Artificial Intelligence

Planning 4: Delete Relaxation Heuristics

Jendrik Seipp

Linköping University

based on slides by Thomas Keller and Malte Helmert (University of Basel)

Relaxed Planning Graphs

Relaxed Planning Graphs

- relaxed planning graphs: represent which variables in Π^+ can be reached and how
- graphs with variable layers V^i and action layers A^i
  - variable layer V^0 contains the variable vertex v^0 for all $v \in I$
  - action layer A^{i+1} contains the action vertex a^{i+1} for action a if V^i contains the vertex v^i for all $v \in pre(a)$
  - variable layer V^{i+1} contains the variable vertex v^{i+1} if previous variable layer contains v^i, or previous action layer contains a^{i+1} with $v \in add(a)$

Relaxed Planning Graphs (Continued)

- goal vertices G^i if $v^i \in V^i$ for all $v \in G$
- graph can be constructed for arbitrary many layers but stabilizes after a bounded number of layers $\rightsquigarrow V^{i+1} = V^i$ and $A^{i+1} = A^i$
- directed edges:
  - from v^i to a^{i+1} if $v \in pre(a)$ (precondition edges)
  - from a^i to v^i if $v \in add(a)$ (effect edges)
  - from v^i to G^i if $v \in G$ (goal edges)
  - from v^i to v^{i+1} (no-op edges)

Illustrative Example

we write actions a with $pre(a) = \{p_1, \ldots, p_k\}$, $add(a) = \{a_1, \ldots, a_l\}$, $del(a) = \emptyset$ and $cost(a) = c$
as $\{p_1, \ldots, p_k\} \xrightarrow{c} \{a_1, \ldots, a_l\}$

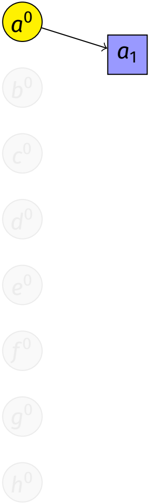
$$
\begin{aligned}
V &= \{a, b, c, d, e, f, g, h\} \\
I &= \{a\} \\
G &= \{c, d, e, f, g\} \\
A &= \{a_1, a_2, a_3, a_4, a_5, a_6\} \\
a_1 &= \{a\} \xrightarrow{3} \{b, c\} \\
a_2 &= \{a, c\} \xrightarrow{1} \{d\} \\
a_3 &= \{b, c\} \xrightarrow{1} \{e\} \\
a_4 &= \{b\} \xrightarrow{1} \{f\} \\
a_5 &= \{d\} \xrightarrow{1} \{e, f\} \\
a_6 &= \{d\} \xrightarrow{1} \{g\}
\end{aligned}
$$

Illustrative Example: Relaxed Planning Graph

$a_1 = \{a\} \xrightarrow{3} \{b, c\}$ $\quad a_2 = \{a, c\} \xrightarrow{1} \{d\}$ $\quad a_3 = \{b, c\} \xrightarrow{1} \{e\}$

$a_4 = \{b\} \xrightarrow{1} \{f\}$ $\quad a_5 = \{d\} \xrightarrow{1} \{e, f\}$ $\quad a_6 = \{d\} \xrightarrow{1} \{g\}$

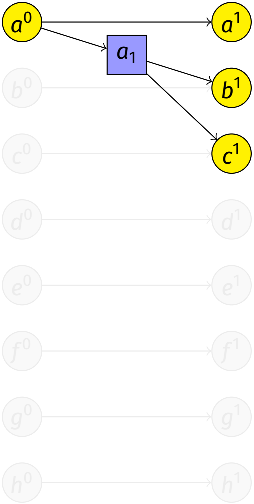
Illustrative Example: Relaxed Planning Graph



$a_1 = \{a\} \xrightarrow{3} \{b, c\}$ $\quad a_2 = \{a, c\} \xrightarrow{1} \{d\}$ $\quad a_3 = \{b, c\} \xrightarrow{1} \{e\}$

$a_4 = \{b\} \xrightarrow{1} \{f\}$ $\quad a_5 = \{d\} \xrightarrow{1} \{e, f\}$ $\quad a_6 = \{d\} \xrightarrow{1} \{g\}$

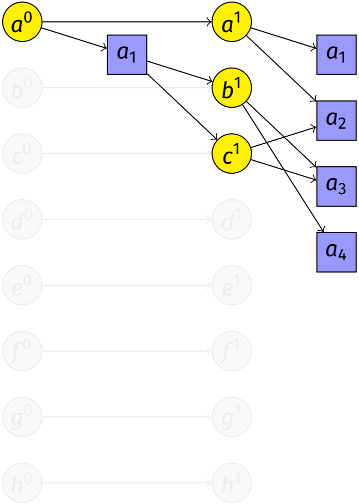
Illustrative Example: Relaxed Planning Graph



$a_1 = \{a\} \xrightarrow{3} \{b, c\}$ $a_2 = \{a, c\} \xrightarrow{1} \{d\}$ $a_3 = \{b, c\} \xrightarrow{1} \{e\}$

$a_4 = \{b\} \xrightarrow{1} \{f\}$ $a_5 = \{d\} \xrightarrow{1} \{e, f\}$ $a_6 = \{d\} \xrightarrow{1} \{g\}$

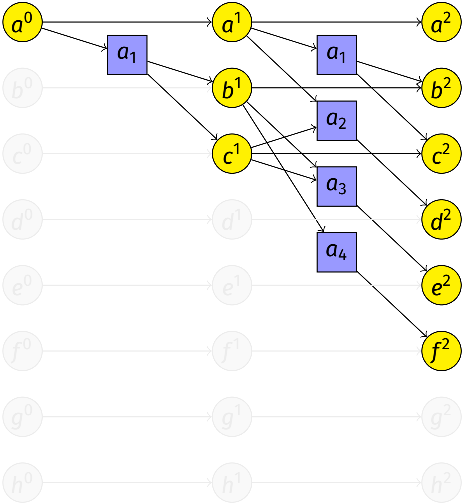
Illustrative Example: Relaxed Planning Graph



$a_1 = \{a\} \xrightarrow{3} \{b, c\}$ $\quad a_2 = \{a, c\} \xrightarrow{1} \{d\}$ $\quad a_3 = \{b, c\} \xrightarrow{1} \{e\}$

$a_4 = \{b\} \xrightarrow{1} \{f\}$ $\quad a_5 = \{d\} \xrightarrow{1} \{e, f\}$ $\quad a_6 = \{d\} \xrightarrow{1} \{g\}$

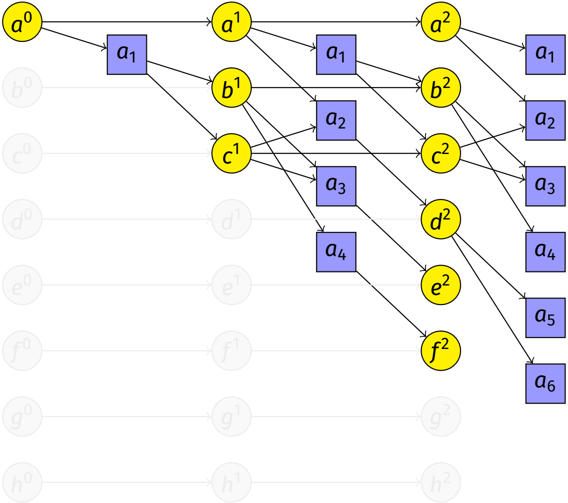
Illustrative Example: Relaxed Planning Graph



$a_1 = \{a\} \xrightarrow{3} \{b, c\}$ $\quad a_2 = \{a, c\} \xrightarrow{1} \{d\}$ $\quad a_3 = \{b, c\} \xrightarrow{1} \{e\}$

$a_4 = \{b\} \xrightarrow{1} \{f\}$ $\quad a_5 = \{d\} \xrightarrow{1} \{e, f\}$ $\quad a_6 = \{d\} \xrightarrow{1} \{g\}$

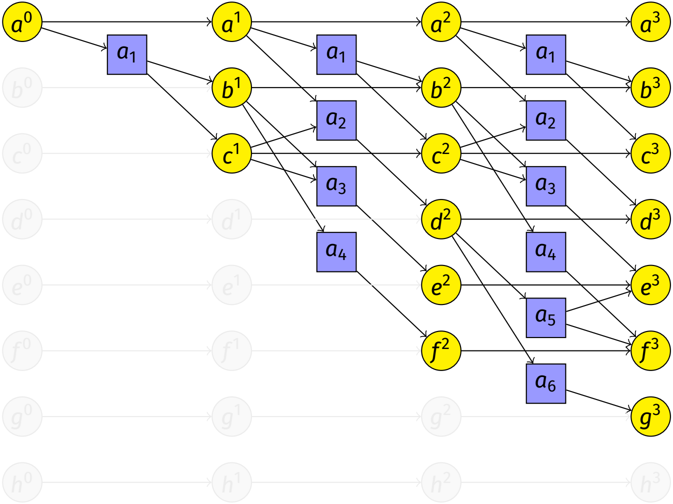
Illustrative Example: Relaxed Planning Graph



$a_1 = \{a\} \xrightarrow{3} \{b, c\}$ $\quad a_2 = \{a, c\} \xrightarrow{1} \{d\}$ $\quad a_3 = \{b, c\} \xrightarrow{1} \{e\}$

$a_4 = \{b\} \xrightarrow{1} \{f\}$ $\quad a_5 = \{d\} \xrightarrow{1} \{e, f\}$ $\quad a_6 = \{d\} \xrightarrow{1} \{g\}$

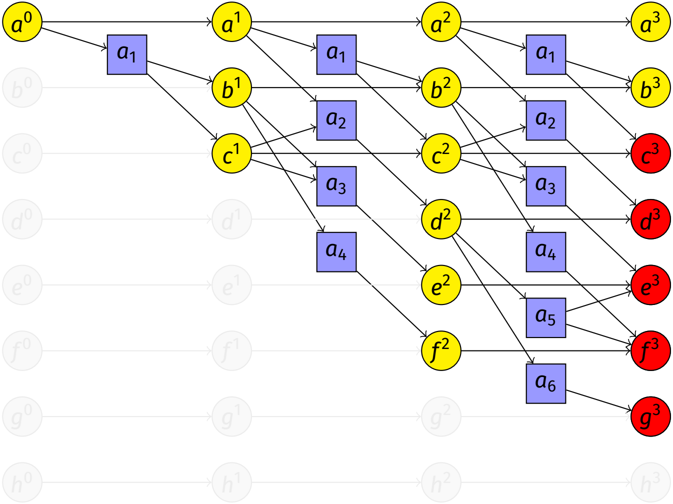
Illustrative Example: Relaxed Planning Graph



$a_1 = \{a\} \xrightarrow{3} \{b, c\}$ $a_2 = \{a, c\} \xrightarrow{1} \{d\}$ $a_3 = \{b, c\} \xrightarrow{1} \{e\}$

$a_4 = \{b\} \xrightarrow{1} \{f\}$ $a_5 = \{d\} \xrightarrow{1} \{e, f\}$ $a_6 = \{d\} \xrightarrow{1} \{g\}$

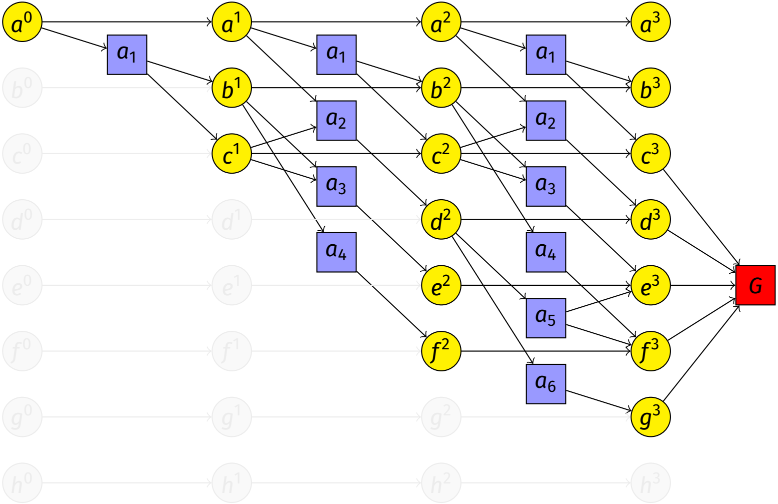
Illustrative Example: Relaxed Planning Graph



$a_1 = \{a\} \xrightarrow{3} \{b, c\}$ $a_2 = \{a, c\} \xrightarrow{1} \{d\}$ $a_3 = \{b, c\} \xrightarrow{1} \{e\}$

$a_4 = \{b\} \xrightarrow{1} \{f\}$ $a_5 = \{d\} \xrightarrow{1} \{e, f\}$ $a_6 = \{d\} \xrightarrow{1} \{g\}$

Illustrative Example: Relaxed Planning Graph



$a_1 = \{a\} \xrightarrow{3} \{b, c\}$ $a_2 = \{a, c\} \xrightarrow{1} \{d\}$ $a_3 = \{b, c\} \xrightarrow{1} \{e\}$

$a_4 = \{b\} \xrightarrow{1} \{f\}$ $a_5 = \{d\} \xrightarrow{1} \{e, f\}$ $a_6 = \{d\} \xrightarrow{1} \{g\}$

Concrete Examples for Generic RPG Heuristic

many planning heuristics are derived from the RPG

in this course:

- maximum heuristic $h^{\max}$ (Bonet & Geffner, 1999)
- additive heuristic h^{add} (Bonet, Loerincs & Geffner, 1997)
- Keyder & Geffner's (2008) variant of the FF heuristic h^{FF} (Hoffmann & Nebel, 2001)

remark:

- the most efficient implementations of these heuristics do not use explicit planning graphs, but rather alternative (equivalent) definitions

Maximum and Additive Heuristics

Maximum and Additive Heuristics

- $h^{\max}$ and h^{add} are the simplest RPG heuristics
- annotate vertices with numerical values
- the vertex values estimate the costs
  - to make a given variable true
  - to reach and apply a given action
  - to reach the goal

Maximum and Additive Heuristics: Heuristic Computation

computation of annotations:

- costs of variable vertices:
  0 in layer 0;
  otherwise minimum of the costs of predecessor vertices
- costs of action and goal vertices:
  maximum ($h^{\max}$) or sum (h^{add}) of predecessor vertex costs;
  for action vertices a^i, also add $cost(a)$

termination criterion:

- stability: terminate if $V^i = V^{i-1}$ and costs of all vertices in V^i equal corresponding vertex costs in V^{i-1}

heuristic value:

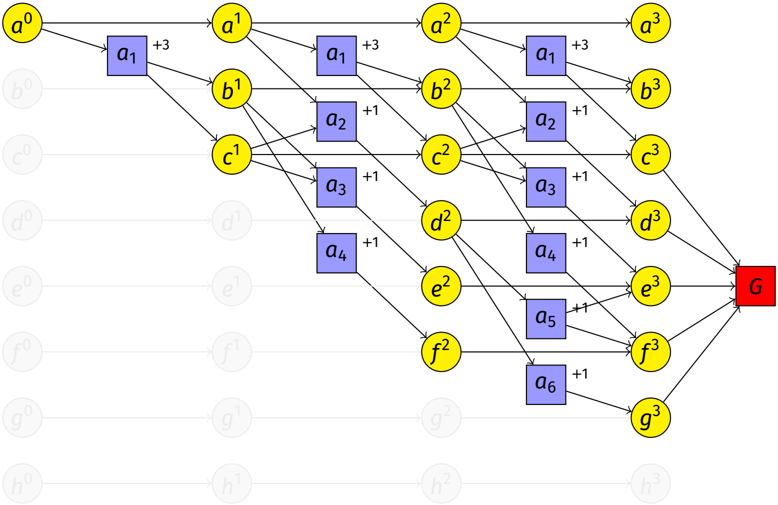
- value of goal vertex in the last layer

Maximum and Additive Heuristics: Intuition

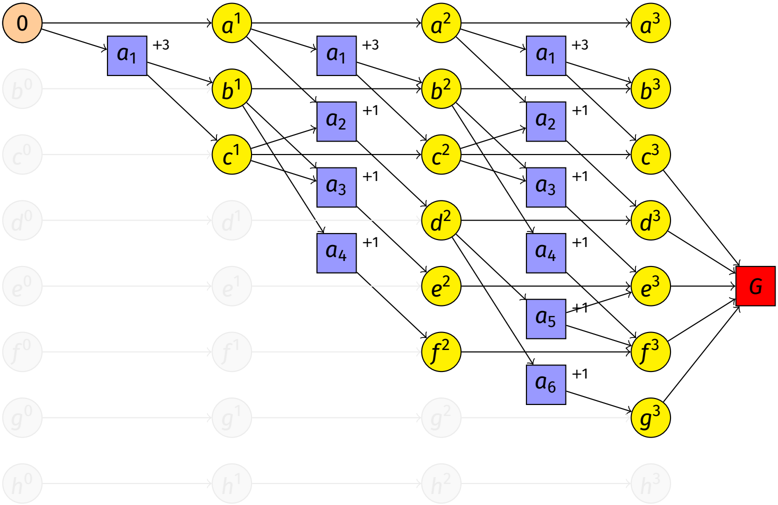
intuition:

- variable vertices:
  - choose cheapest way of reaching the variable
- action/goal vertices:
  - $h^{\max}$ makes optimistic assumptions:
    when reaching the most expensive precondition variable, we can reach the other precondition variables in parallel (hence maximization of costs)
  - h^{add} makes pessimistic assumptions:
    all precondition variables must be reached completely independently of each other (hence summation of costs)

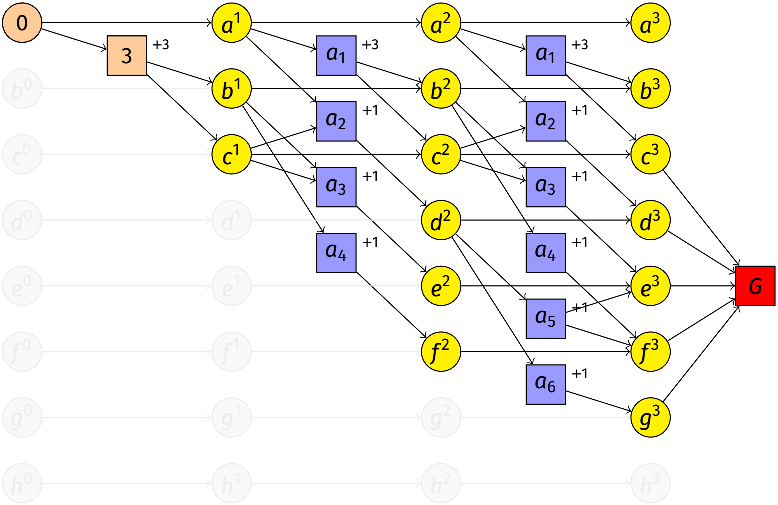
Illustrative Example: $h^{\max}$



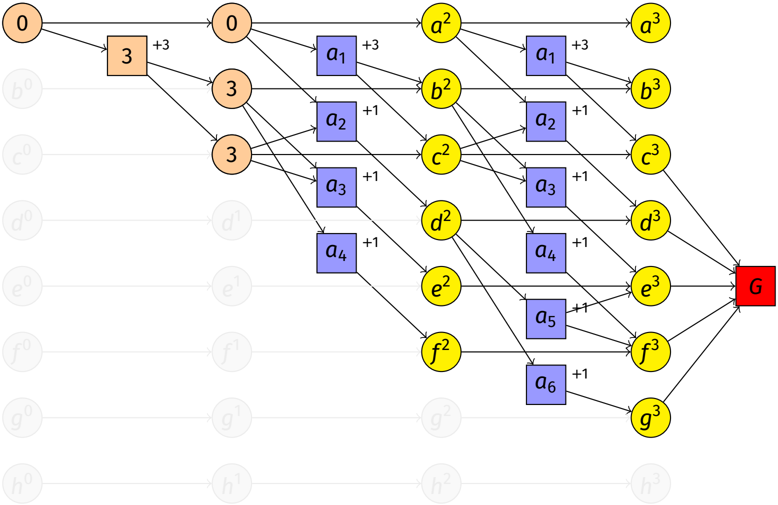
Illustrative Example: $h^{\max}$



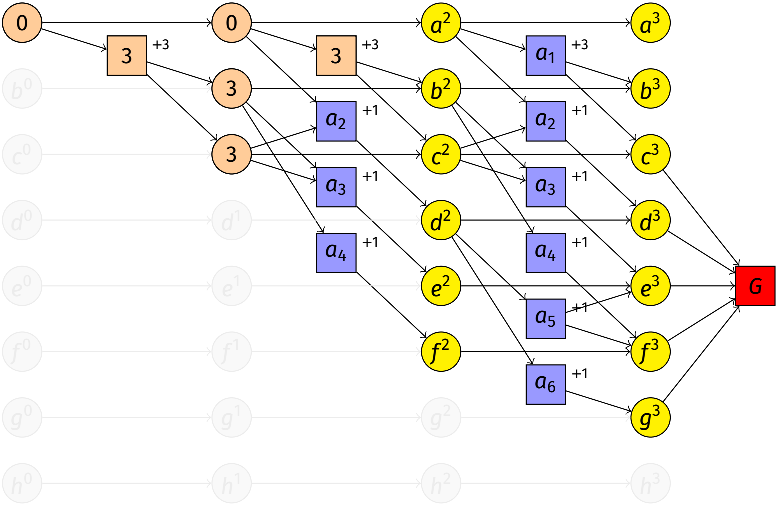
Illustrative Example: $h^{\max}$



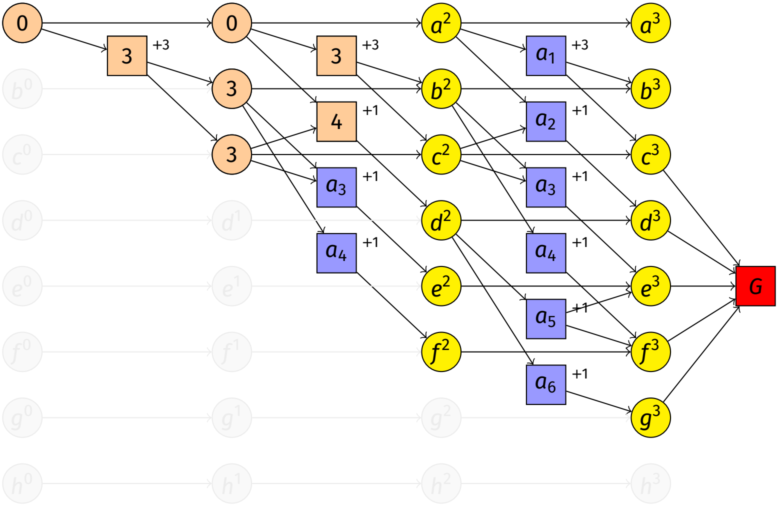
Illustrative Example: $h^{\max}$



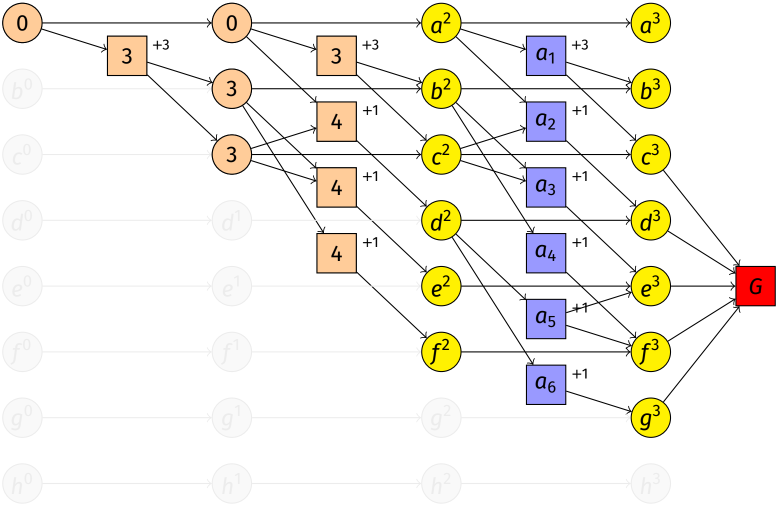
Illustrative Example: $h^{\max}$



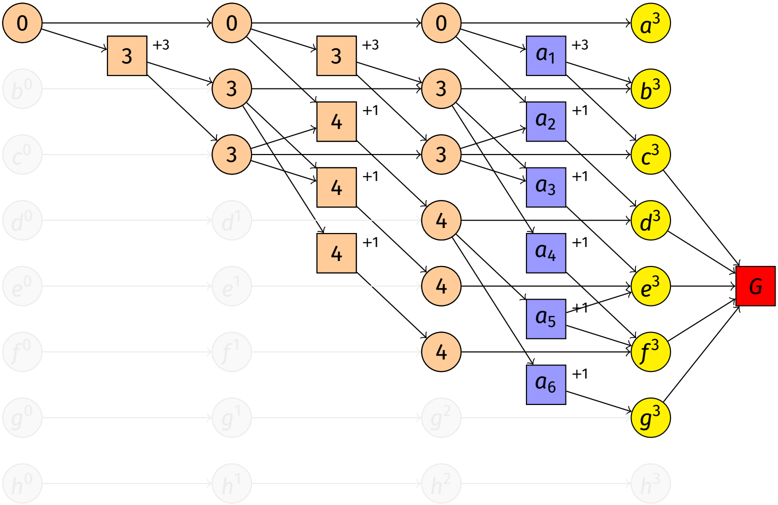
Illustrative Example: $h^{\max}$



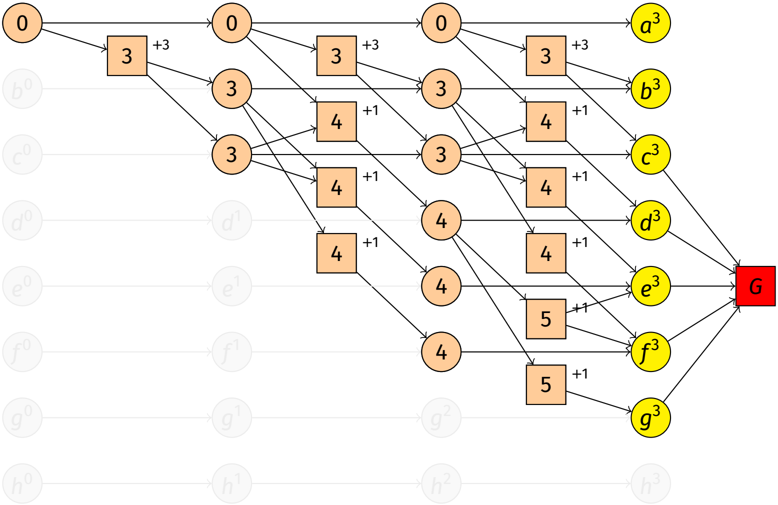
Illustrative Example: $h^{\max}$



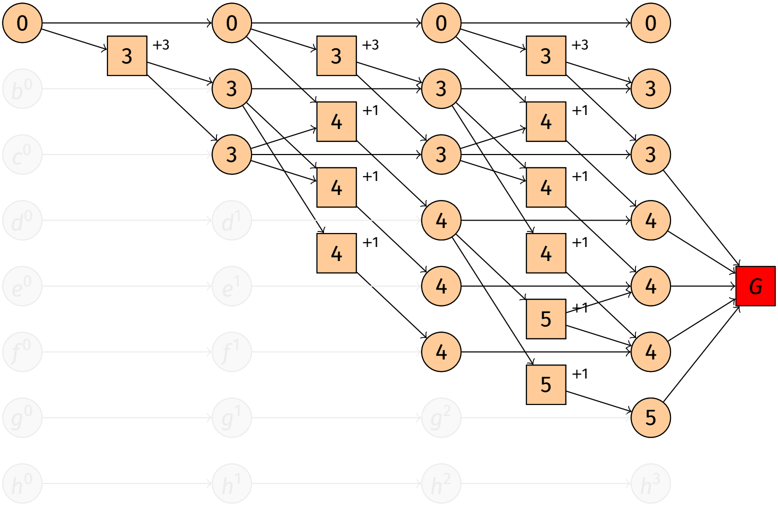
Illustrative Example: $h^{\max}$



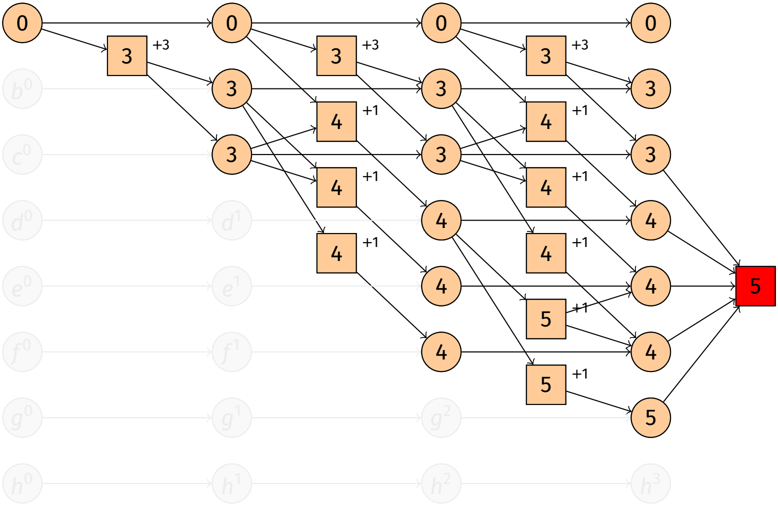
Illustrative Example: $h^{\max}$



Illustrative Example: $h^{\max}$

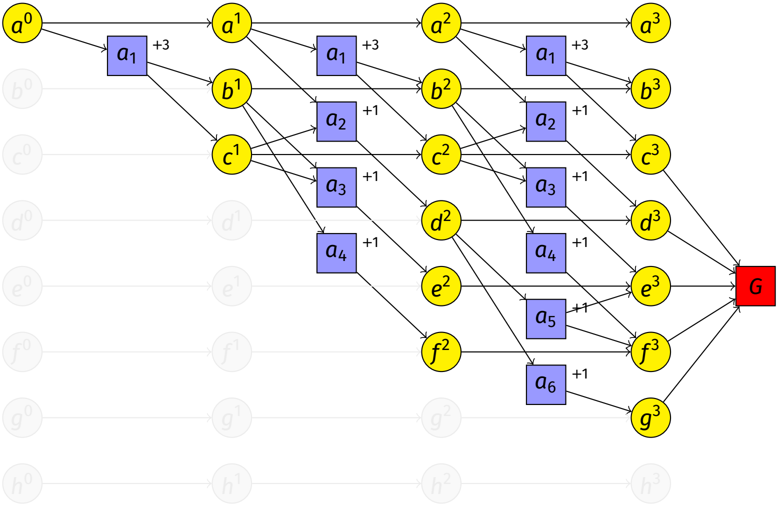


Illustrative Example: $h^{\max}$

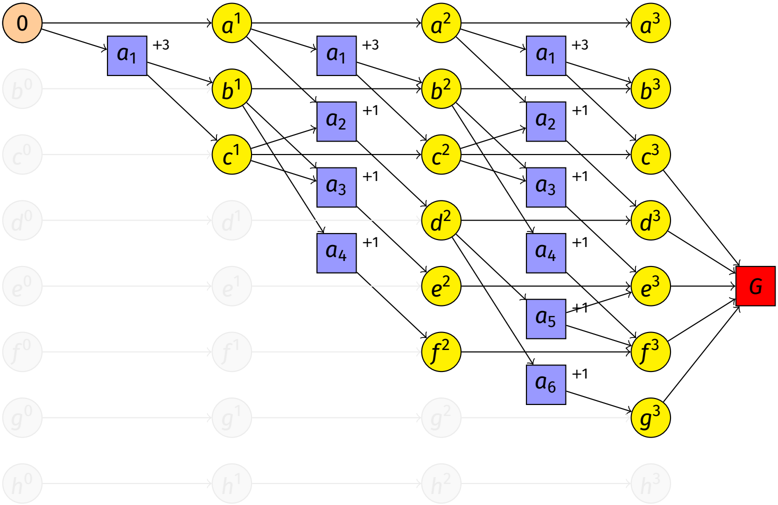


$$h^{\max}(\{a\}) = 5$$

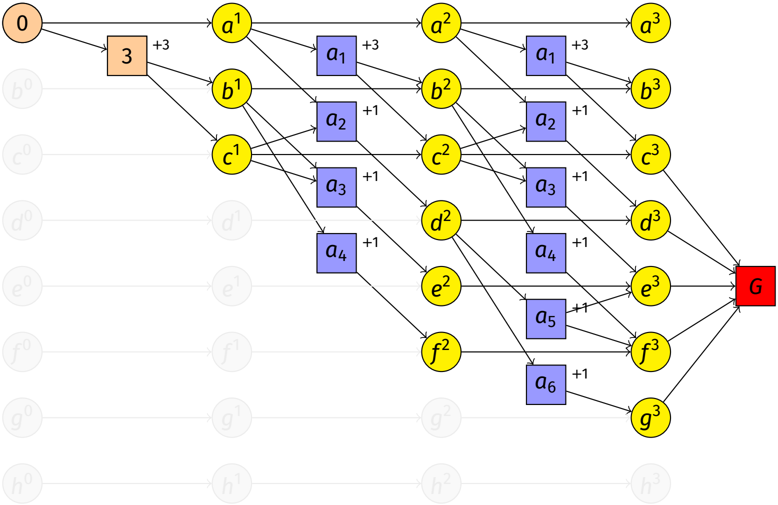
Illustrative Example: h^{add}



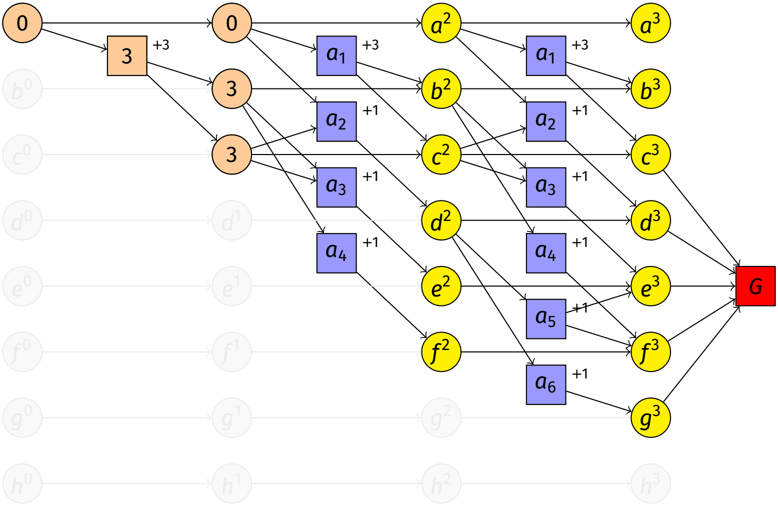
Illustrative Example: h^{add}



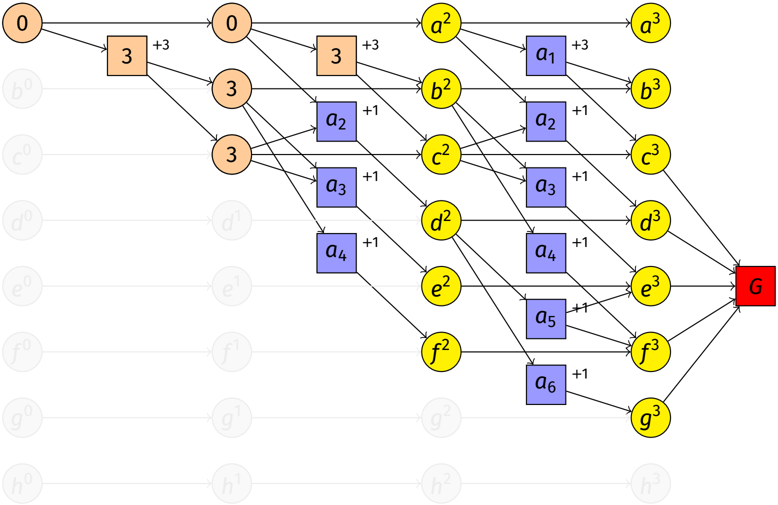
Illustrative Example: h^{add}



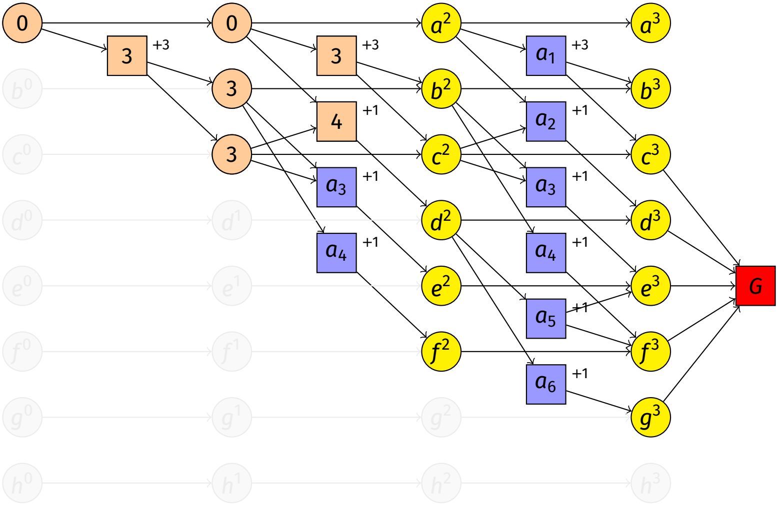
Illustrative Example: h^{add}



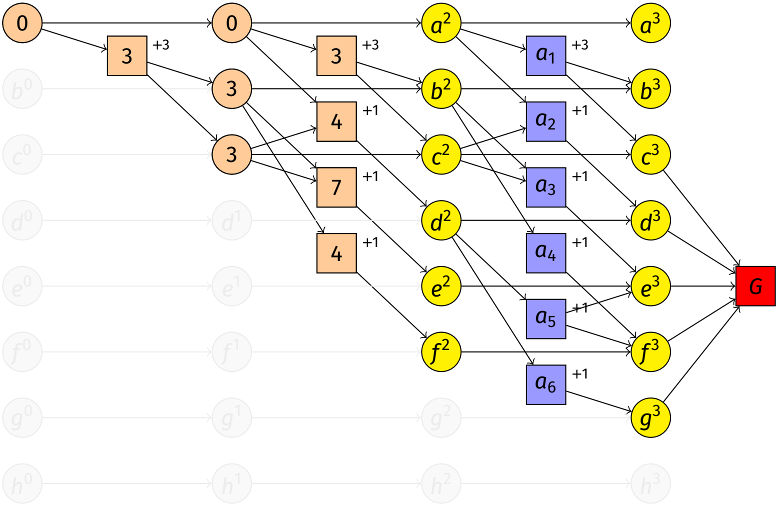
Illustrative Example: h^{add}



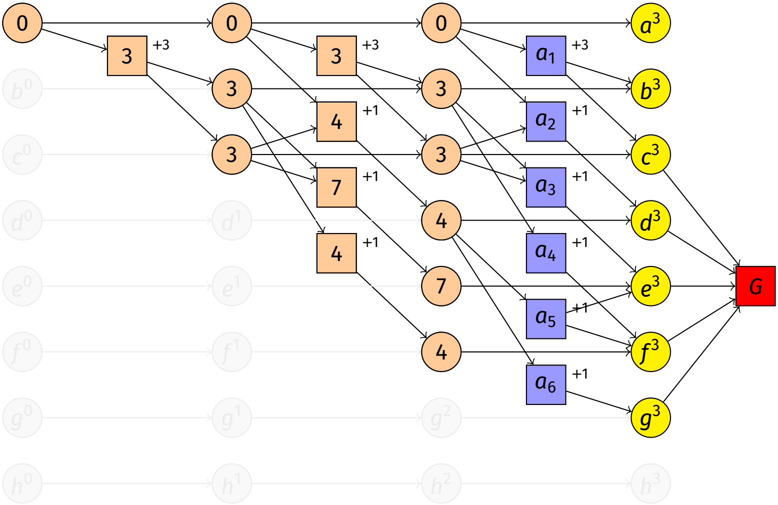
Illustrative Example: h^{add}



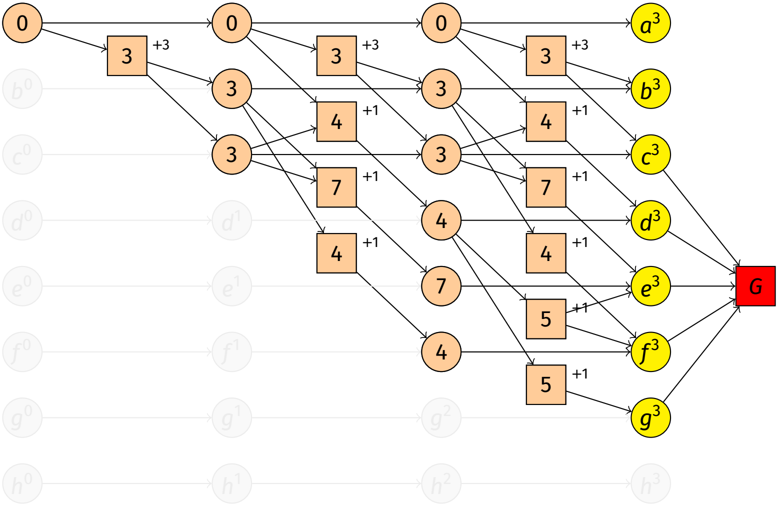
Illustrative Example: h^{add}



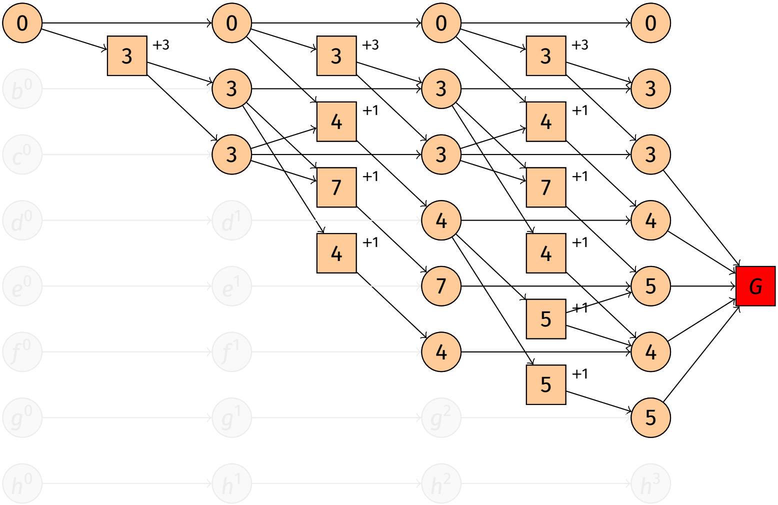
Illustrative Example: h^{add}



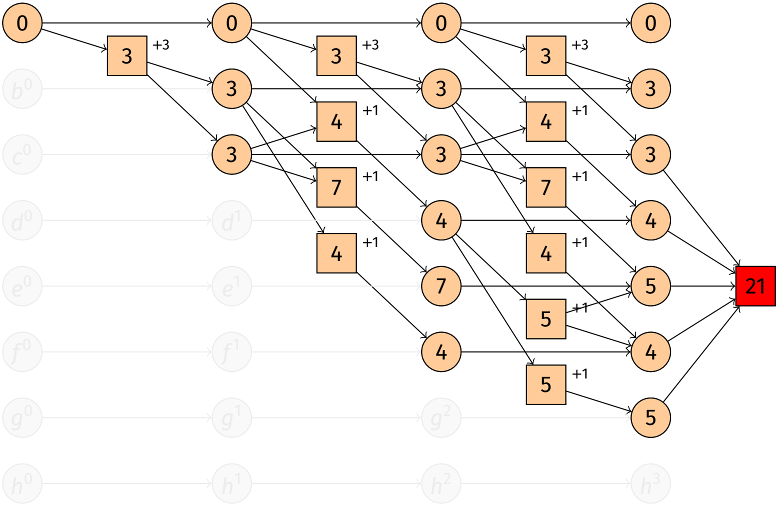
Illustrative Example: h^{add}



Illustrative Example: h^{add}



Illustrative Example: h^{add}



$$h^{\text{add}}(\{a\}) = 21$$

h^{max} and h^{add}: Remarks

comparison of h^{max} and h^{add}:

- both are safe and goal-aware
- h^{max} is admissible and consistent; h^{add} is neither.

$\rightsquigarrow$ h^{add} not suited for optimal planning

$h^{\max}$ and h^{add}: Remarks

comparison of $h^{\max}$ and h^{add}:

- both are safe and goal-aware
- $h^{\max}$ is admissible and consistent; h^{add} is neither.

$\rightsquigarrow$ h^{add} not suited for optimal planning

- however, h^{add} is usually much more informative than $h^{\max}$
  greedy best-first search with h^{add} is a decent algorithm

$h^{\max}$ and h^{add}: Remarks

comparison of $h^{\max}$ and h^{add}:

- both are safe and goal-aware
- $h^{\max}$ is admissible and consistent; h^{add} is neither.

$\rightsquigarrow$ h^{add} not suited for optimal planning

- however, h^{add} is usually much more informative than $h^{\max}$
  greedy best-first search with h^{add} is a decent algorithm
- apart from not being admissible, h^{add} often vastly overestimates the actual costs because
  positive synergies between subgoals are not recognized

$h^{\max}$ and h^{add}: Remarks

comparison of $h^{\max}$ and h^{add}:

- both are safe and goal-aware
- $h^{\max}$ is admissible and consistent; h^{add} is neither.

⇝ h^{add} not suited for optimal planning

- however, h^{add} is usually much more informative than $h^{\max}$
  greedy best-first search with h^{add} is a decent algorithm
- apart from not being admissible, h^{add} often vastly overestimates the actual costs because
  positive synergies between subgoals are not recognized

⇝ FF heuristic

FF Heuristic

FF Heuristic

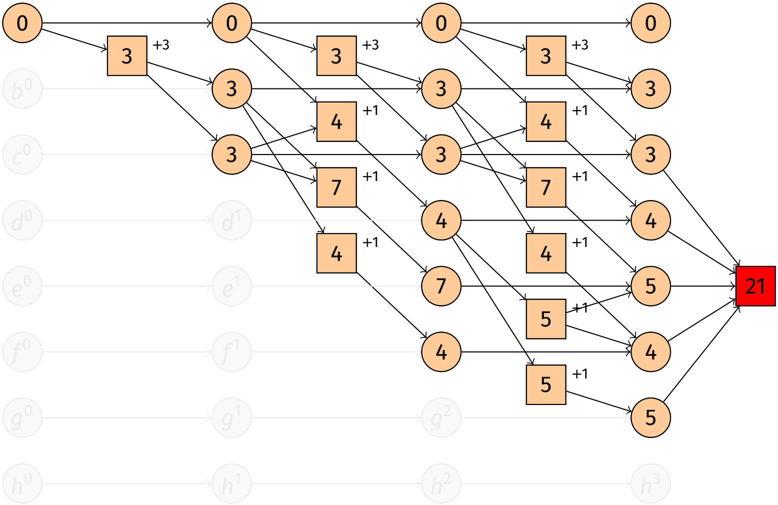
identical to h^{add}, but additional steps at the end:

- mark goal vertex in the last graph layer
- apply the following marking rules until nothing more to do:
  - marked action or goal vertex?
    $\rightsquigarrow$ mark all predecessors
  - marked variable vertex v^i in layer $i \geq 1$?
    $\rightsquigarrow$ mark one predecessor with minimal h^{add} value
    (tie-breaking: prefer variable vertices; otherwise arbitrary)

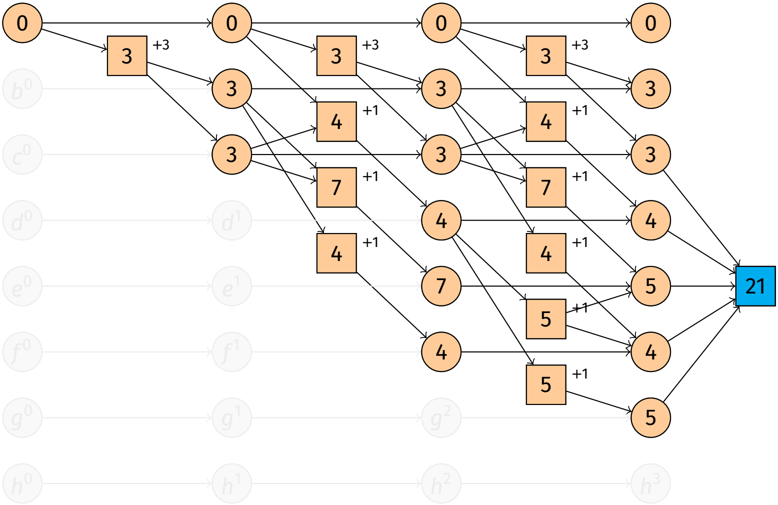
heuristic value:

- the actions corresponding to the marked action vertices build a relaxed plan
- the cost of this plan is the heuristic value

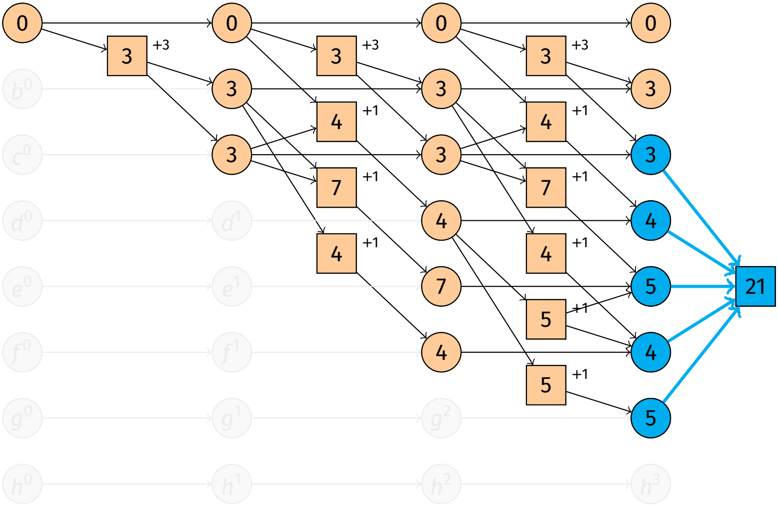
Illustrative Example: h^{FF}



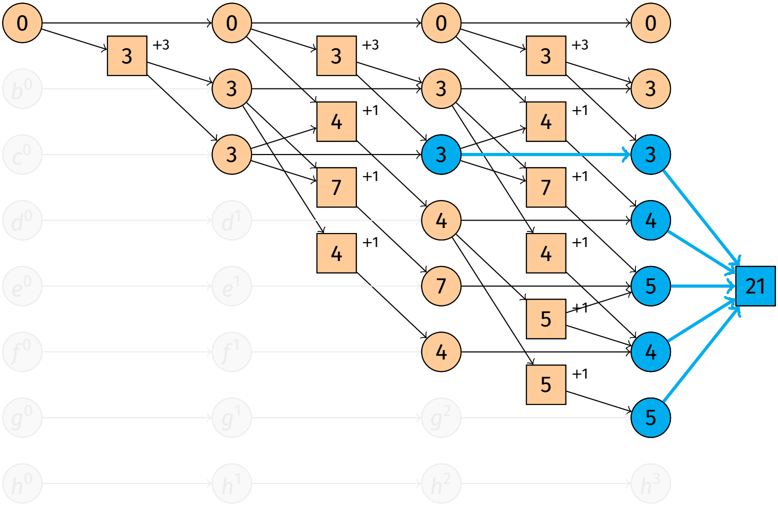
Illustrative Example: h^{FF}



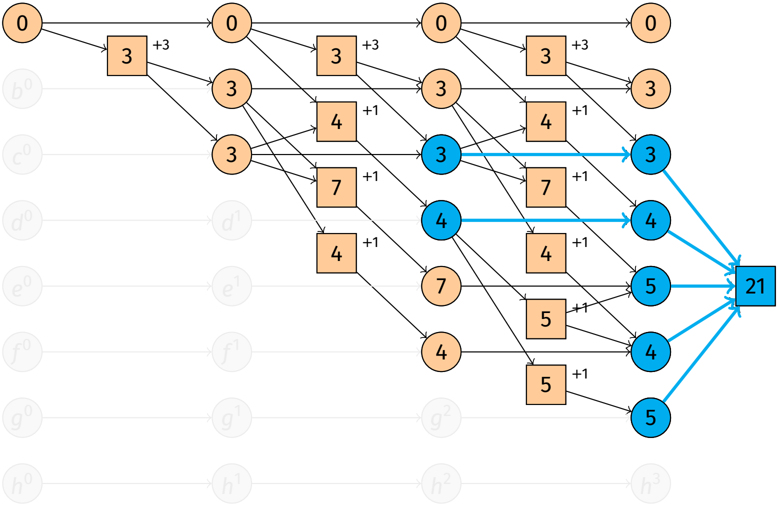
Illustrative Example: h^{FF}



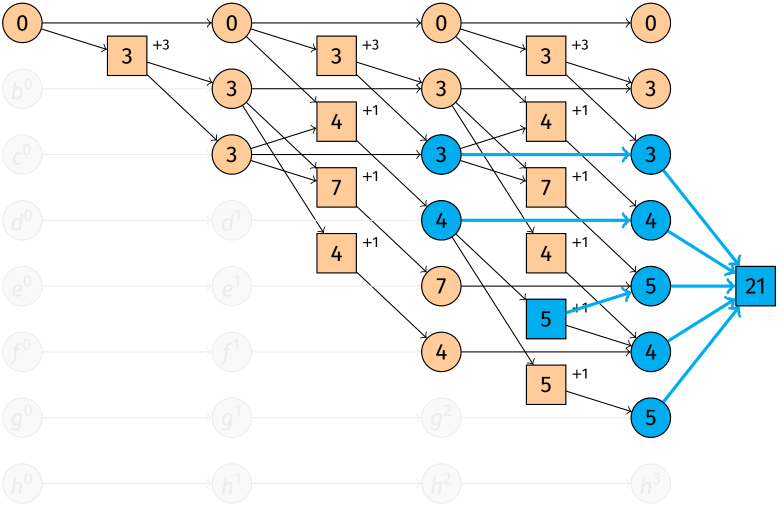
Illustrative Example: h^{FF}



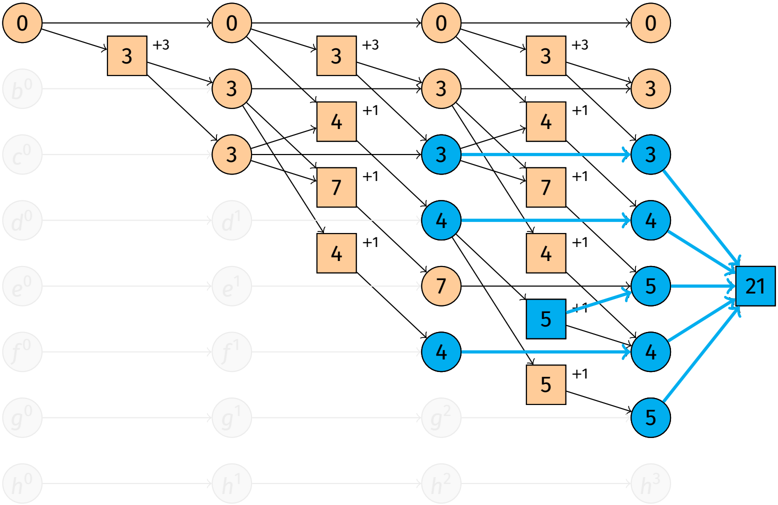
Illustrative Example: h^{FF}



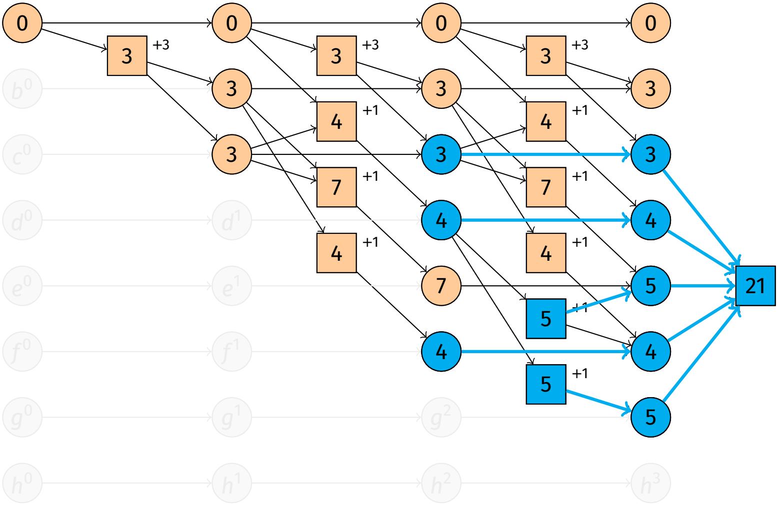
Illustrative Example: h^{FF}



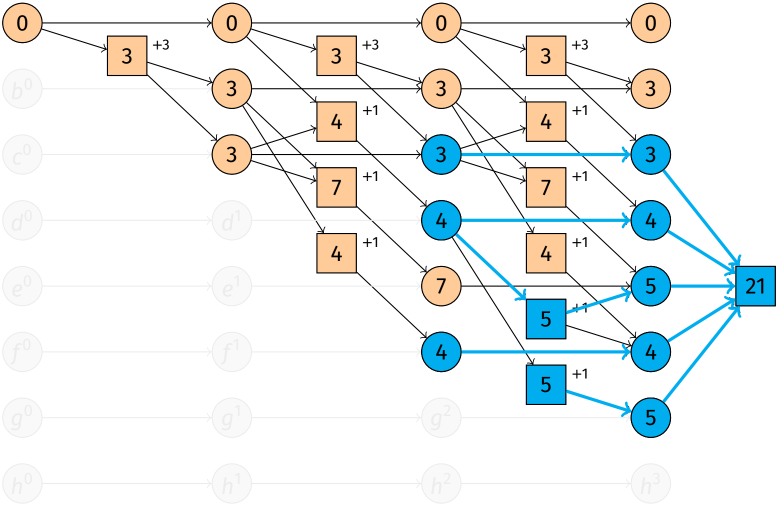
Illustrative Example: h^{FF}



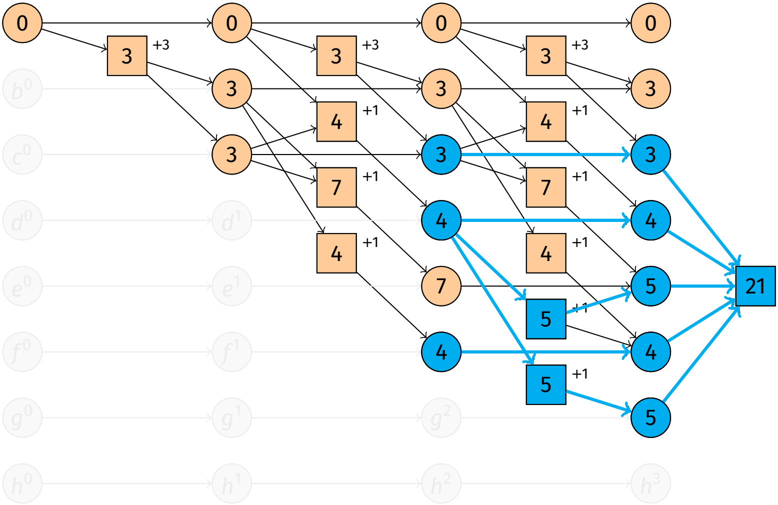
Illustrative Example: h^{FF}



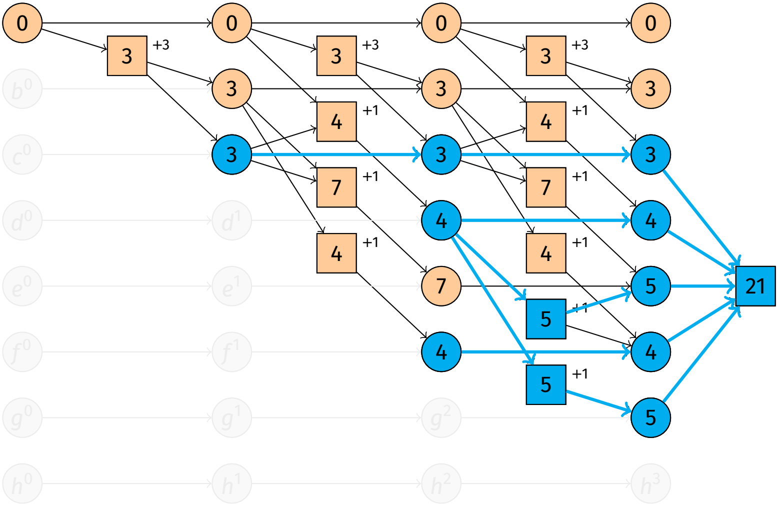
Illustrative Example: h^{FF}



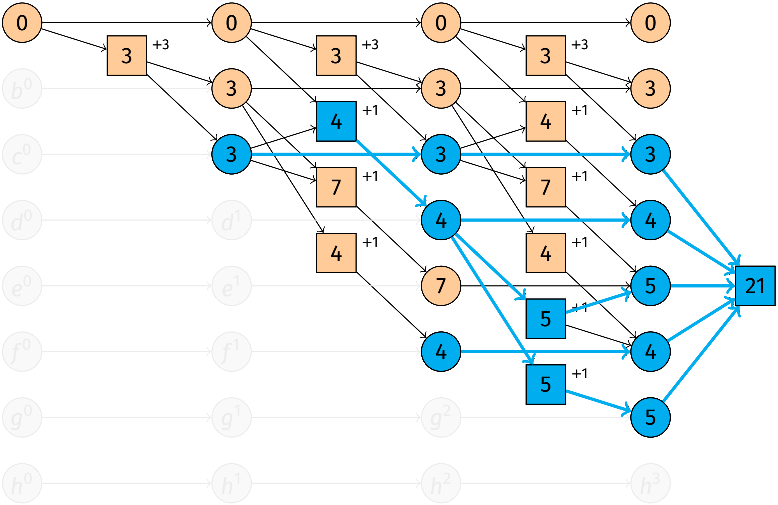
Illustrative Example: h^{FF}



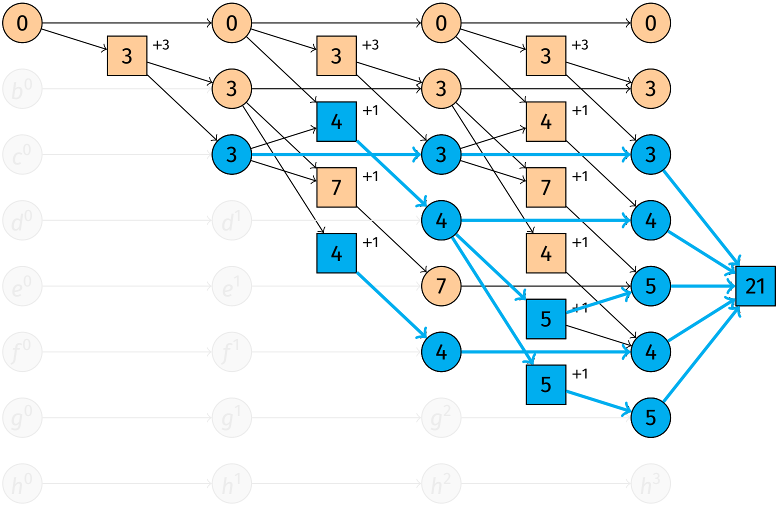
Illustrative Example: h^{FF}



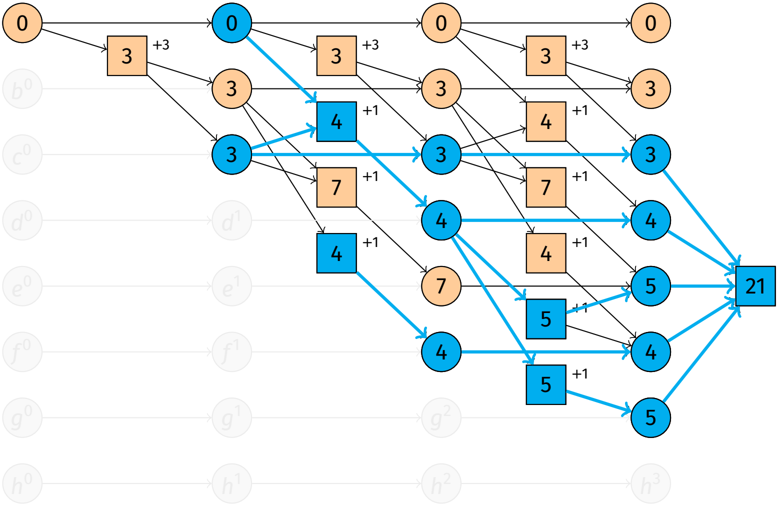
Illustrative Example: h^{FF}



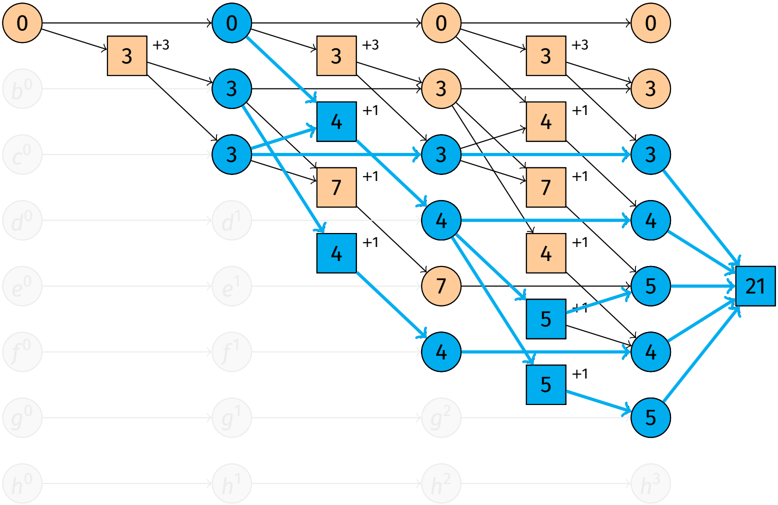
Illustrative Example: h^{FF}



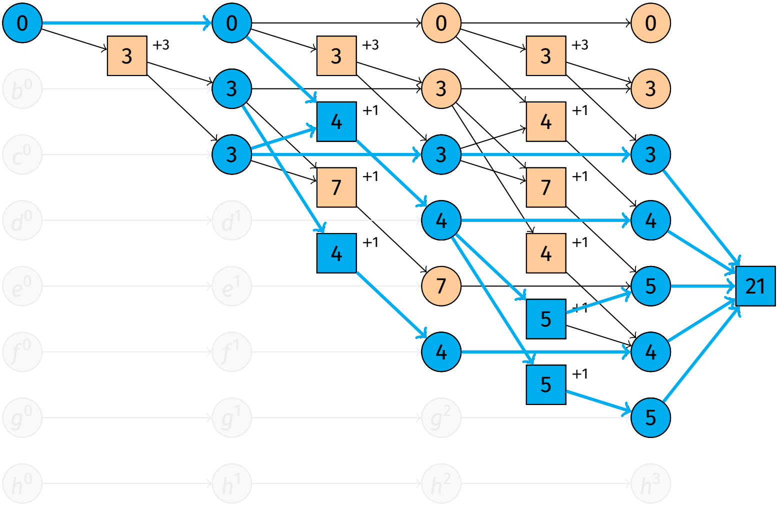
Illustrative Example: h^{FF}



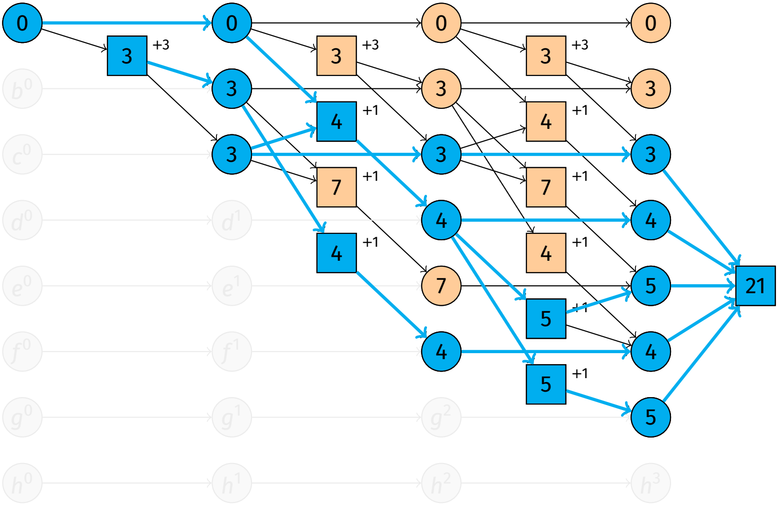
Illustrative Example: h^{FF}



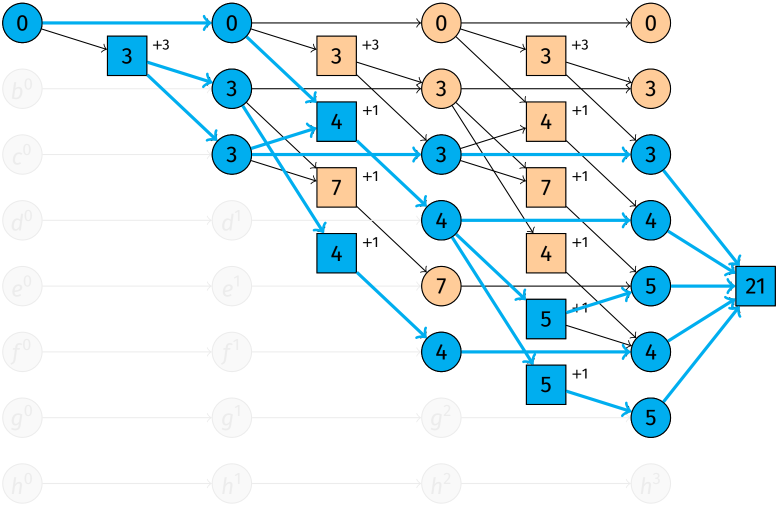
Illustrative Example: h^{FF}



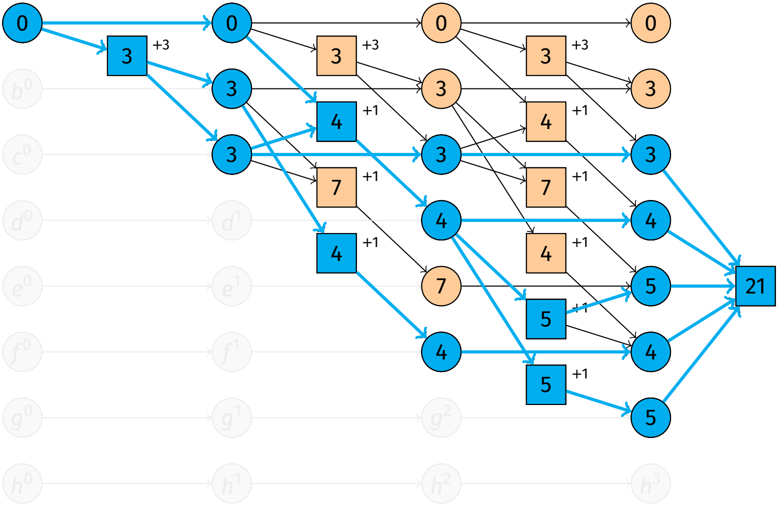
Illustrative Example: h^{FF}



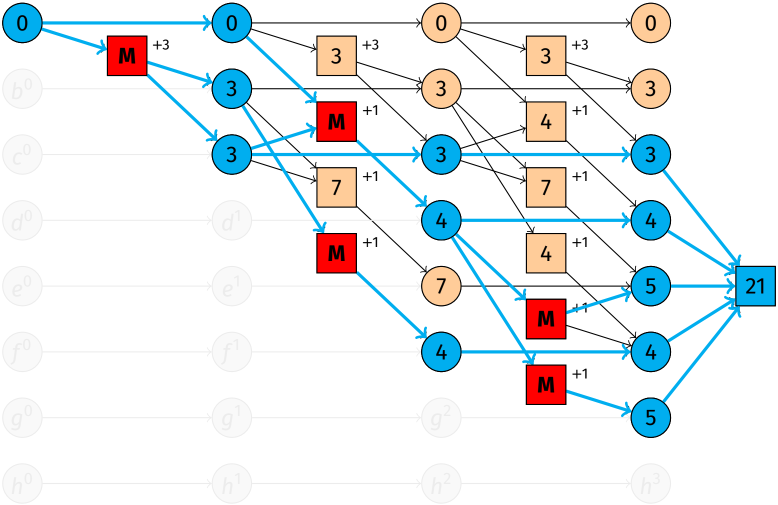
Illustrative Example: h^{FF}



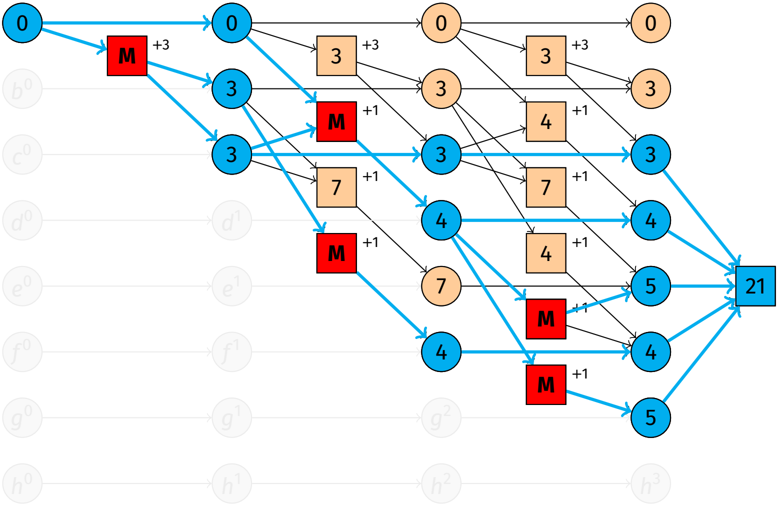
Illustrative Example: h^{FF}



Illustrative Example: h^{FF}



Illustrative Example: h^{FF}



$$h^{\text{FF}}(\{a\}) = 3 + 1 + 1 + 1 + 1 = 7$$

FF Heuristic: Remarks

- like h^{add}, h^{FF} is safe and goal-aware, but neither admissible nor consistent
- approximation of h^+ which is always at least as good as h^{add}
- usually significantly better
- can be computed in almost linear time ($O(n \log n)$) in the size of the description of the planning task

FF Heuristic: Remarks

- like h^{add}, h^{FF} is safe and goal-aware, but neither admissible nor consistent
- approximation of h^{+} which is always at least as good as h^{add}
- usually significantly better
- can be computed in almost linear time ($O(n \log n)$) in the size of the description of the planning task
- computation of heuristic value depends on tie-breaking of marking rules (h^{FF} not well-defined)
- one of the most successful planning heuristics

Comparison of Relaxation Heuristics

Relationships of Relaxation Heuristics

Let s be a state in the STRIPS planning task $\langle V, I, G, A \rangle$.

Then

- $h^{\max}(s) \leq h^{+}(s) \leq h^{*}(s)$
- $h^{\max}(s) \leq h^{+}(s) \leq h^{\mathrm{FF}}(s) \leq h^{\mathrm{add}}(s)$
- h^{*} and h^{FF} are incomparable
- h^{*} and h^{add} are incomparable

further remarks:

- for non-admissible heuristics, it is generally neither good nor bad to compute higher values than another heuristic
- for relaxation heuristics, the objective is to approximate h^{+} as closely as possible

Quiz

Kahoot!