Automated Planning

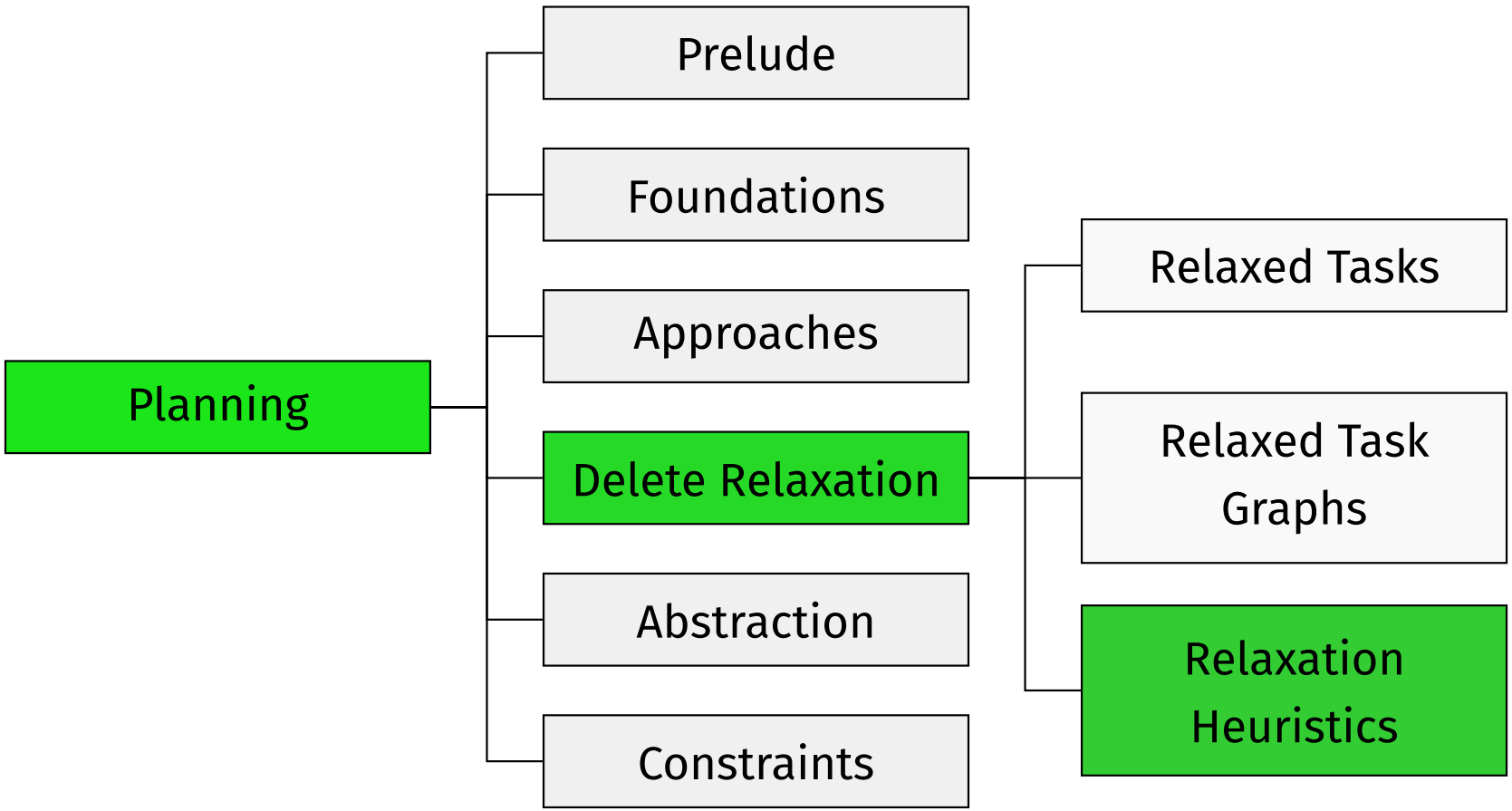
D6. Delete Relaxation: h^{FF} and Comparison of Heuristics

Jendrik Seipp

Linköping University

based on slides from the AI group at the University of Basel

Content of this Course



The FF Heuristic

Inaccuracies in h^{max} and h^{add}

- h^{max} is often inaccurate because it undercounts:
  the heuristic estimate only reflects the cost of a critical path,
  which is often only a small fraction of the overall plan.
- h^{add} is often inaccurate because it overcounts:
  if the same subproblem is reached in many ways, it will be counted
  many times although it only needs to be solved once.

The FF Heuristic

With best achiever graphs, there is a simple solution to the overcounting of h^{add}: count all effect nodes that h^{add} would count, but only count each of them once.

Definition (FF Heuristic)

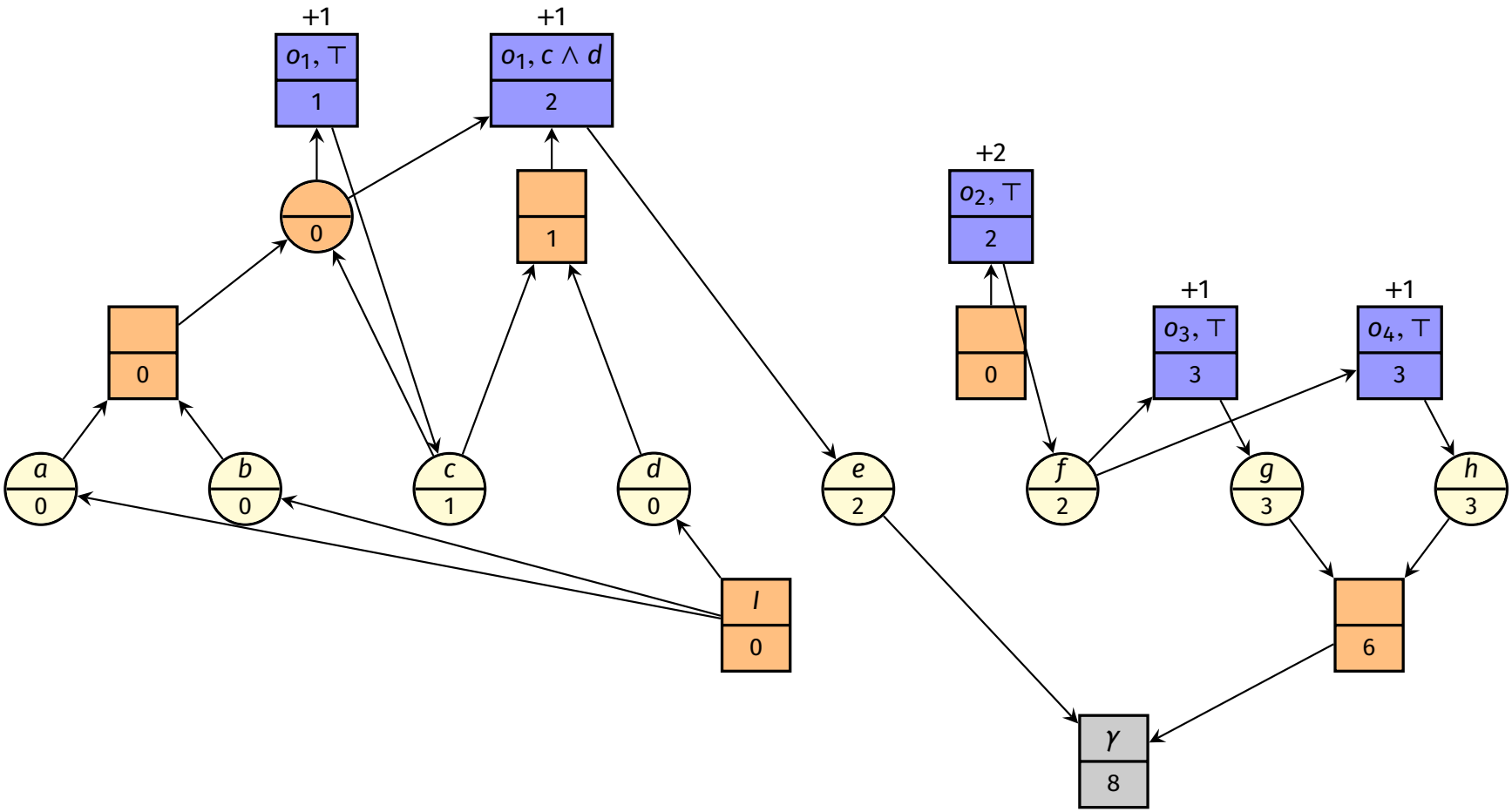
Let $\Pi = \langle V, I, O, \gamma \rangle$ be a propositional planning task in positive normal form. The FF heuristic for a state s of Π, written $h^{\text{FF}}(s)$, is computed as follows:

- Construct the RTG for the task $\langle V, s, O^+, \gamma \rangle$
- Construct the best achiever graph G^{add}.
- Compute the set of effect nodes $\{n_{o_1}^{\chi_1}, \ldots, n_{o_k}^{\chi_k}\}$ from which n_γ in G^{add} is reachable.
- Return $h^{\text{FF}}(s) = \sum_{i=1}^{k} cost(o_i)$.

Note: h^{FF} is not well-defined; different tie-breaking policies for best achievers can lead to different heuristic values

Example: FF Heuristic (1)

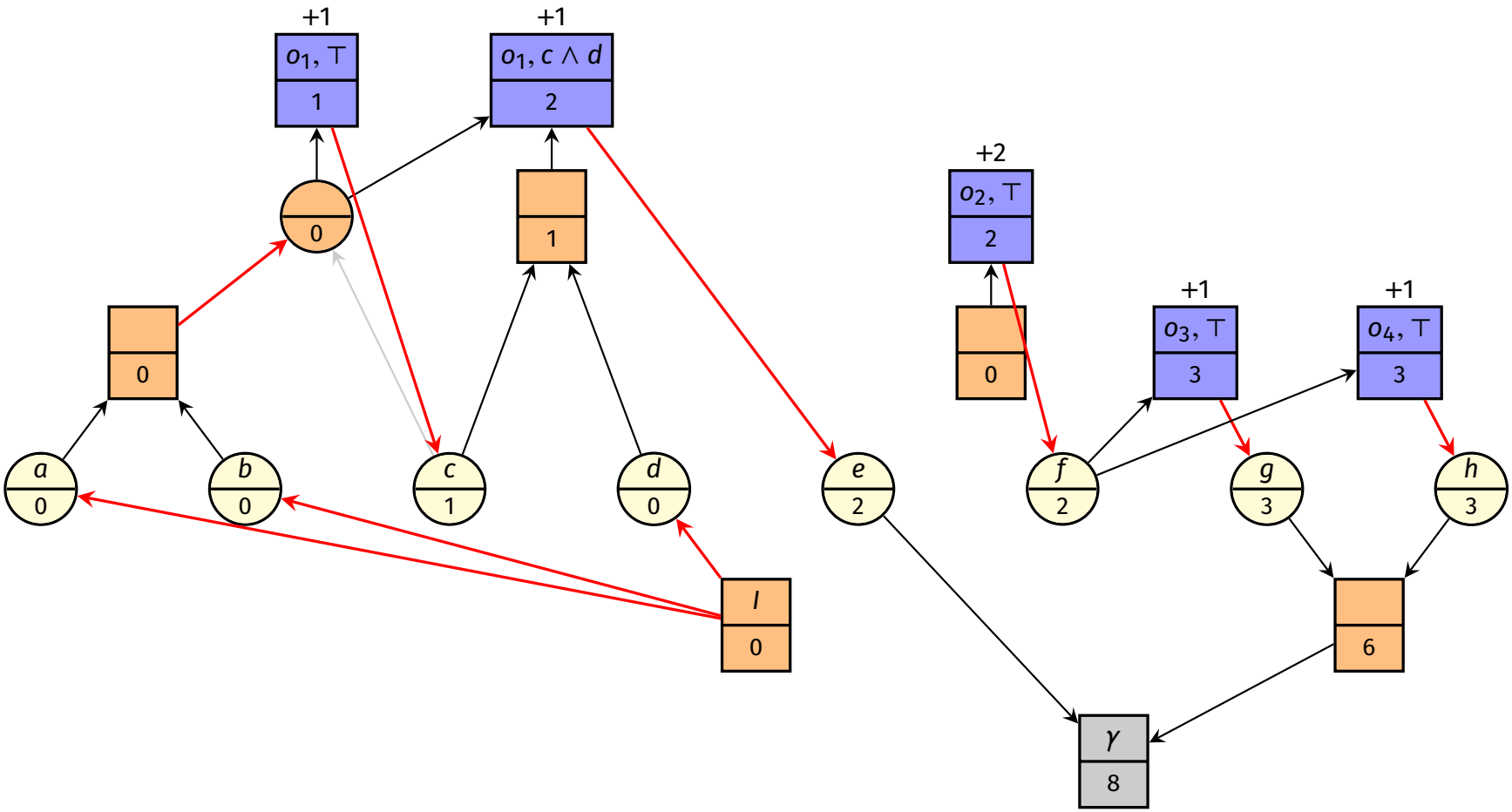
FF heuristic computation



Construct RTG.

Example: FF Heuristic (1)

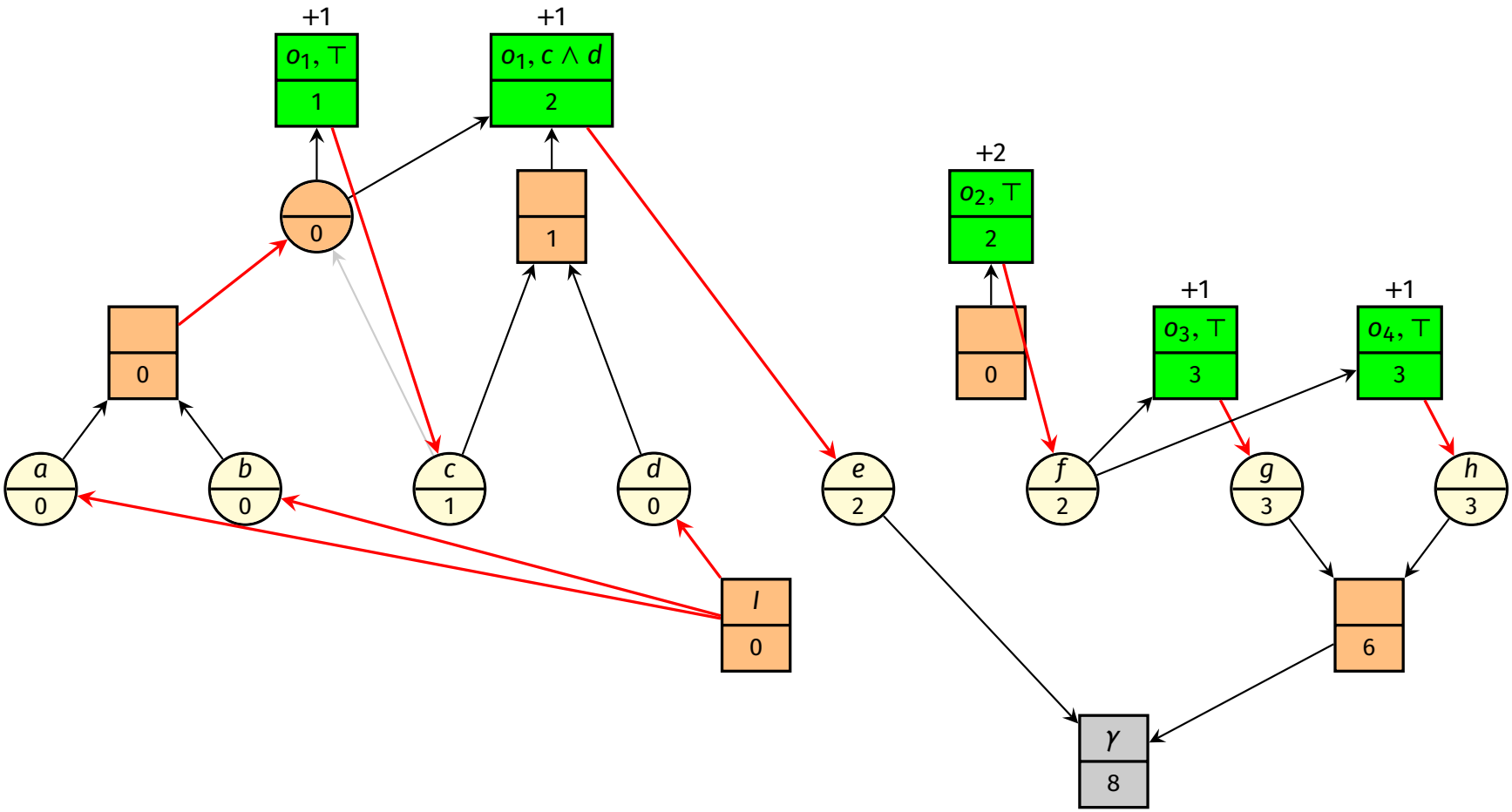
FF heuristic computation



Construct best achiever graph G^{add}.

Example: FF Heuristic (1)

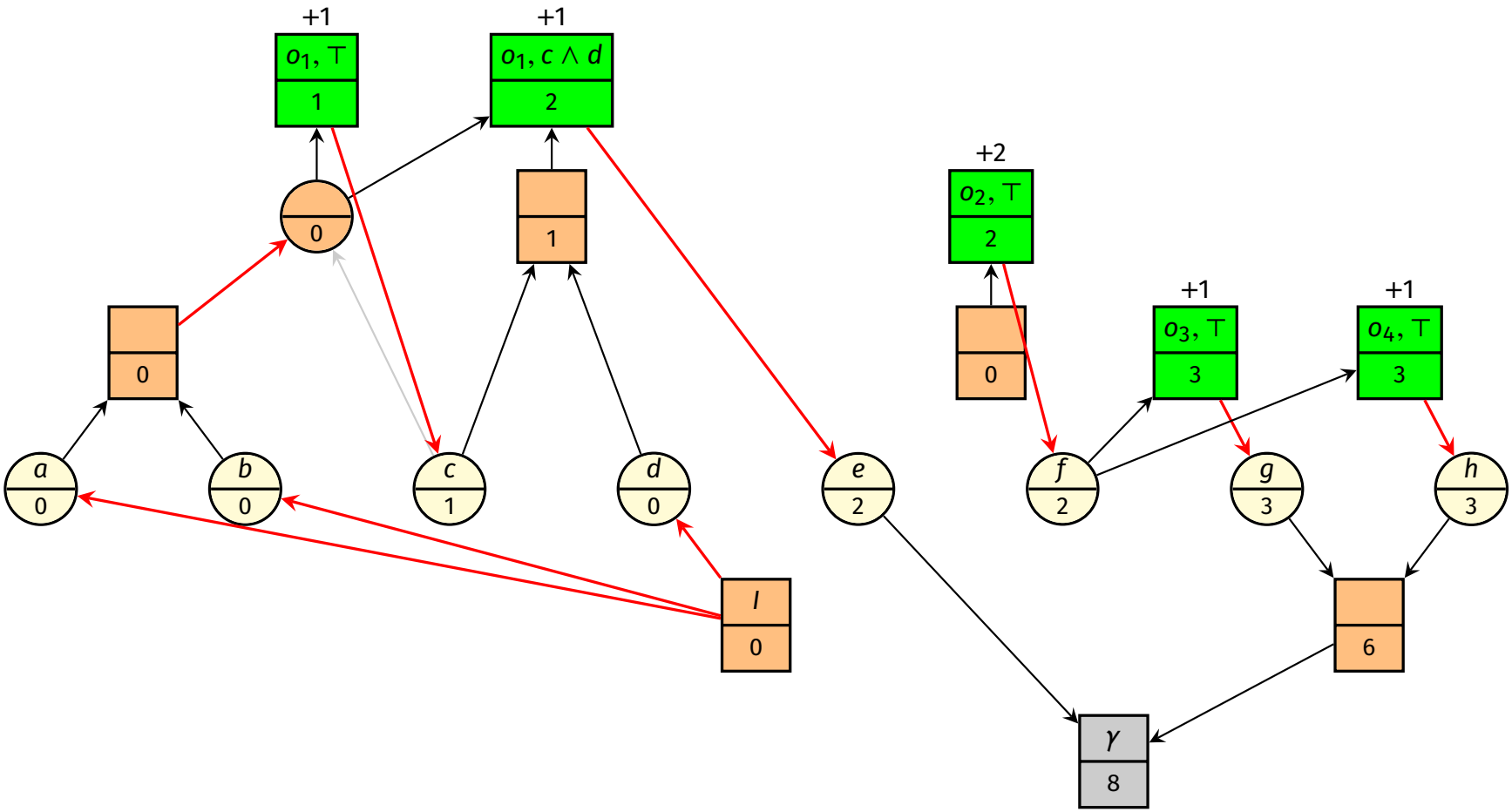
FF heuristic computation



Compute effect nodes from which goal node is reachable.

Example: FF Heuristic (1)

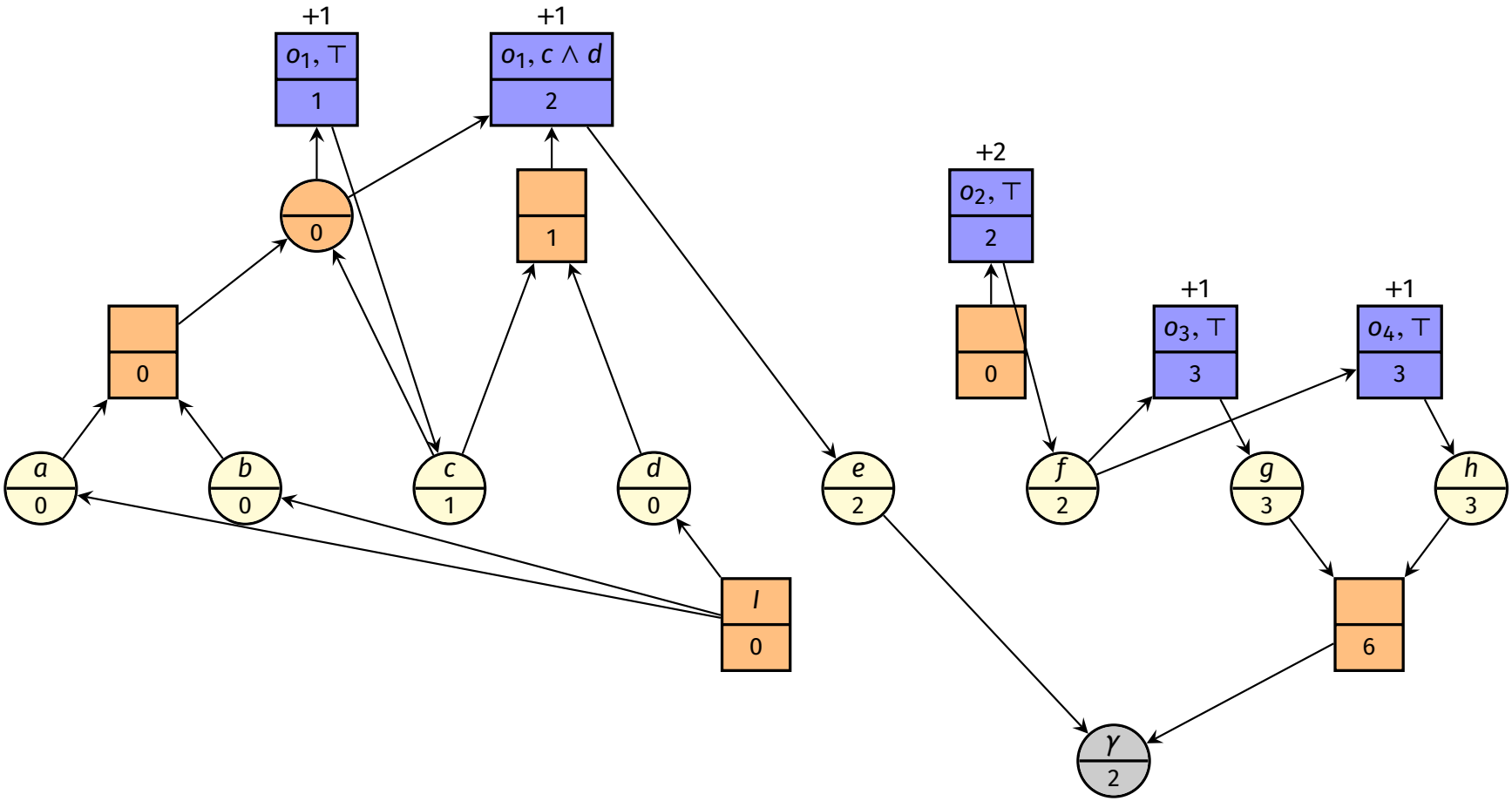
FF heuristic computation



$$h^{FF}(s) = 1 + 1 + 2 + 1 + 1 = 6$$

Example: FF Heuristic (2)

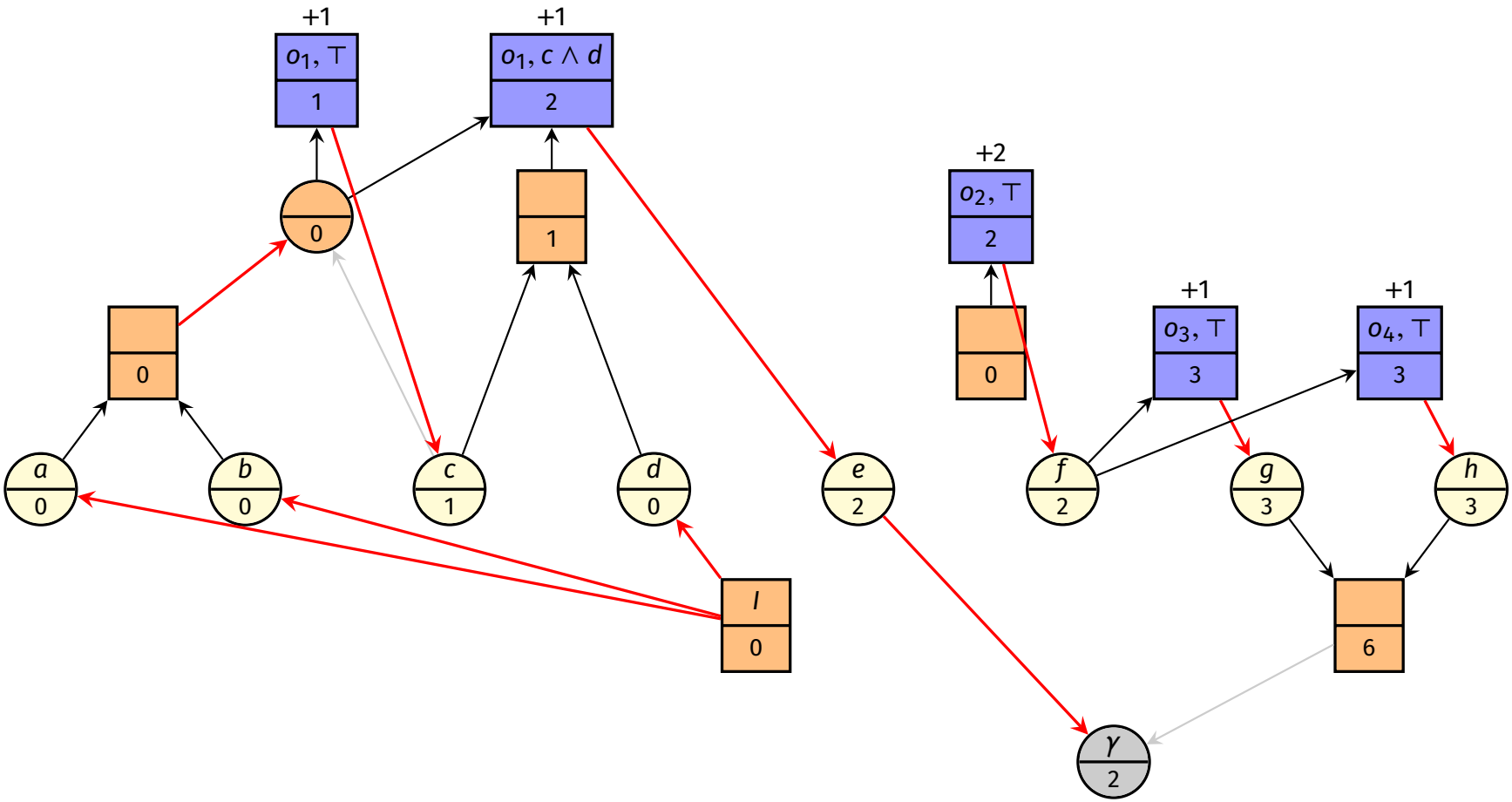
FF heuristic computation; modified goal $e \vee (g \wedge h)$



Construct RTG.

Example: FF Heuristic (2)

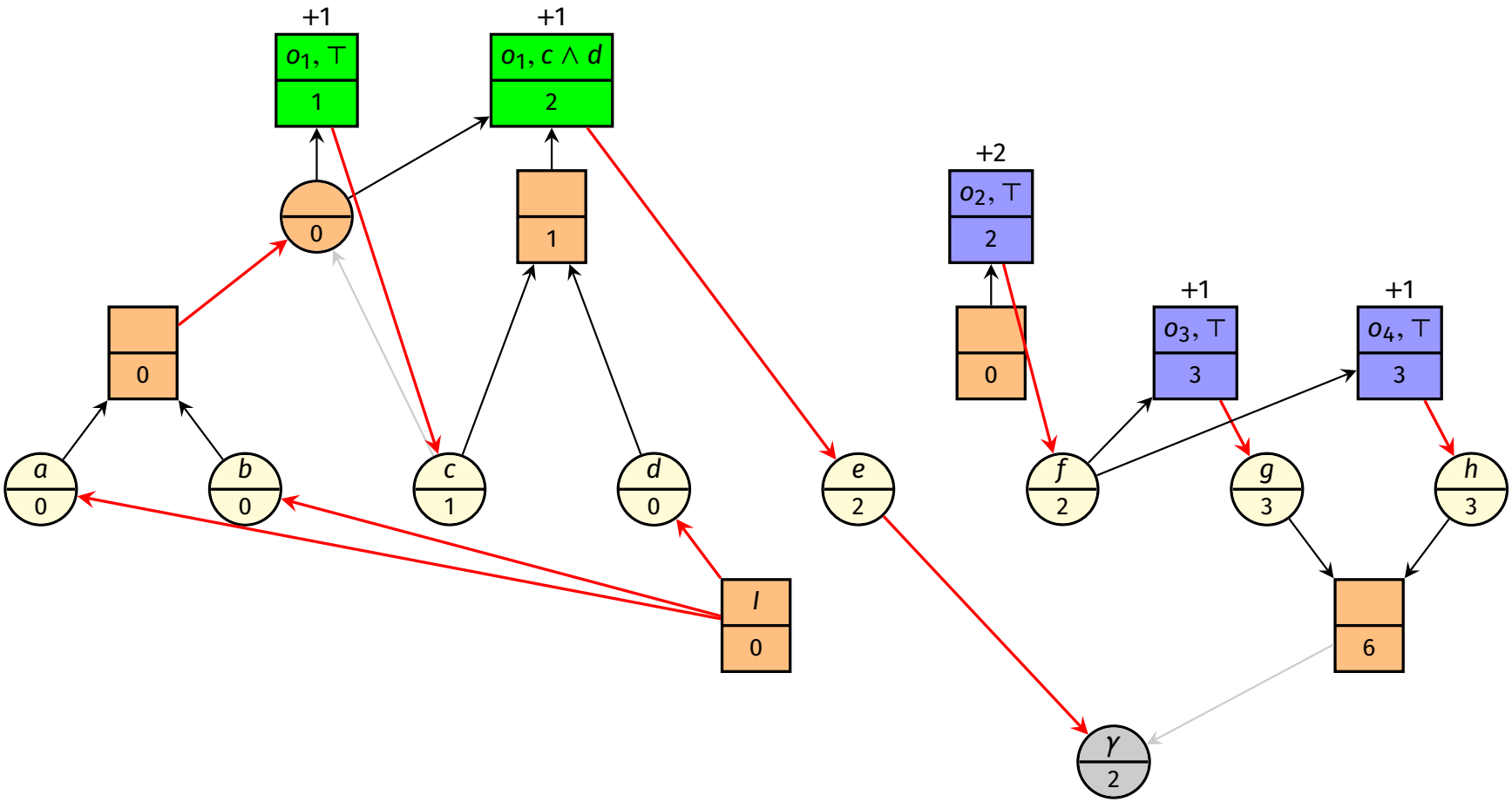
FF heuristic computation; modified goal $e \vee (g \wedge h)$



Construct best achiever graph G^{add}.

Example: FF Heuristic (2)

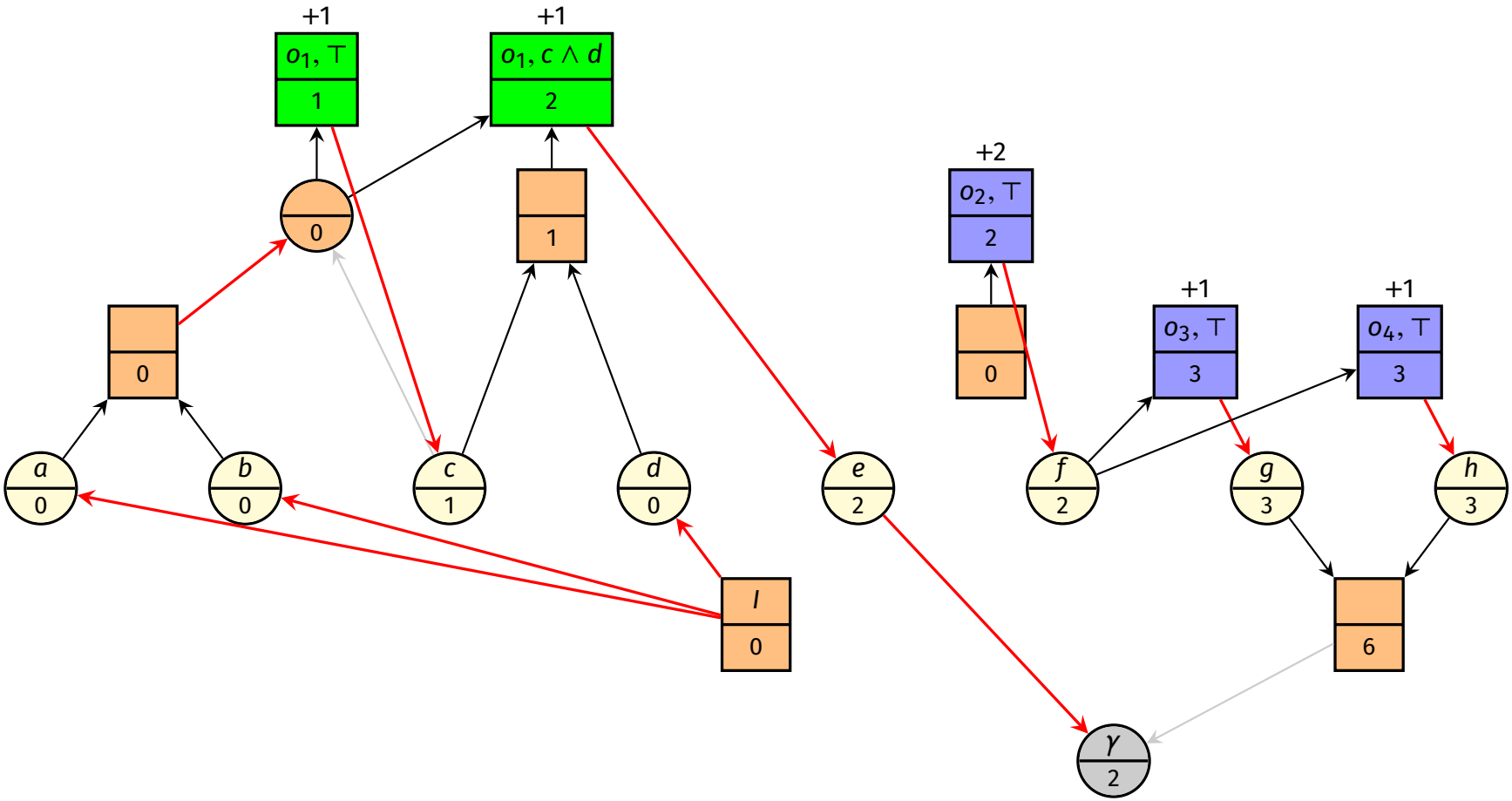
FF heuristic computation; modified goal $e \vee (g \wedge h)$



Compute effect nodes reachable from goal node.

Example: FF Heuristic (2)

FF heuristic computation; modified goal $e \vee (g \wedge h)$



$$h^{\text{FF}}(s) = 1 + 1 = 2$$

$h^{\max}$ vs. h^{add} vs. h^{FF} vs. h^{+}

Reminder: Optimal Delete Relaxation Heuristic

Definition (h^+ Heuristic)

Let Π be a propositional planning task in positive normal form, and let s be a state of Π.

The optimal delete relaxation heuristic for s, written $h^+(s)$, is the perfect heuristic value $h^*(s)$ of state s in the delete-relaxed task Π^+.

- Reminder: $h^+(s)$ is hard to compute.
  (BCPlanEx is NP-complete for delete-relaxed tasks.)
- The optimal delete relaxation heuristic is often used as a reference point for comparison.

Relationships between Delete Relaxation Heuristics (1)

Theorem

Let Π be a propositional planning task in positive normal form, and let s be a state of Π.

Then:

1. $h^{max}(s) \leq h^{+}(s) \leq h^{FF}(s) \leq h^{add}(s)$
2. $h^{max}(s) = \infty$ *iff* $h^{+}(s) = \infty$ *iff* $h^{FF}(s) = \infty$ *iff* $h^{add}(s) = \infty$
3. h^{max} *and* h^{+} *are admissible and consistent.*
4. h^{FF} *and* h^{add} *are neither admissible nor consistent.*
5. *All four heuristics are safe and goal-aware.*

Summary

Summary

- The FF heuristic repairs the double-counting of h^{add} and therefore approximates h^+ more closely.
- The key idea is to mark all effect nodes "used" for the h^{add} value of the goal and count each of them once.
- In general, $h^{\max}(s) \leq h^+(s) \leq h^{\text{FF}}(s) \leq h^{\text{add}}(s)$.
- $h^{\max}$ and h^+ are admissible; h^{FF} and h^{add} are not.