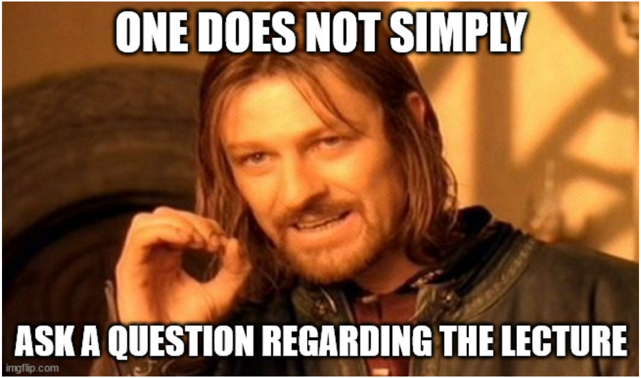
Landmark heuristics

Michaela Urbanovská

PUI Tutorial
Week 4

Lecture check

- Any questions regarding the lecture?



Feedback

- 4 answers (yaaay!)
- Tempo of lectures + tutorials OK

LM-Cut

LM-Cut Heuristic

- Relaxation heuristic
- Uses disjunctive operator landmarks
- Admissible (and actually very successful) heuristic

Disjunctive operator landmark

Disjunctive operator landmark $L \subseteq O$ is set of operators such that every plan π contains at least one operator $o \in L$.

Previously...

$F = \{a, b, c, d, e, f, g\}$

$s_I = \{a, b\}$

$s_G = \{f, g\}$

$O =$

	pre	add	del	c
o_1	a	c, d	a	1
o_2	a, b	e	$\emptyset$	
o_3	b, e	d, f	a, e	
o_4	b	a	$\emptyset$	
o_5	d, e	g	e	

	a	b	c	d	e	f	g
Δ_1	0	0	$\emptyset$ 1	$\emptyset$ 1	$\emptyset$ 1	$\emptyset$ 2	$\emptyset$ 2

	o_1	o_2	o_3	o_4	o_5
U	1	2	2	1	2
	0	1 → 0	1 → 0	0	1 → 0

$k = a$: o_1, o_2

$k = b$: o_2, o_3, o_4

$k = c$

$k = d$: o_5

$k = e$: o_3, o_5

$k = f$

$k = g$

$C = \{a, b, c, d, e, f, g\}$

$h^{max}(s_I) = 2$

$h^{add}(s_I) = 4$

LM-Cut

Delta function

- Function Δ_1 from previous tutorial:
- $\Delta_1(s, f) =$

$$\begin{cases} 0 & \text{if } f \in s, \\ \text{inf} & \text{if } \forall o \in O : f \notin add(o), \\ min\{c(o) + \Delta_1(s, o) | o \in O, f \in add(o)\} & otherwise. \end{cases}$$

LM-Cut

Supporter of an operator

Supporter is a **fact**.
Function $supp(o) = argmax_{f \in pre(o)} = \Delta_1(s, f)$ maps each operator $o \in O$ to its **supporter**.
(s denotes the state where we compute the heuristic estimate)

LM-Cut

Justification graph

$G = (N, E)$ is a **directed labeled** multigraph.

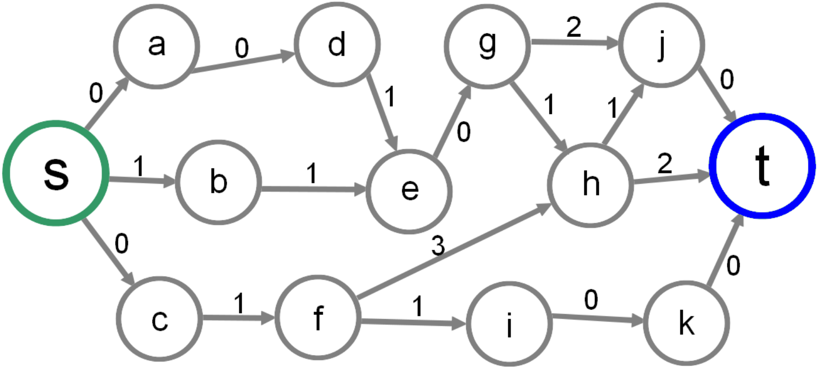
- $N = \{n_f | f \in F\}$ *(set of nodes)*
- $E = \{(n_s, n_t, o) | o \in O, s = supp(o), t \in add(o)\}$ *(set of edges)*
- Edge $e = (a, b, l)$ denotes edge from a to b with label l

LM-Cut

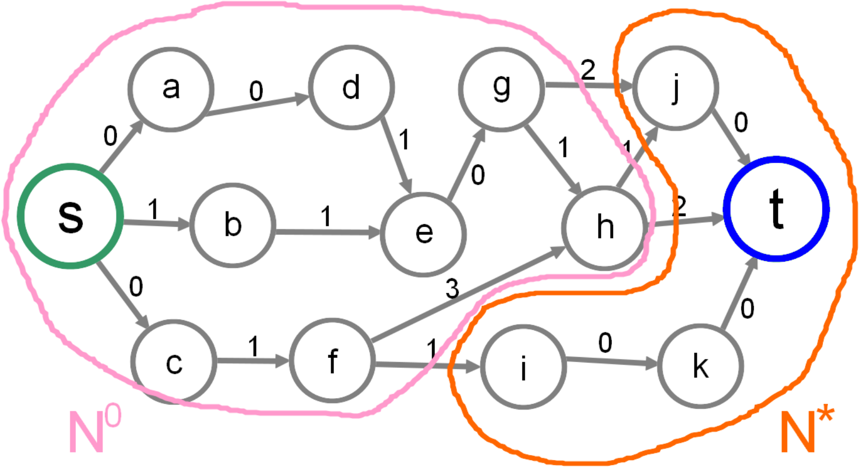
s-t cut

- **s-t cut** $C(G, s, t) = (N^0, N^* \cup N^b)$
- partitioning of nodes from the **justification graph** $G = (N, E)$
- N^* contains nodes from which t can be reached with zero-cost path
- N^0 contains nodes which can be reached from s without passing any node from N^*
- $N^b = N \setminus (N^0 \cup N^*)$ *(all the other nodes)*

LM-Cut



LM-Cut



LM-Cut

Algorithm 2: Algorithm for computing $\mathrm{h}^{\text{lm-cut}}(s)$.

Input: $\Pi = \langle \mathcal{F}, \mathcal{O}, s_{init}, s_{goal}, \mathrm{c} \rangle$, state s

Output: $\mathrm{h}^{\text{lm-cut}}(s)$

if $\mathrm{h}^{\max}(\Pi, s_{init}) = \infty$ **then**

    $\mathrm{h}^{\text{lm-cut}}(s) \leftarrow \infty$ and terminate;

end

$\mathrm{h}^{\text{lm-cut}}(s) \leftarrow 0$;

$\Pi_1 = \langle \mathcal{F}' = \mathcal{F} \cup \{I, G\}, \mathcal{O}' = \mathcal{O} \cup \{o_{init}, o_{goal}\}, s'_{init} = \{I\}, s'_{goal} = \{G\}, \mathrm{c}_1 \rangle$, where $\mathrm{pre}(o_{init}) = \{I\}$, $\mathrm{add}(o_{init}) = s$, $\mathrm{del}(o_{init}) = \emptyset$, $\mathrm{pre}(o_{goal}) = s_{goal}$, $\mathrm{add}(o_{goal}) = \{G\}$, $\mathrm{del}(o_{goal}) = \emptyset$, $\mathrm{c}_1(o_{init}) = 0$, $\mathrm{c}_1(o_{goal}) = 0$, and $\mathrm{c}_1(o) = \mathrm{c}(o)$ for all $o \in \mathcal{O}$;

$i \leftarrow 1$;

while $\mathrm{h}^{\max}(\Pi_i, s'_{init}) \neq 0$ **do**

    Construct a justification graph G_i from Π_i;

    Construct an s-t-cut $\mathcal{C}_i(G_i, n_I, n_G) = (N_i^0, N_i^\star \cup N_i^b)$;

    Create a landmark L_i as a set of labels of edges that cross the cut $\mathcal{C}_i$, i.e., they lead from N_i^0 to $N_i^\star$;

    $m_i \leftarrow \min_{o \in L_i} \mathrm{c}_i(o)$;

    $\mathrm{h}^{\text{lm-cut}}(s) \leftarrow \mathrm{h}^{\text{lm-cut}}(s) + m_i$;

    Set $\Pi_{i+1} = \langle \mathcal{F}', \mathcal{O}', s'_{init}, s'_{goal}, \mathrm{c}_{i+1} \rangle$, where $\mathrm{c}_{i+1}(o) = \mathrm{c}_i(o) - m_i$ if $o \in L_i$, and $\mathrm{c}_{i+1}(o) = \mathrm{c}_i(o)$ otherwise;

    $i \leftarrow i + 1$;

end

LM-Cut

- Sustainability of class materials is also important! So we're reusing!
- Last year's materials: here

Recap

- LM-Cut implementation
- Everything you need to do the assignment 1!

Feedback form link