" landmark

There are often many ...

A landmark is not a unique location (unlike real-world landmarks): The intuition breaks down...

Landmarks: Ideas that Don't Work

"If I reach a state satisfying a landmark, I won't have to backtrack"!



Landmarks: Commented references

- Hoffmann et al. [8]: Introduces landmarks and the backchaining generation method.
- Richter et al. [16]: Describes the DTG method for finding landmarks and the landmark heuristic used by LAMA.
- Zhu and Givan [17]: Describes the method for finding the complete set of causal delete-relaxation landmarks in polynomial time.
- Gregory et al. [7]: Further methods for finding disjunctive landmarks.
- Keyder et al. [11]: Conjunctive landmarks (not discussed here); explains how to approximate greedy-necessary orderings in Zhu & Givan's landmark discovery method.
- Karpas and Domshlak [9]: Describes the admissible landmark heuristic and multi-path dependence.
- Domshlak et al. [2] and Domshlak et al. [3]: Describes how to compile landmarks into a planning problem and use them with abstraction heuristics.
- Mirkis and Domshlak [14]: Describe a framework for exploiting such landmarks in heuristic-search for over subscription planning.

References I

[1] Olivier Buffet and Joerg Hoffmann. All that glitters is not gold: Using landmarks for reward shaping in fpg. *ICAPS-10 Workshop on Planning and Scheduling Under Uncertainty*, 05 2010.

[2] Carmel Domshlak, Michael Katz, and Sagi Lefler. When abstractions met landmarks. In Ronen I. Brafman, Hector Geffner, Jörg Hoffmann, and Henry A. Kautz, editors, *Proceedings of the 20th International Conference on Automated Planning and Scheduling, ICAPS 2010, Toronto, Ontario, Canada, May 12-16, 2010*, pages 50–56. AAAI, 2010. URL `http://www.aaai.org/ocs/index.php/ICAPS/ICAPS10/paper/view/1431`. 18

[3] Carmel Domshlak, Michael Katz, and Sagi Lefler. Landmark-enhanced abstraction heuristics. *Artif. Intell.*, 189:48–68, 2012. doi: 10.1016/j.artint.2012.05.003. URL `https://doi.org/10.1016/j.artint.2012.05.003`. 18

[4] Hector Geffner and Blai Bonet. *A Concise Introduction to Models and Methods for Automated Planning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2013. ISBN 9781608459698. doi: 10.2200/S00513ED1V01Y201306AIM022. URL `https://doi.org/10.2200/S00513ED1V01Y201306AIM022`.

[5] Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated planning - theory and practice*. Elsevier, 2004. ISBN 978-1-55860-856-6.

[6] Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated Planning and Acting*. Cambridge University Press, 2016. ISBN 978-1-107-03727-4. URL `http://www.cambridge.org/de/academic/subjects/computer-science/artificial-intelligence-and-natural-language-processing/automated-planning-and-acting?format=HB`.

References II

[7] P. Gregory, S. Cresswell, D. Long, and J. Porteous. On the extraction of disjunctive landmarks from planning problems via symmetry reduction. In W. Harvey and Z. Kiziltan, editors, *Proceedings of SymCon'04: The 4th International Workshop on Symetry and Constraint Satisfaction Problems*, pages 34–41. Springer, 2004. URL `http://zeynep.web.cs.unibo.it/SymCon04/GregoryCresswellLongPorteous.pdf`. 18

[8] Jörg Hoffmann, Julie Porteous, and Laura Sebastia. Ordered landmarks in planning. *J. Artif. Intell. Res.*, 22:215–278, 2004. doi: 10.1613/jair.1492. URL `https://doi.org/10.1613/jair.1492`. 18

[9] Erez Karpas and Carmel Domshlak. Cost-optimal planning with landmarks. In Craig Boutilier, editor, *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1728–1733, 2009. URL `http://ijcai.org/Proceedings/09/Papers/288.pdf`. 18

[10] Erez Karpas and Silvia Richter. Landmarks - Definition, Discovery Methods and Uses – ICAPS 2010 Tutorial, 2010. `https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.423.1518&rep=rep1&type=pdf`. 21, 22

[11] Emil Keyder, Silvia Richter, and Malte Helmert. Sound and complete landmarks for and/or graphs. In Helder Coelho, Rudi Studer, and Michael J. Wooldridge, editors, *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 335–340. IOS Press, 2010. URL `http://www.booksonline.iospress.nl/Content/View.aspx?piid=17768`. 18

[12] L2021. Landmark (wikipedia), 2021. `https://en.wikipedia.org/wiki/Landmark`. 3

References III

[13] Orna Lichtenstein, Amir Pnueli, and Lenore D. Zuck. The glory of the past. In *Logic of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 196–218. Springer, 1985. 25

[14] Vitaly Mirkis and Carmel Domshlak. Landmarks in oversubscription planning. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 633–638. IOS Press, 2014. doi: 10.3233/978-1-61499-419-0-633. URL `https://doi.org/10.3233/978-1-61499-419-0-633`. 18

[15] Julie Porteous, Laura Sebastia, and Jorg Hoffmann. On the extraction, ordering, and usage of landmarks in planning. 2001. Requires Template change to Chapter in Book/Report/Conference proceeding Conference contribution: Proceedings of the 6th European Conference on Planning (ECP 01); 6th European Conference on Planning, ECP 2001 ; Conference date: 12-09-2001 Through 14-07-2016. 12, 13

[16] Silvia Richter, Malte Helmert, and Matthias Westphal. Landmarks revisited. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 975–982. AAAI Press, 2008. URL `http://www.aaai.org/Library/AAAI/2008/aaai08-155.php`. 18

[17] Lin Zhu and Robert Givan. Landmark extraction via planning graph propagation. In *In ICAPS Doctoral Consortium*, 2003. 18