Planning and Optimization

C3. Delete Relaxation: AND/OR Graphs

Malte Helmert and Gabriele Röger

Universität Basel

October 27, 2016

Planning and Optimization

October 27, 2016 — C3. Delete Relaxation: AND/OR Graphs

C3.1 AND/OR Graphs

C3.2 Forced Nodes

C3.3 Most and Least Conservative Valuation

C3.4 Summary

Motivation

- Our next goal is to devise efficiently computable heuristics based on delete relaxation.
- The heuristics we will consider can all be understood in terms of computations on graphical structures called AND/OR graphs.
- In this chapter, we introduce AND/OR graphs and study some of their major properties.
- In the next chapter, we will relate AND/OR graphs to relaxed planning tasks.

C3.1 AND/OR Graphs

AND/OR Graphs

Definition (AND/OR Graph)

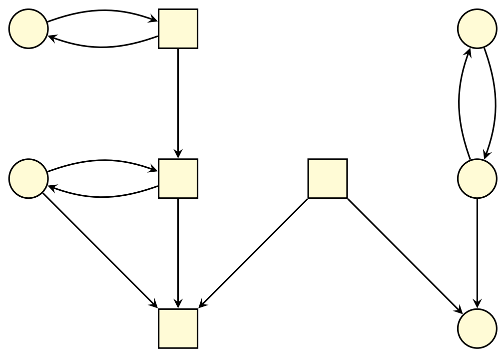
An AND/OR graph $\langle N, A, type \rangle$ is a directed graph $\langle N, A \rangle$ with a node label function $type : N \to \{\wedge, \vee\}$ partitioning nodes into

- AND nodes ($type(v) = \wedge$) and
- OR nodes ($type(v) = \vee$).

We write $succ(n)$ for the successors of node $n \in N$, i.e., $succ(n) = \{n' \in N \mid \langle n, n' \rangle \in A\}$.

Note: We draw AND nodes as squares and OR nodes as circles.

AND/OR Graph Example



AND/OR Graph Valuations

Definition (Consistent Valuations of AND/OR Graphs)

Let G be an AND/OR graph with nodes N.

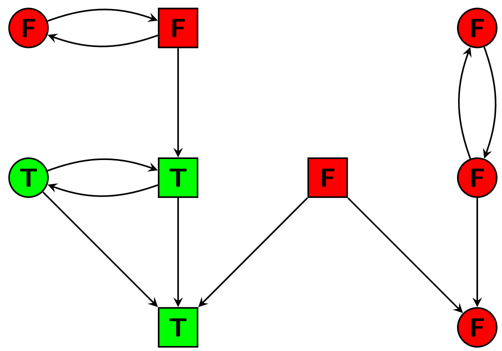
A valuation or truth assignment of G is a valuation $\alpha : N \to \{\mathbf{T}, \mathbf{F}\}$, treating the nodes as propositional variables.

We say that α is consistent if

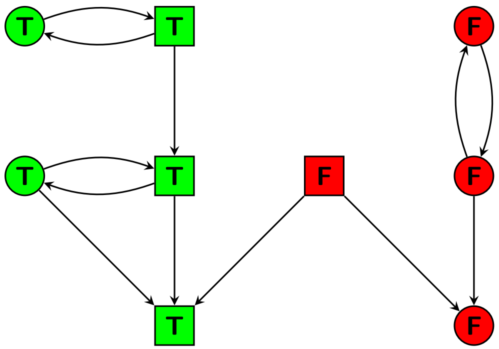
- for all AND nodes $n \in N$: $\alpha \models n$ iff $\alpha \models \bigwedge_{n' \in succ(n)} n'$.
- for all OR nodes $n \in N$: $\alpha \models n$ iff $\alpha \models \bigvee_{n' \in succ(n)} n'$.

Note that $\bigwedge_{n' \in \emptyset} n' = \top$ and $\bigvee_{n' \in \emptyset} n' = \bot$.

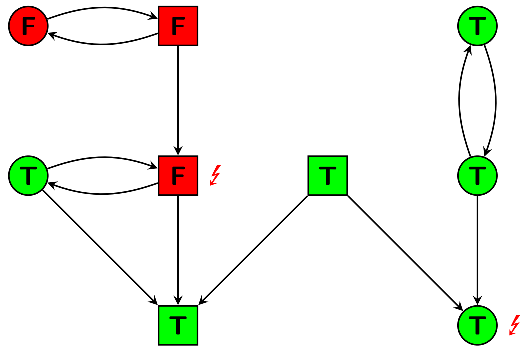
Example: A Consistent Valuation



Example: Another Consistent Valuation



Example: An Inconsistent Valuation



C3.2 Forced Nodes

How Do We Find Consistent Valuations?

If we want to use valuations of AND/OR graphs algorithmically, a number of questions arise:

- Do consistent valuations exist for every AND/OR graph?
- Are they unique?
- If not, how are different consistent valuations related?
- Can consistent valuations be computed efficiently?

Some Partial Answers

- We already know from our previous example that consistent valuations are in general not unique.
- We will now study two special kinds of valuations:
  - the most conservative valuation
  - the least conservative valuation
- We show that these two valuations are always consistent and can be computed efficiently.
- We also show that all consistent valuations lie "in between" these two valuations.
- In particular, an AND/OR graph has a unique consistent valuation iff its most and least conservative valuation coincide.

Forced True Nodes

Definition (Forced True Nodes)

Let G be an AND/OR graph.

The set of nodes of G that are forced true is defined by finite application of the following rules:

- If n is an AND node where all successors are forced true, then n is forced true.
- If n is an OR node where at least one successor is forced true, then n is forced true.

Forced False Nodes

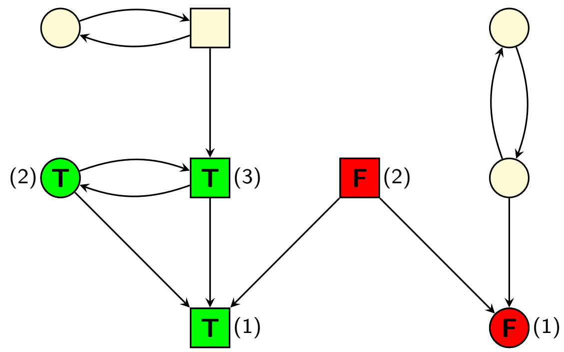
Definition (Forced False Nodes)

Let G be an AND/OR graph.

The set of nodes of G that are forced false is defined by finite application of the following rules:

- If n is an AND node where at least one successor is forced false, then n is forced false.
- If n is an OR node where all successors are forced false, then n is forced false.

Example: Forced Nodes



Remarks on Forced Nodes

Notes:

- Forced nodes are well-defined because the rules defining them are monotonic.
- They can be computed in linear time in the size of the graph.
- There exists at least one forced true node iff the graph has an AND node without successors.
- There exists at least one forced false node iff the graph has an OR node without successors.

Forced Nodes and Consistent Valuations

Theorem

Let G be an AND/OR graph with nodes N.

Let $N_{\mathbf{T}} \subseteq N$ be the forced true nodes of G, and let $N_{\mathbf{F}} \subseteq N$ be the forced false nodes of G.

Let $\alpha : N \to \{\mathbf{T}, \mathbf{F}\}$ be a consistent valuation of G. Then:

- $\alpha(n) = \mathbf{T}$ *for all* $n \in N_{\mathbf{T}}$
- $\alpha(n) = \mathbf{F}$ *for all* $n \in N_{\mathbf{F}}$

This property explains why they are called forced nodes.

Forced Nodes and Consistent Valuations: Proof (1)

Proof.

We prove the property of forced true nodes.
The property of forced false nodes can be proved analogously.

Proof by contradiction: assume that there is a consistent valuation α and a forced true node $n \in N_{\mathbf{T}}$ with $\alpha(n) \neq \mathbf{T}$, i.e., $\alpha(n) = \mathbf{F}$.

Let $N_{\mathbf{T}} = \{n_1, \dots, n_k\}$, where the nodes are ordered in a way that matches a possible sequence of rule applications:
i.e., the i-th rule application derives that n_i is forced true.

Let n_i be the first node in the order with $\alpha(n_i) = \mathbf{F}$.
Hence, $\alpha(n_j) = \mathbf{T}$ for all $j < i$. ...

Forced Nodes and Consistent Valuations: Proof (2)

Proof (continued).

Case 1: n_i is an AND node.

In the i-th rule application, n_i is shown as forced true, so all its successors must have been shown as forced true in an earlier rule application.

Hence all nodes in $succ(n_i)$ are of the form n_j with $j < i$.
By the choice of n_i, we have $\alpha(n_j) = \mathbf{T}$ for all these nodes.

We conclude $\alpha \not\models n_i$ and $\alpha \models \bigwedge_{n' \in succ(n_i)} n'$,
which shows that α is not consistent: a contradiction. ...

Forced Nodes and Consistent Valuations: Proof (3)

Proof (continued).

Case 2: n_i is an OR node.

In the i-th rule application, n_i is shown as forced true, so at least one successor must have been shown as forced true in an earlier rule application.

Hence at least one node in $succ(n_i)$ is of the form n_j with $j < i$. By the choice of n_i, we have $\alpha(n_j) = \mathbf{T}$ for such nodes.

We conclude $\alpha \not\models n_i$ and $\alpha \models \bigvee_{n' \in succ(n_i)} n'$, which shows that α is not consistent: a contradiction. □

C3.3 Most and Least Conservative Valuation

Most and Least Conservative Valuation

Definition (Most and Least Conservative Valuation)

Let G be an AND/OR graph with nodes N.

The most conservative valuation $\alpha^G_{\mathsf{mcv}} : N \to \{\mathbf{T}, \mathbf{F}\}$ and the least conservative valuation $\alpha^G_{\mathsf{lcv}} : N \to \{\mathbf{T}, \mathbf{F}\}$ of G are defined as:

$$\alpha^G_{\mathsf{mcv}}(n) = \begin{cases} \mathbf{T} & \text{if } n \text{ is forced true} \\ \mathbf{F} & \text{otherwise} \end{cases}$$

$$\alpha^G_{\mathsf{lcv}}(n) = \begin{cases} \mathbf{F} & \text{if } n \text{ is forced false} \\ \mathbf{T} & \text{otherwise} \end{cases}$$

Most/Least Conservative Valuations are Consistent

Theorem

Let G be an AND/OR graph. Then:

1. *α^G_{mcv} is consistent.*
2. *α^G_{lcv} is consistent.*
3. *For all consistent valuations α of G, $on(\alpha^G_{\mathsf{mcv}}) \subseteq on(\alpha) \subseteq on(\alpha^G_{\mathsf{lcv}})$.*

It follows that:

- Consistent valuations always exist and can be efficiently computed.
- All consistent valuations lie between the most and least conservative one.
- There is a unique consistent valuation iff $\alpha^G_{\mathsf{mcv}} = \alpha^G_{\mathsf{lcv}}$, or equivalently iff each node is forced true or forced false.

MCV/LCV are Consistent: Proof (1)

Proof.

We prove part 1. Part 2 is analogous, and part 3 follows directly from the previous result and the definitions of α^G_{mcv} and α^G_{lcv}.

To prove part 1, we must show that α^G_{mcv} satisfies the consistency condition for every node n. Consider any node n. ...

MCV/LCV are Consistent: Proof (2)

Proof (continued).

Case 1a: n is an AND node and n is forced true.

All successors of n are forced true.
(Otherwise n would not be forced true.)
Then α^G_{mcv} maps n and all its successors to **T**,
satisfying the consistency condition for n.

Case 1b: n is an AND node and n is not forced true.

At least one successor n' of n is not forced true.
(Otherwise n would be forced true.)
Then α^G_{mcv} maps n and n' to false,
satisfying the consistency condition for n. ...

MCV/LCV are Consistent: Proof (3)

Proof (continued).

Case 2a: n is an OR node and n is forced true.

At least one successor n' of n is forced true.
(Otherwise n would not be forced true.)
Then α^G_{mcv} maps n and n' to true,
satisfying the consistency condition for n.

Case 2b: n is an OR node and n is not forced true.

No successor of n is forced true.
(Otherwise n would be forced true.)
Then α^G_{mcv} maps n and all its successors to **F**,
satisfying the consistency condition for n. □

C3.4 Summary

- AND/OR graphs are directed graphs with AND nodes and OR nodes.
- We can assign truth values to AND/OR graph nodes.
- Such valuations are called consistent if they match the intuitive meaning of “AND” and “OR”.
- Consistent valuations always exist.
- Consistent valuations can be computed efficiently.
- All consistent valuations fall between two extremes:
  - the most conservative valuation, where only nodes that are forced to be true are true
  - the least conservative valuation, where all nodes that are not forced to be false are true