

Relevance for SAT(ID)

Joachim Jansen¹, Bart Bogaerts^{1,2}, Jo Devriendt¹
Gerda Janssens¹, Marc Denecker¹

1 KU Leuven, Leuven, Belgium, firstname.lastname@kuleuven.be

2 Aalto University, Espoo, Finland, bart.bogaerts@aalto.fi

July 13th, 2016

The logo for KU Leuven, featuring the text "KU LEUVEN" in white, bold, uppercase letters on a dark blue rectangular background with a light blue border.The logo for Aalto University School of Science, featuring a large black letter "A" with a red exclamation mark to its right, and the text "Aalto University School of Science" below it.

PC(ID), SAT(ID)

- ▶ $\text{SAT}(\text{ID}) = \text{satisfiability check of PC}(\text{ID})$

PC(ID), SAT(ID)

- ▶ SAT(ID) = satisfiability check of PC(ID)
- ▶ **P**ropositional **C**alculus + **I**nductive **D**efinitions

PC(ID), SAT(ID)

- ▶ SAT(ID) = satisfiability check of PC(ID)
- ▶ **P**ropositional **C**alculus + **I**nductive **D**efinitions
- ▶ PC(ID) encoding $\mathcal{T} = \{p_{\mathcal{T}}, \Delta\}$

PC(ID), SAT(ID)

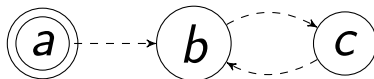
- ▶ SAT(ID) = satisfiability check of PC(ID)
- ▶ **P**ropositional **C**alculus + **I**nductive **D**efinitions
- ▶ PC(ID) encoding $\mathcal{T} = \{p_{\mathcal{T}}, \Delta\}$
- ▶ $p_{\mathcal{T}} = \text{true}$ iff \mathcal{T} is satisfied, Δ defines everything

PC(ID), SAT(ID)

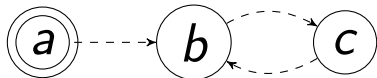
- ▶ SAT(ID) = satisfiability check of PC(ID)
- ▶ **P**ropositional **C**alculus + **I**nductive **D**efinitions
- ▶ PC(ID) encoding $\mathcal{T} = \{p_{\mathcal{T}}, \Delta\}$
- ▶ $p_{\mathcal{T}} = \text{true}$ iff \mathcal{T} is satisfied, Δ defines everything

Example

- ▶ Only node a is initially reachable
- ▶ Choose edges and colors of nodes s.t.
 - ▶ node b is reachable
 - ▶ every reachable node is colored green



Example (continued)

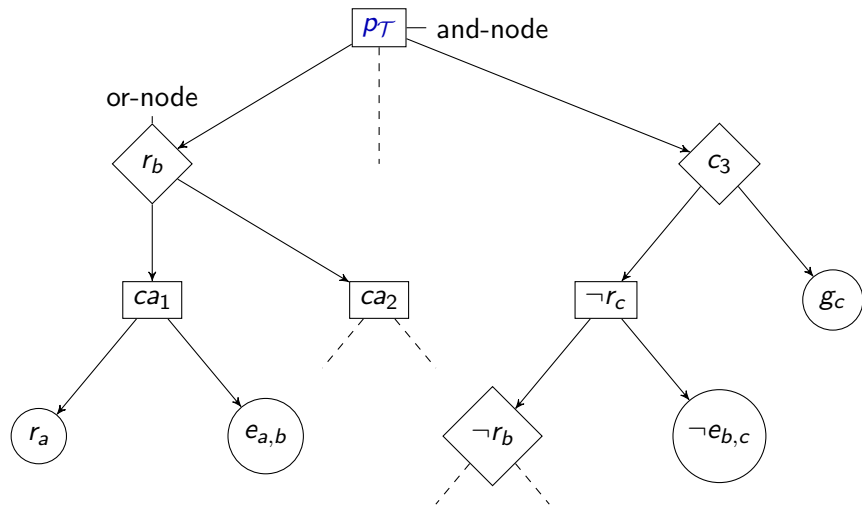


$$\Delta = \left\{ \begin{array}{l} p_T \leftarrow reach_b \wedge constr_1 \wedge constr_2 \wedge constr_3. \\ constr_1 \leftarrow \neg reach_a \vee green_a. \\ constr_2 \leftarrow \neg reach_b \vee green_b. \\ constr_3 \leftarrow \neg reach_c \vee green_c. \\ reach_a \quad . \\ reach_b \leftarrow case_1 \vee case_2. \\ case_1 \leftarrow reach_a \wedge edge_{a,b}. \\ case_2 \leftarrow reach_c \wedge edge_{c,b}. \\ reach_c \leftarrow reach_b \wedge edge_{b,c}. \end{array} \right.$$

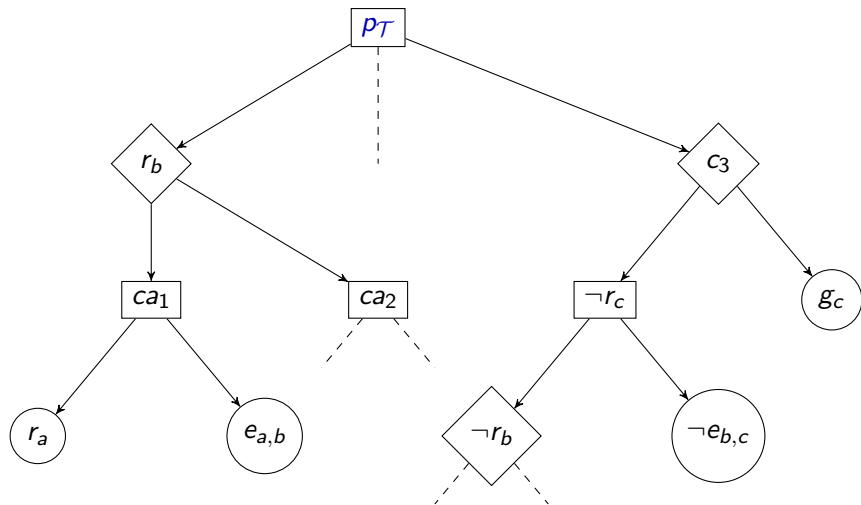
- ▶ $reach_x$ = node x is reached
- ▶ $constr_x$ = color constraints on node x

- ▶ $green_x$ = node x is green
- ▶ $edge_{x,y}$ = edge from x to y selected

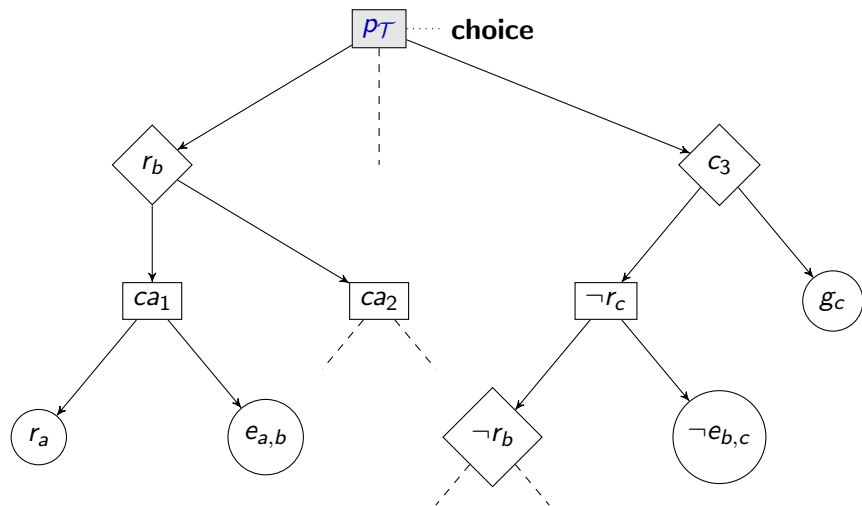
Visualising the hierarchy



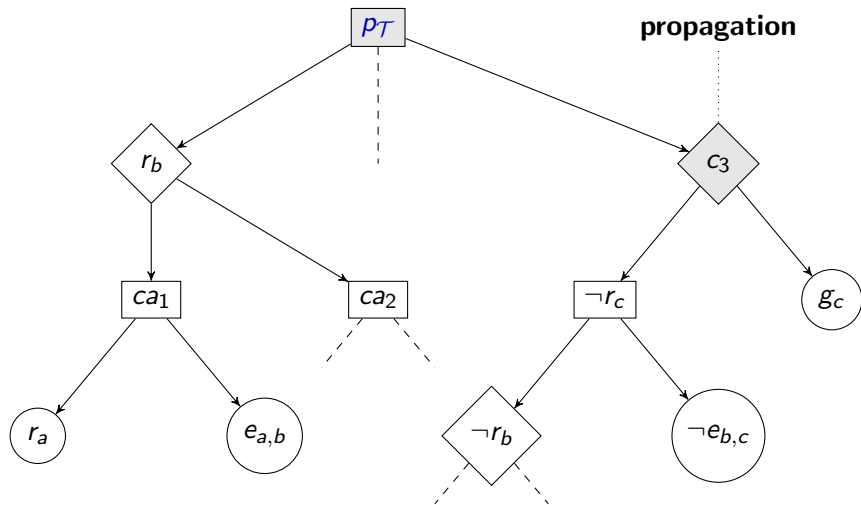
Visualising the Search process



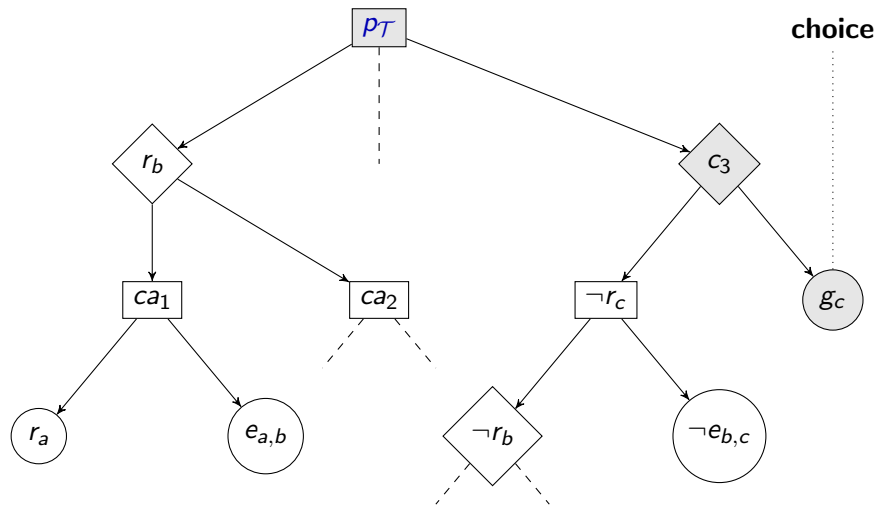
Visualising the Search process



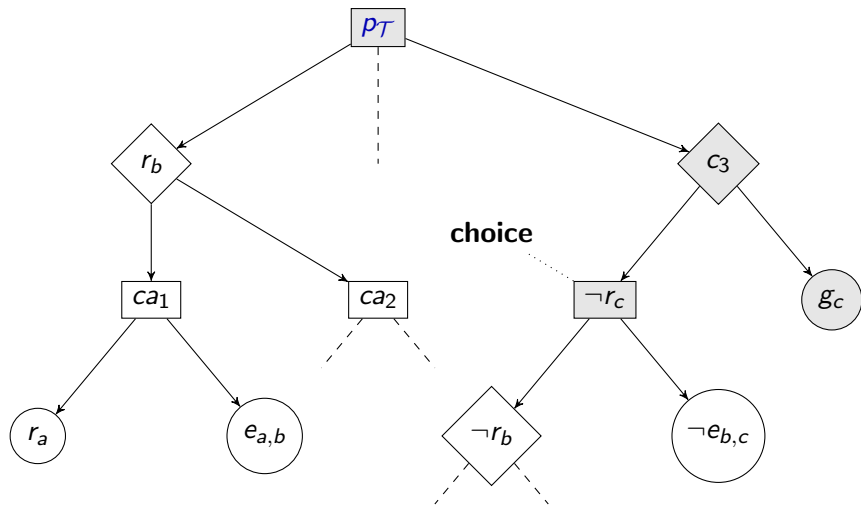
Visualising the Search process



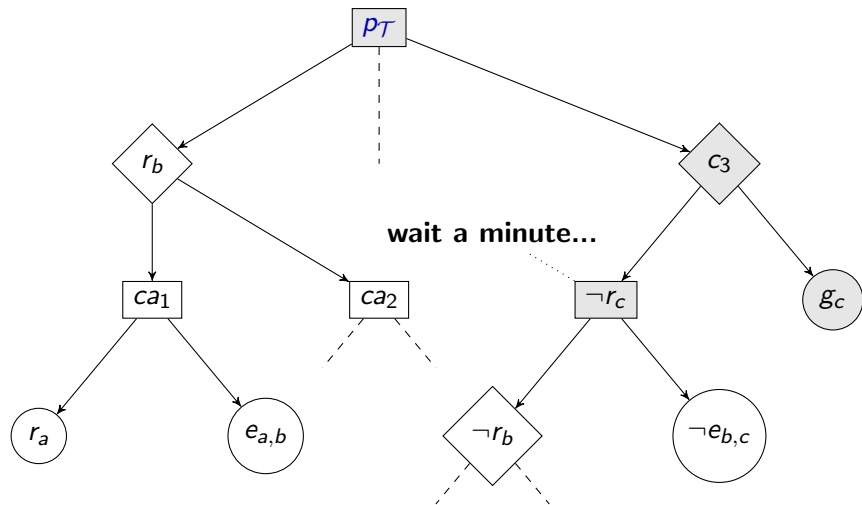
Visualising the Search process



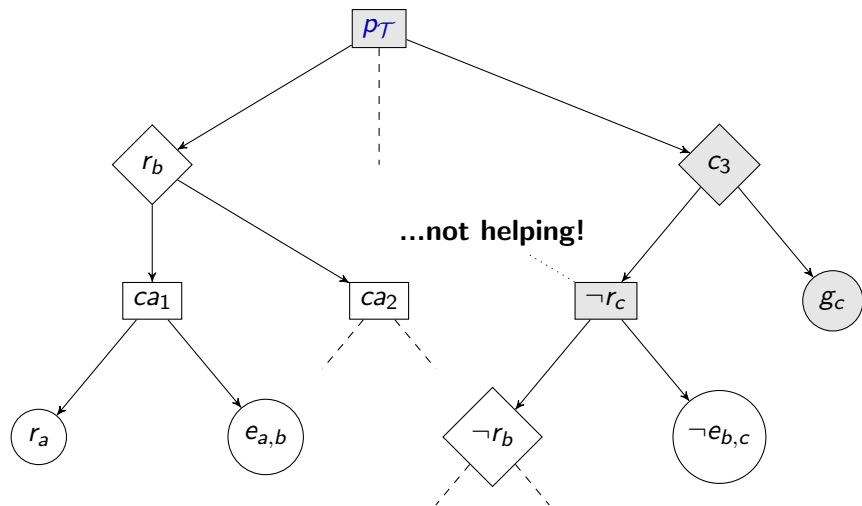
Visualising the Search process



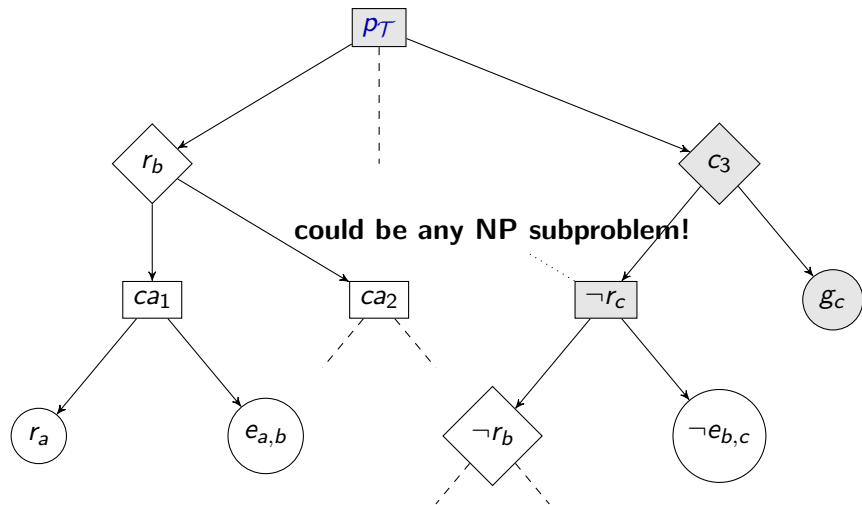
Visualising the Search process



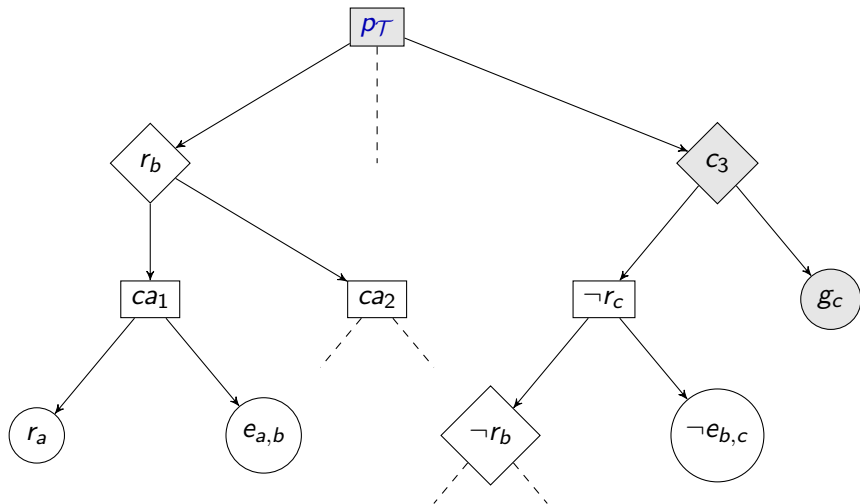
Visualising the Search process



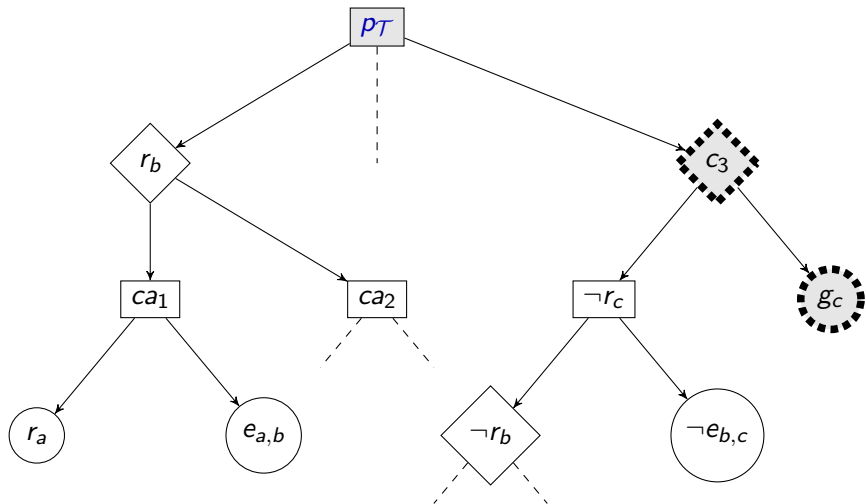
Visualising the Search process



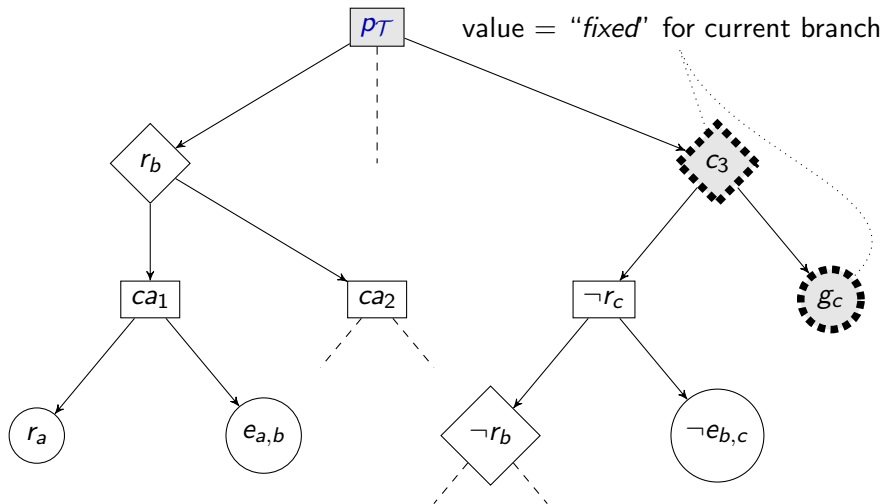
Making stuff true \rightarrow making stuff *justified*



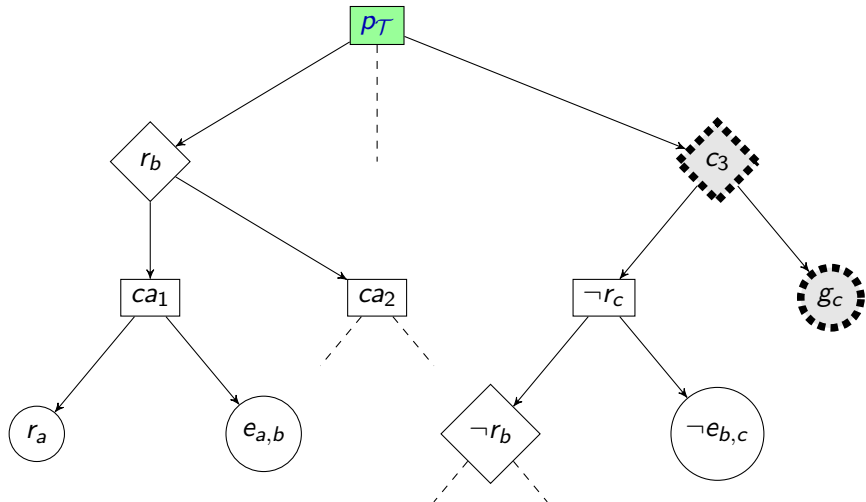
Making stuff true \rightarrow making stuff *justified*



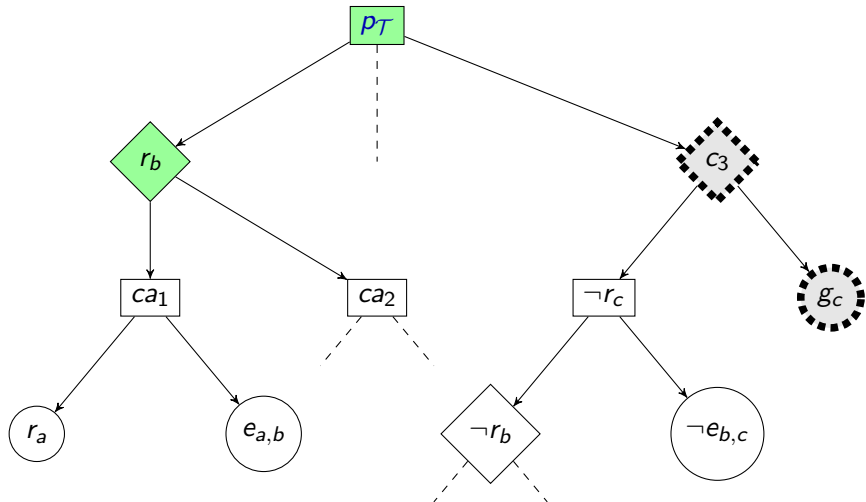
Making stuff true \rightarrow making stuff *justified*



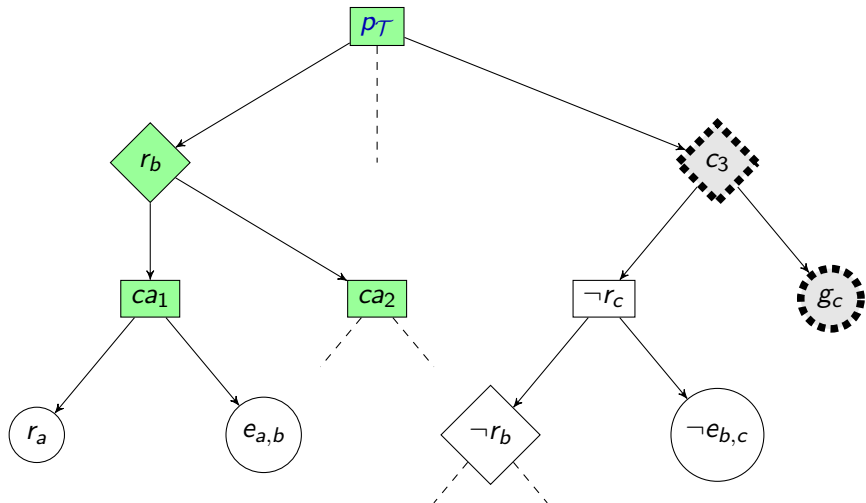
Relevant = can still influence p_T



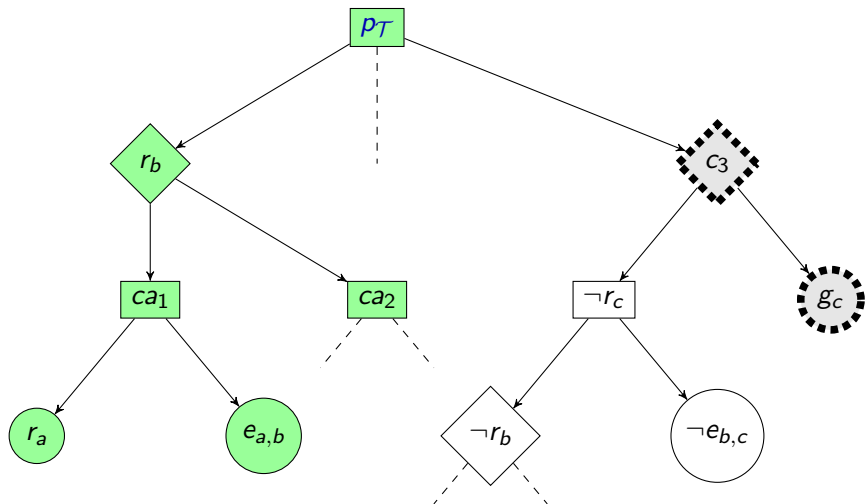
Relevant = can still influence p_T



Relevant = can still influence p_T



Relevant = can still influence p_T



Adjusting the Solver

- ▶ Decide only on **Relevant** literals.

Adjusting the Solver

- ▶ Decide only on **Relevant** literals.
- ▶ Stop search when p_T is *justified*
 - ▶ Guarantee that a two-valued solution can be generated efficiently

Experiment Setup (1)

- ▶ Problems from previous ASP competitions
- ▶ Solver = Minisatid, Heuristic = VSIDS

Experiment Setup (1)

- ▶ Problems from previous ASP competitions
- ▶ Solver = Minisatid, Heuristic = VSIDS
- ▶ Measuring
 - ▶ Ratio of irrelevant decisions (%)
 - ▶ Ratio of conflicts originating from irrelevant decisions (%)

Experimental Results (1)

Problem	% Irr. Decisions	% Irr. Conflicts
HP	27.37%	36.99%
NQueens	22.55%	0.43%
PPM	22.93%	4.98%
Sokoban	48.20%	0.96%
Solitaire	13.32%	3.95%
SM	96.40%	0.01%
Visit All	15.02%	16.45%

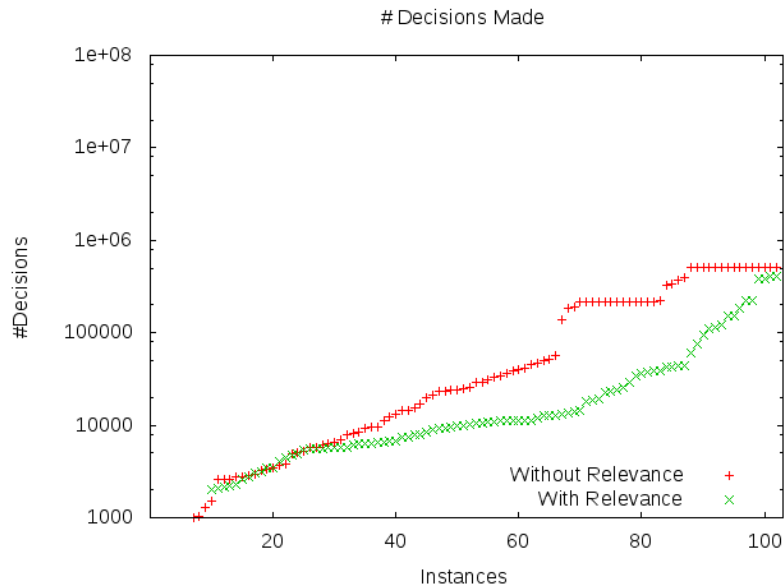
Experiment Setup (2)

- ▶ Problems from previous ASP competitions
- ▶ Solver = Minisatid, Heuristic = VSIDS

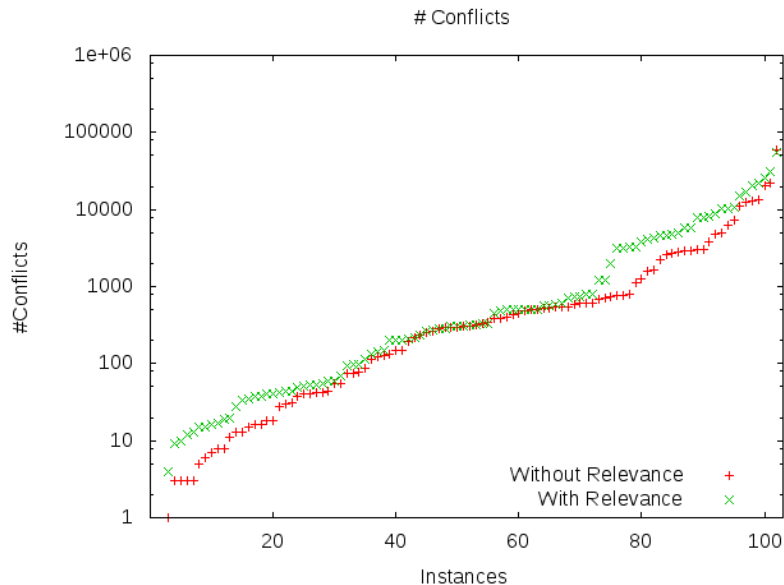
Experiment Setup (2)

- ▶ Problems from previous ASP competitions
- ▶ Solver = Minisatid, Heuristic = VSIDS
- ▶ Measuring
 - ▶ Number of decisions (#)
 - ▶ Number of conflicts (#)

Experimental Results (2)



Experimental Results (2)



Take-away messages

- ▶ Exploit problem hierarchy using *Relevance*

Take-away messages

- ▶ Exploit problem hierarchy using **Relevance**
- ▶ Preliminary promising results: fewer *decisions*

Take-away messages

- ▶ Exploit problem hierarchy using **Relevance**
- ▶ Preliminary promising results: fewer *decisions*
- ▶ Hand-crafted example with fewer *conflicts*: see paper.

Questions?