And-Tree-based Search Example: Model-elimiation

CPSC 433: Artificial Intelligence
Fall 2024

Jonathan Hudson, Ph.D.
Assistant Professor (Teaching)
Department of Computer Science
University of Calgary

August 8, 2024

UNIVERSITY OF CALGARY

And-Tree Applied to Model-elimination

Concrete Example: Model-elimination

- Another, now analytical, way to solve the problem of determining if a formula is a consequence of a set of formulas
- Again works with sets of clauses
- A problem is divided into subproblems by employing a clause $L_1 \vee \ldots \vee L_n$:
  n subproblems are generated, each of which assumes that additionally a certain instance σ of L_i is true (each subproblem uses a different L_i but the same σ)
  - Note: σ is the mgu as we saw prior

Modelelimination (II)

- We start with a "world" containing no predicate or its negation (i.e. everything is possible)

- Then we select a leaf in our tree and a clause
  $L_1 \vee \ldots \vee L_n$ and generate the successor nodes as described above.
  One additional condition is that at least one of the resulting subproblems is solved (except for a transition out of the "empty" world).

- A subproblem is solved, if it contains P and $\neg$P' such that there is a σ with $\sigma(P) \equiv \sigma(P')$ (usually we use $\sigma = mgu(P,P')$)

Modelelimination (III)

- By using the mgu, each time we do this, we have to apply it to all subproblems we have generated so far (in order to guarantee that solutions to subproblems are compatible).

- Our problem is solved (positively), if all subproblems are solved.

Model-elimination: Examples

Modelelimination (IV)

- Solve the following problem instances:

1) $p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$
2) $p, q, \neg q$
3) $P(x) \lor R(x), \neg R\big(f(a,b)\big), \neg P(g(a,b))$

Example 1

Modelelimination (IV)

- Solve the following problem instances:

$p \vee q, p \vee \neg q, \neg p \vee q, \neg p \vee \neg q$

Modelelimination (IV)

- Solve the following problem instances:

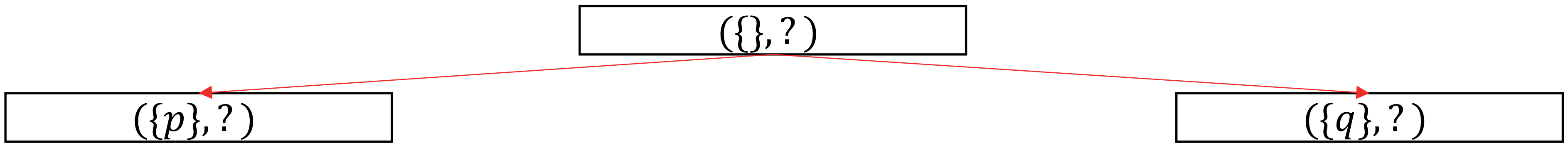
$p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$

$(\{\}, ?)$

Modelelimination (IV)

- Solve the following problem instances:

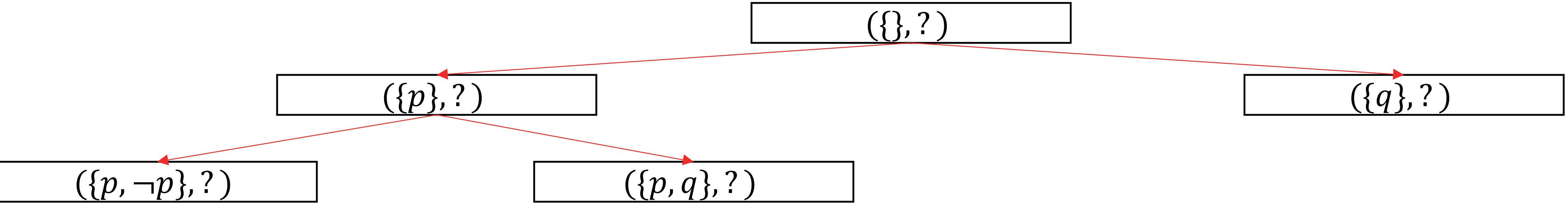
$\boldsymbol{p} \lor \boldsymbol{q}, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$



Modelelimination (IV)

- Solve the following problem instances:

$p \lor q, p \lor \neg q, \neg \boldsymbol{p} \lor \boldsymbol{q}, \neg p \lor \neg q$

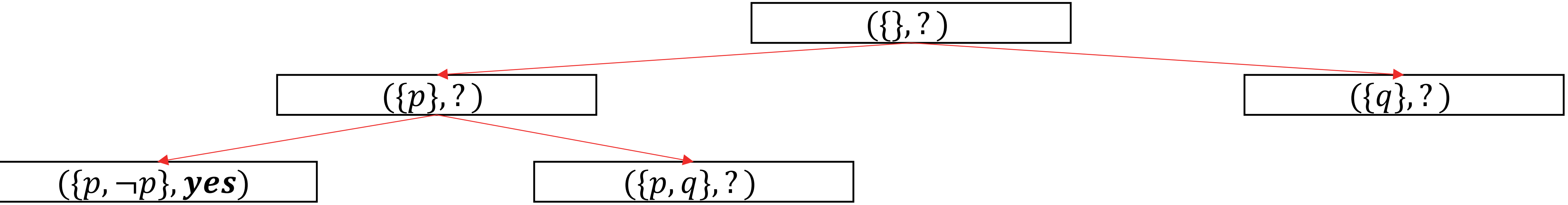


UNIVERSITY OF CALGARY

Modelelimination (IV)

- Solve the following problem instances:

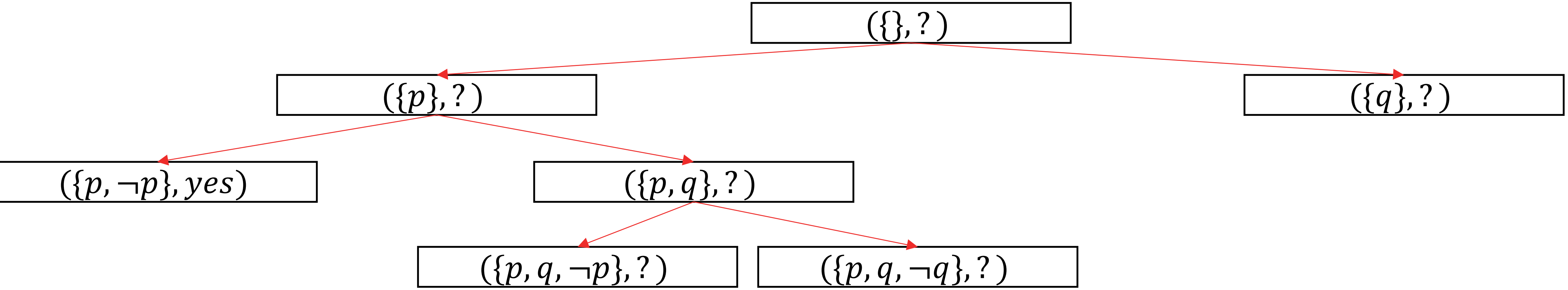
$p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$



Modelelimination (IV)

- Solve the following problem instances:

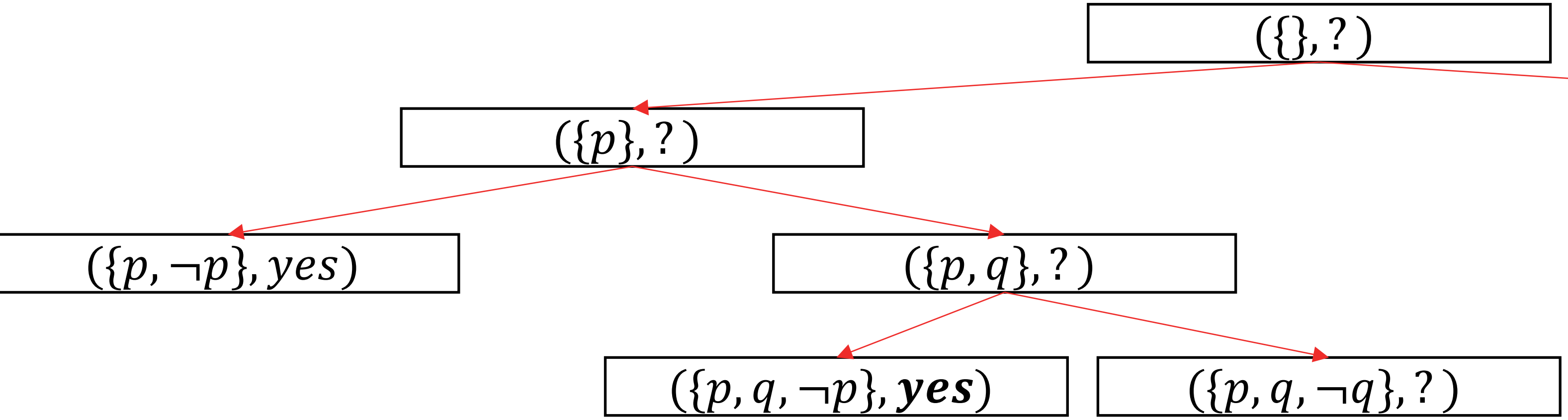
$p \lor q, p \lor \neg q, \neg p \lor q, \neg \boldsymbol{p} \lor \neg \boldsymbol{q}$



Modelelimination (IV)

- Solve the following problem instances:

$p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$

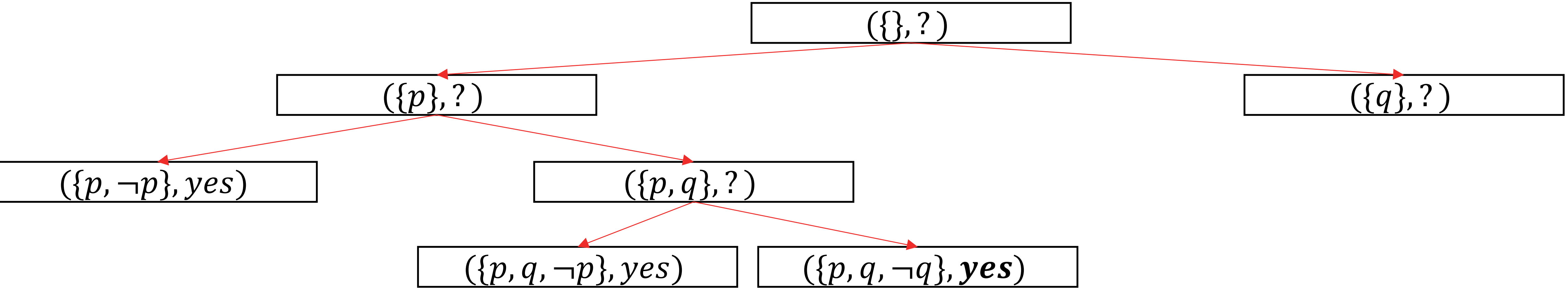


UNIVERSITY OF CALGARY

Modelelimination (IV)

- Solve the following problem instances:

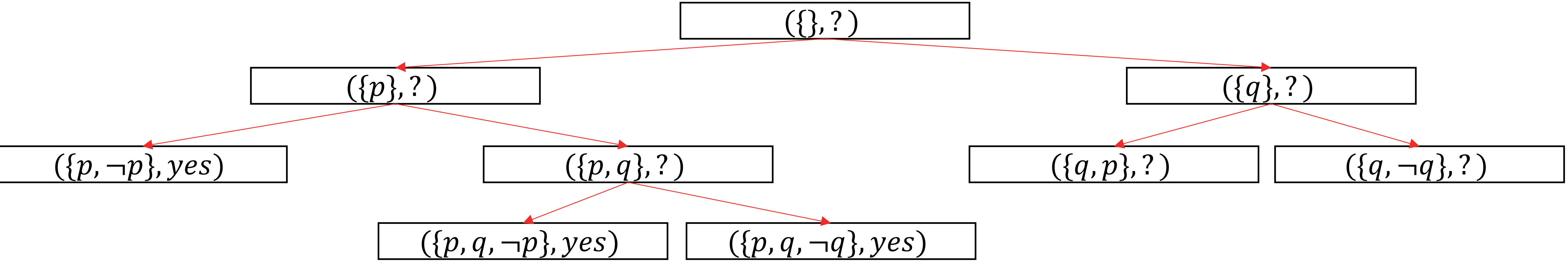
$p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$



Modelelimination (IV)

- Solve the following problem instances:

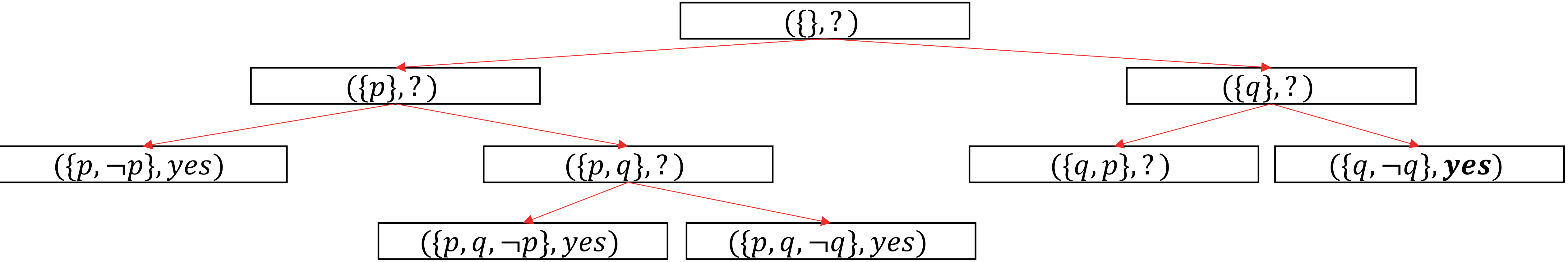
$p \lor q, \boldsymbol{p} \lor \neg \boldsymbol{q}, \neg p \lor q, \neg p \lor \neg q$



Modelelimination (IV)

- Solve the following problem instances:

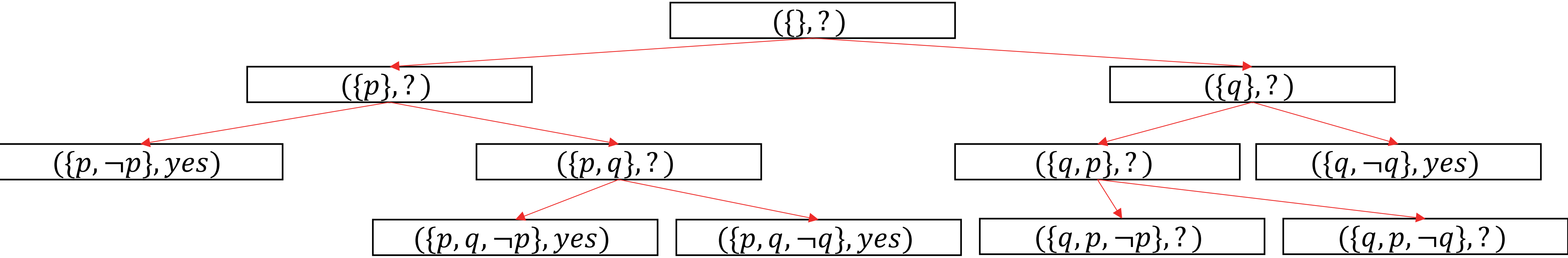
$p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$



Modelelimination (IV)

- Solve the following problem instances:

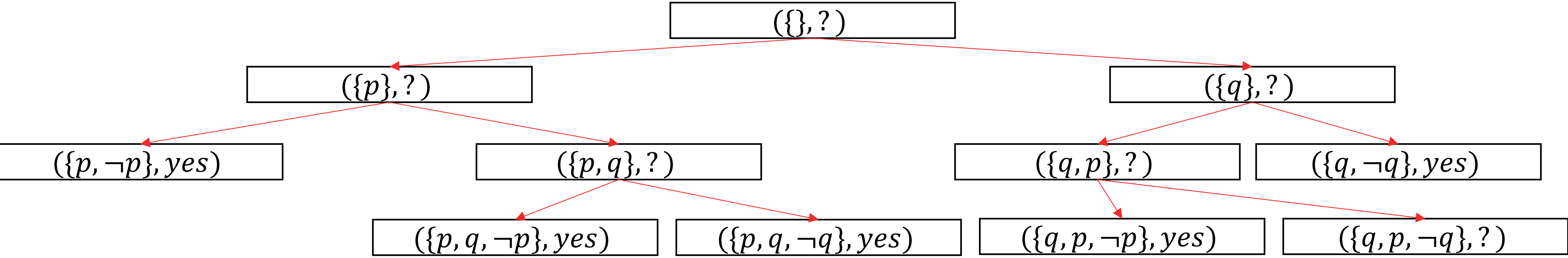
$p \lor q, p \lor \neg q, \neg p \lor q, \neg \boldsymbol{p} \lor \neg \boldsymbol{q}$



Modelelimination (IV)

- Solve the following problem instances:

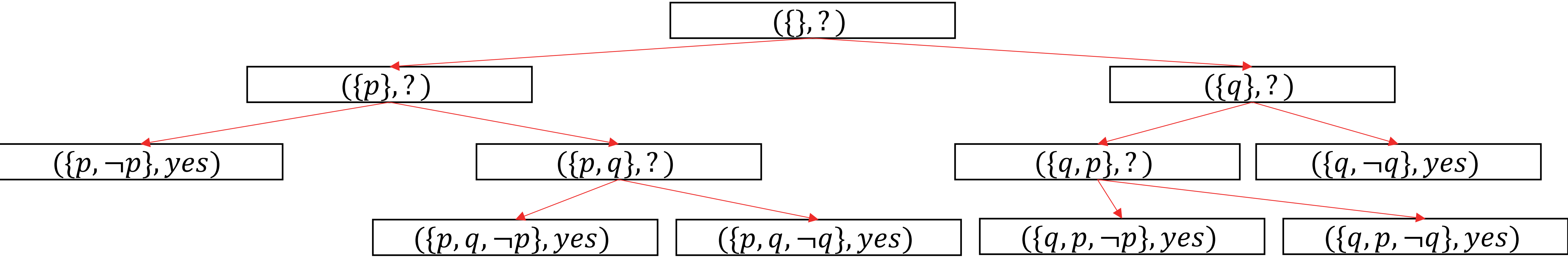
$p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$



Modelelimination (IV)

- Solve the following problem instances:

$p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$

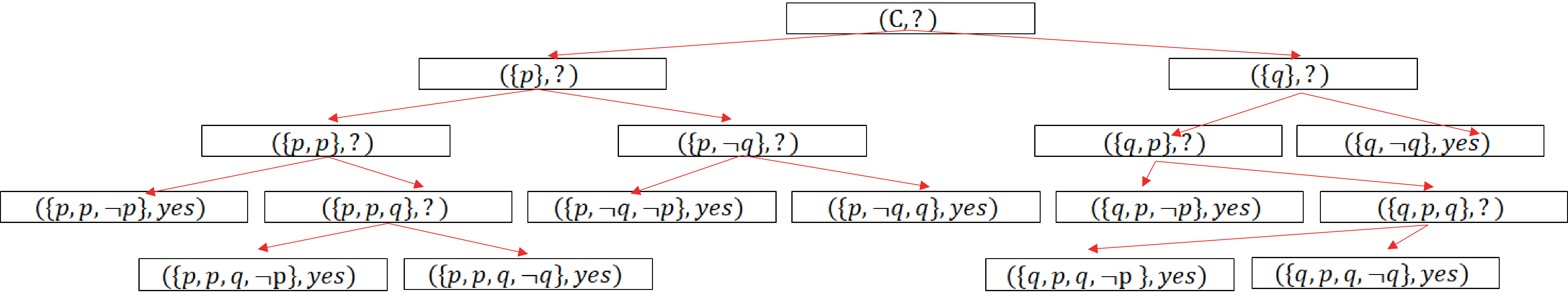


Modelelimination (IV)

- More than one way to solve! Search control matters!

$p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$

$\emptyset = C$



Example 2

Modelelimination (IV)

- Solve the following problem instances:

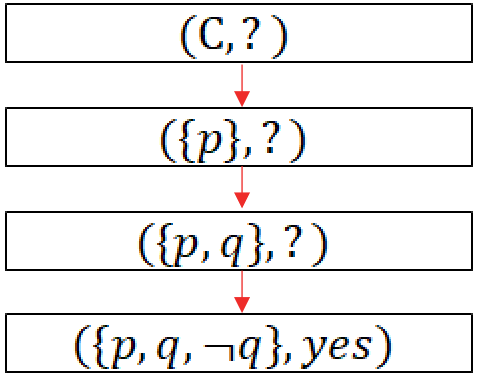
$p\ , q, \neg\ q$
$\emptyset = C$

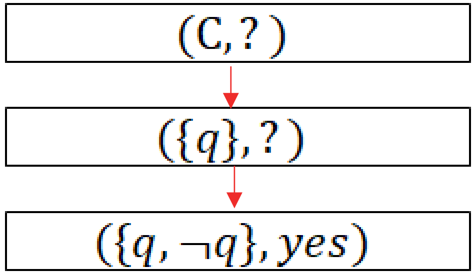
Modelelimination (IV)

- Again more than one way to solve:

$p\ , q, \neg\ q$

$\emptyset = C$





UNIVERSITY OF CALGARY

Example 3

Modelelimination (IV)

- Solve the following problem instances:

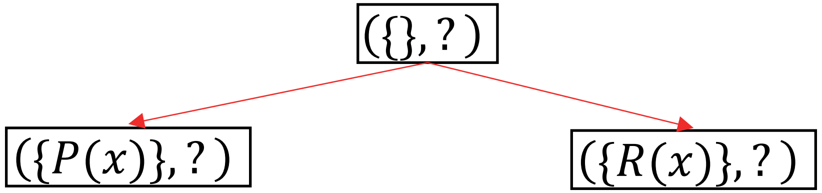
$P(x) \vee R(x), \neg R(f(a,b)), \neg P(g(a,b))$

$\boxed{(\{\}, ?)}$

Modelelimination (IV)

- Solve the following problem instances:

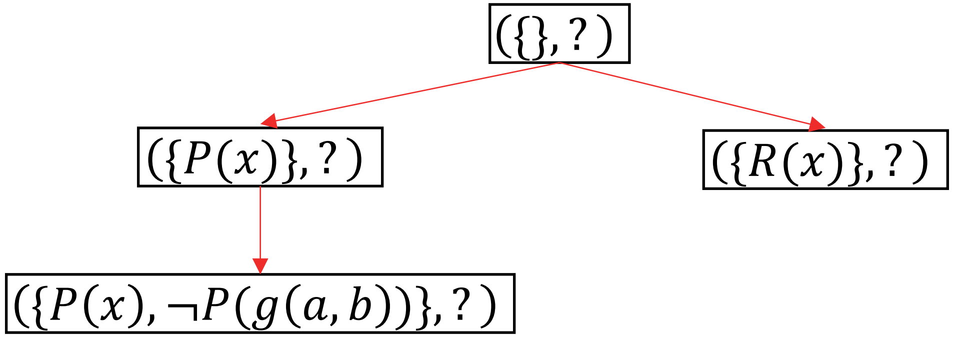
P(x) $\vee$ **R(x),** $\neg$R(f(a,b)), $\neg$P(g(a,b))



Modelelimination (IV)

- Solve the following problem instances:

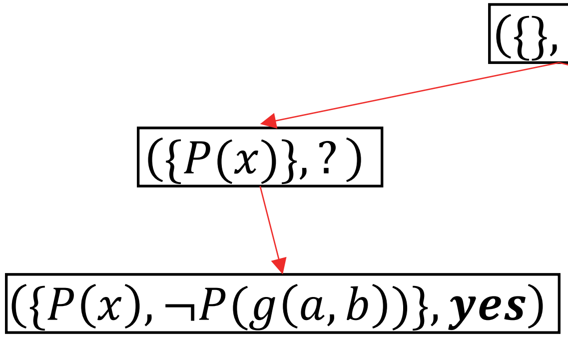
$P(x) \vee R(x)$, $\neg R(f(a,b))$, $\mathbf{\neg P(g(a,b))}$



Modelelimination (IV)

- Solve the following problem instances:

$P(x) \vee R(x)$, $\neg R(f(a,b))$, $\neg P(g(a,b))$



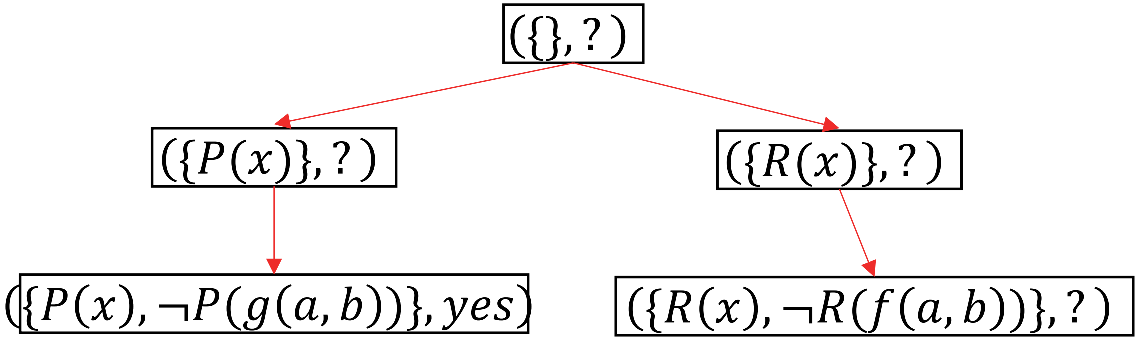
$$\boldsymbol{mgu} = \{\boldsymbol{x} \approx \boldsymbol{g}(\boldsymbol{a}, \boldsymbol{b})\}$$

UNIVERSITY OF CALGARY

Modelelimination (IV)

- Solve the following problem instances:

P(x) ∨ R(x), **¬R(f(a,b)),** ¬P(g(a,b))



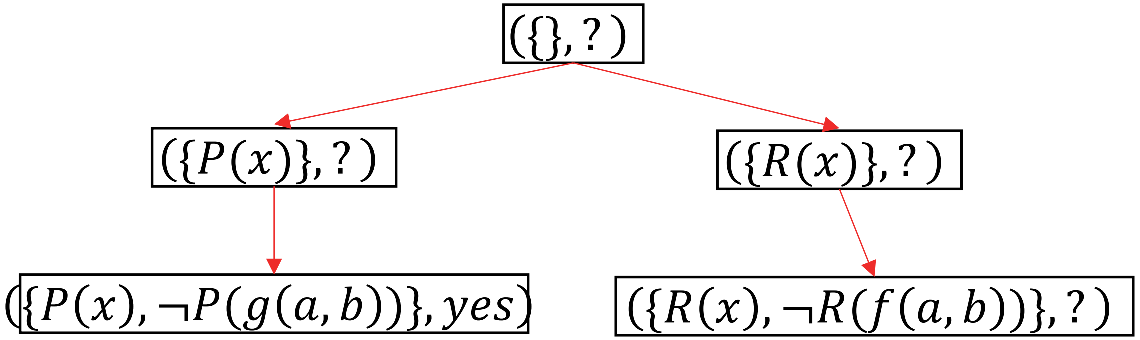
$$mgu = \{x \approx g(a,b)\}$$

UNIVERSITY OF CALGARY

Modelelimination (IV)

- Solve the following problem instances:

$P(x) \vee R(x), \neg R(f(a,b)), \neg P(g(a,b))$



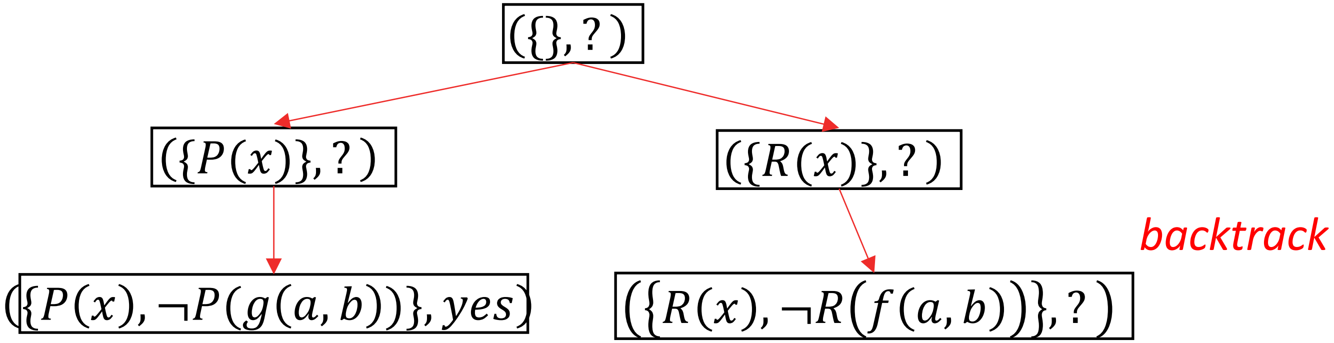
$$mgu = \{x \approx g(a,b), \boldsymbol{x} \approx \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{b})\}$$

UNIVERSITY OF CALGARY

Modelelimination (IV)

- Solve the following problem instances:

P(x) ∨ R(x), ¬R(f(a,b)), ¬P(g(a,b))

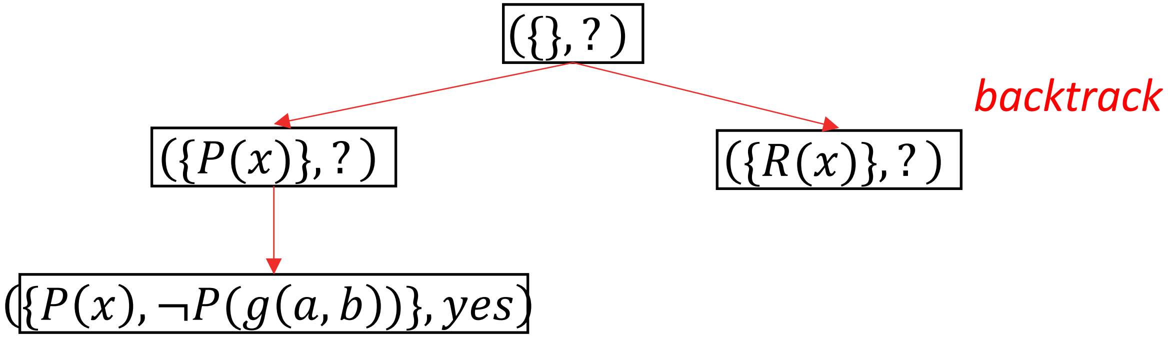


$$mgu = \{x \approx g(a,b), x \approx f(a,b)\}$$

Modelelimination (IV)

- Solve the following problem instances:

$P(x) \vee R(x), \neg R(f(a,b)), \neg P(g(a,b))$



$$mgu = \{x \approx g(a,b)\}$$

Modelelimination (IV)

- Solve the following problem instances:

$P(x) \vee R(x), \neg R(f(a,b)), \neg P(g(a,b))$

$\boxed{(\{\}, ?)}$

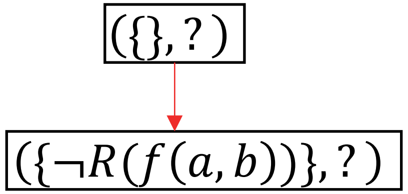
$mgu = \{\}$

UNIVERSITY OF CALGARY

Modelelimination (IV)

- Solve the following problem instances:

P(x) ∨ R(x), **¬R(f(a,b)),** ¬P(g(a,b))



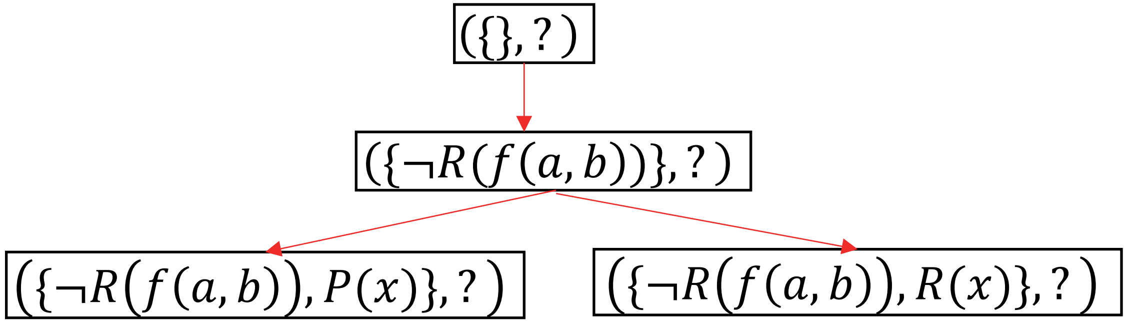
$mgu = \{\}$

UNIVERSITY OF CALGARY

Modelelimination (IV)

- Solve the following problem instances:

P(x) ∨ R(x), ¬R(f(a,b)), ¬P(g(a,b))

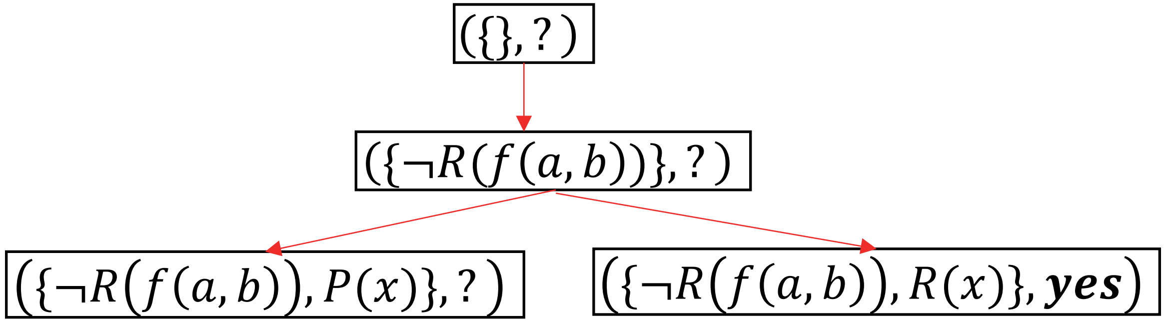


$mgu = \{\}$

Modelelimination (IV)

- Solve the following problem instances:

$P(x) \vee R(x), \neg R(f(a,b)), \neg P(g(a,b))$

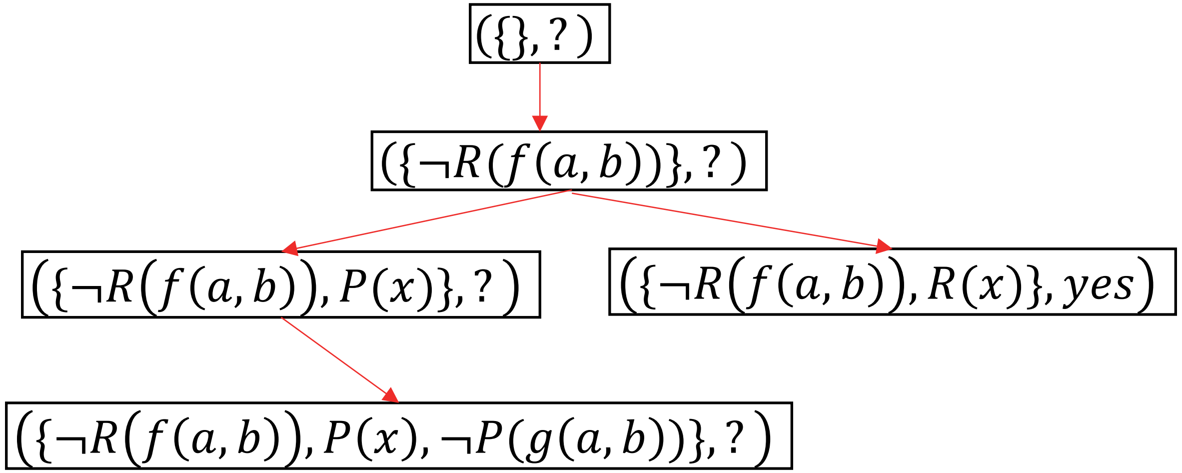


$$\boldsymbol{mgu = \{x \approx f(a,b)\}}$$

Modelelimination (IV)

- Solve the following problem instances:

P(x) ∨ R(x), ¬R(f(a,b)), **¬P(g(a,b))**

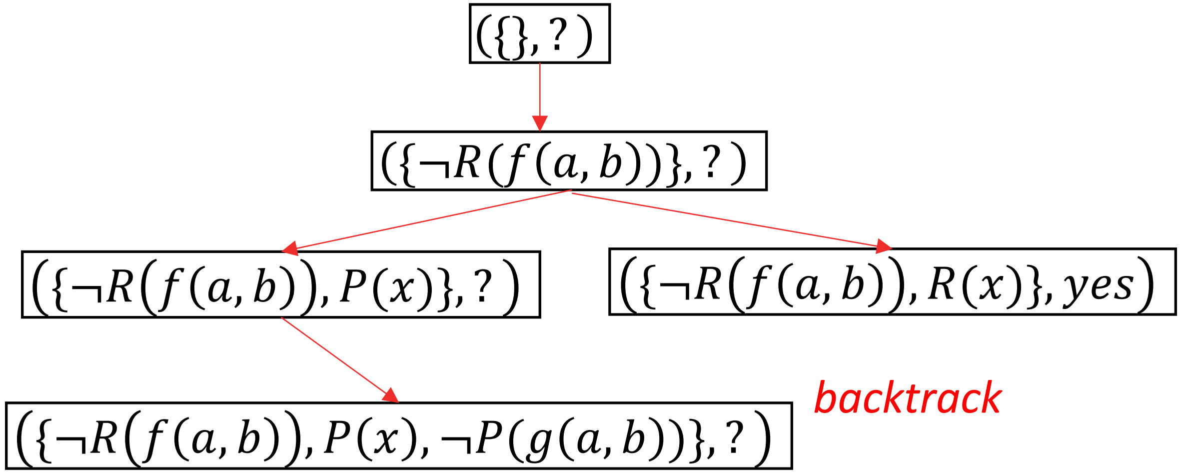


$mgu = \{x \approx f(a,b)\}$

Modelelimination (IV)

- Solve the following problem instances:

$P(x) \vee R(x), \neg R(f(a,b)), \neg P(g(a,b))$



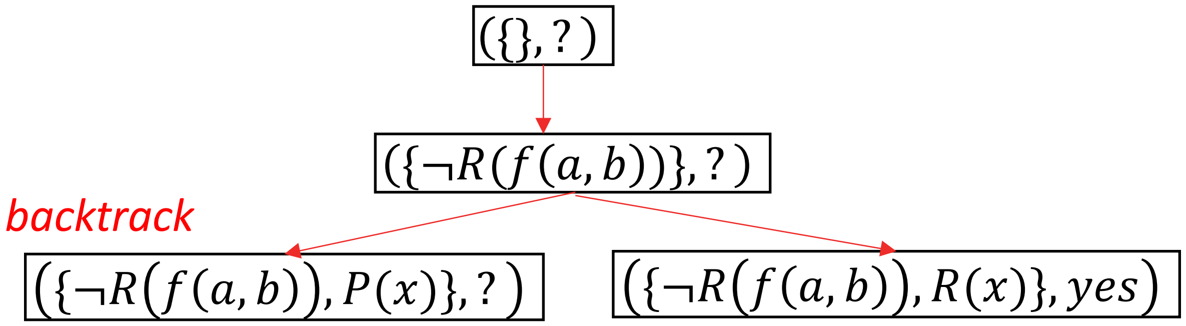
$mgu = \{x \approx f(a,b), x \approx g(a,b)\}$

UNIVERSITY OF CALGARY

Modelelimination (IV)

- Solve the following problem instances:

$P(x) \vee R(x), \neg R(f(a,b)), \neg P(g(a,b))$

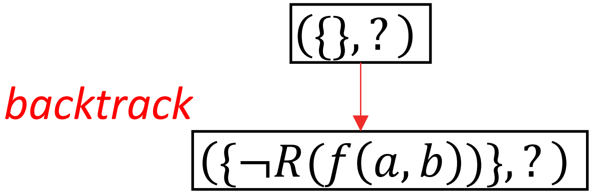


$$mgu = \{x \approx f(a,b))\}$$

Modelelimination (IV)

- Solve the following problem instances:

P(x) ∨ R(x), ¬R(f(a,b)), ¬P(g(a,b))



$mgu = \{\}$

UNIVERSITY OF CALGARY

Modelelimination (IV)

- Solve the following problem instances:

P(x) ∨ R(x), ¬R(f(a,b)), ¬P(g(a,b))

$(\{\}, ?)$

$mgu = \{\}$

UNIVERSITY OF CALGARY

Modelelimination (IV)

- Solve the following problem instances:

P(x) ∨ R(x), ¬R(f(a,b)), **¬P(g(a,b))**

$$(\{\}, ?)$$

$$\downarrow$$

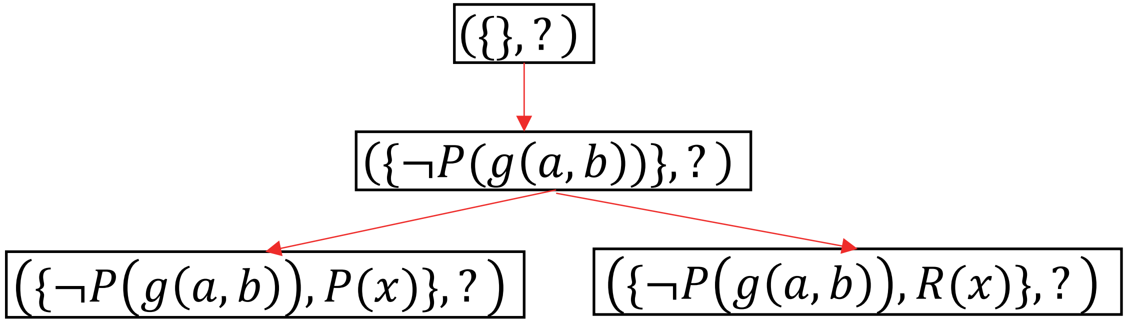
$$(\{\neg P(g(a,b))\}, ?)$$

$$mgu = \{\}$$

Modelelimination (IV)

- Solve the following problem instances:

P(x) ∨ R(x), ¬R(f(a,b)), ¬P(g(a,b))

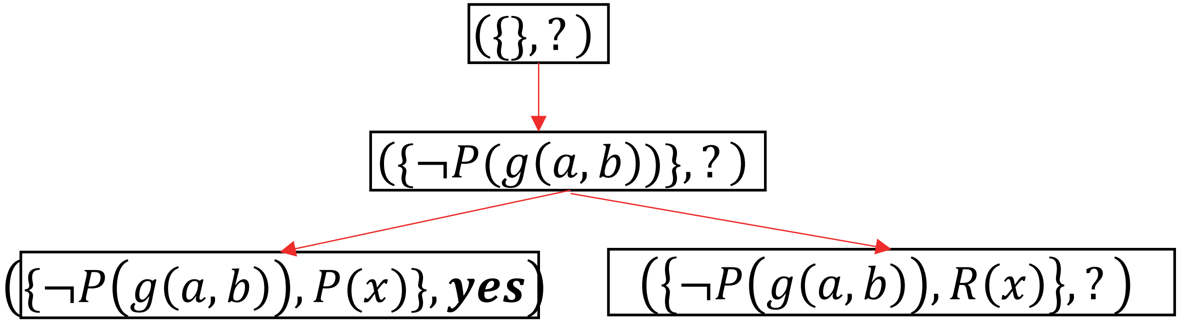


$mgu = \{\}$

Modelelimination (IV)

- Solve the following problem instances:

$P(x) \vee R(x), \neg R(f(a,b)), \neg P(g(a,b))$

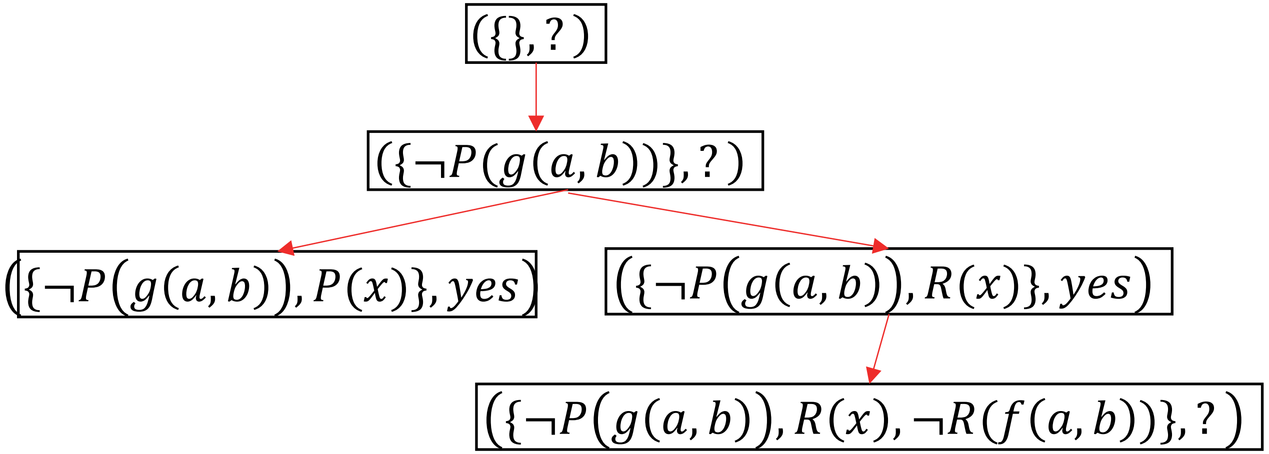


$$\boldsymbol{mgu = \{x \approx g(a,b)\}}$$

Modelelimination (IV)

- Solve the following problem instances:

$P(x) \vee R(x)$, $\mathbf{\neg R(f(a,b))}$, $\neg P(g(a,b))$

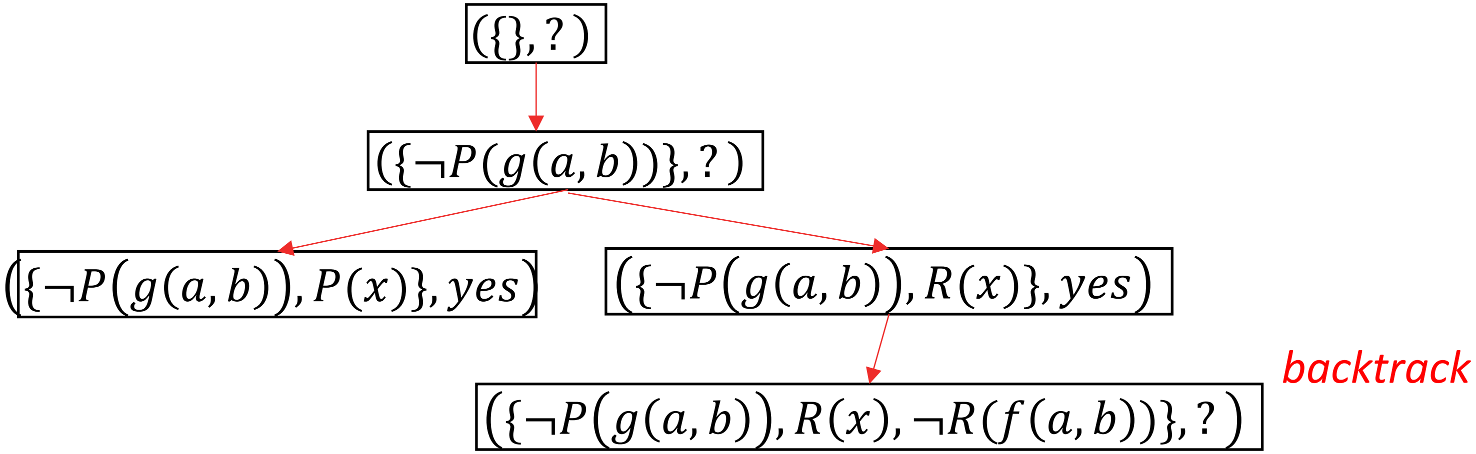


$$mgu = \{x \approx g(a,b)\}$$

Modelelimination (IV)

- Solve the following problem instances:

$P(x) \vee R(x), \neg R(f(a,b)), \neg P(g(a,b))$



$mgu = \{x \approx g(a,b), x \approx f(a,b)\}$

Modelelimination (IV)

- Solve the following problem instances:

$P(x) \vee R(x), \neg R(f(a,b)), \neg P(g(a,b))$

$\boxed{(\{\}, ?)}$

$mgu = \{\}$

UNIVERSITY OF CALGARY

Summary

Modelelimination (IV)

- Solve the following problem instances:

1) $p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q$ **success**
2) $p, q, \neg q$ **success**
3) $P(x) \lor R(x), \neg R\big(f(a,b)\big), \neg P(g(a,b))$ **failure**

Model-elimination: And-Tree-Based

Model-elimination (IV)

Tasks:

- Describe Model-elimination as and-tree-based search model
- Describe formally a search control for your model that uses backtracking to avoid generating an infinite branch in the tree representing the state (if the problem instance is solvable)

UNIVERSITY OF CALGARY

Model?

Model-elimination (IV)

Describe Model-elimination as and-tree-based search model

- We have set of Clauses $C = \{c_1, \dots, c_p\}$ of p clauses where is clause $c_i \in C$ is of form $c_i = L_1 \vee \cdots \vee L_n$(disjunction of literals) so will define a set all literals $L_{all} = \{L_j \mid L_j \text{ from } c_i\ \forall c_i \in C\}$ (set of all literals present in C)

- $\boldsymbol{Prob} = \{pr_1, \dots, pr_m\}$ where a $pr_i \in \boldsymbol{Prob}$ is
  - $pr_i \in 2^{L_{all}}$
  - (a single problem is some subset of L_j parts or $\boldsymbol{Prob} = 2^{L_{all}}$)

UNIVERSITY OF CALGARY

Model-elimination (IV)

Describe Model-elimination as and-tree-based search model

- $\boldsymbol{Div}$ will be defined by the relationship that if $pr \in \boldsymbol{Prob}$ is selected to divide into sub-problems then based a choice of $c_i \in C$ where $c_i = L_1 \vee \cdots \vee L_n$ then n sub-problems are created where each sub-problem pr_j fulfills
  - $pr_j = pr \cup L_j$
  - (each sub-problem j is a combination of the existing set of literals with the j^{th} literal)
  - If we want to avoid infinite divisions me might also add that one pr_j must be created such that the L_j being added is such that $\neg L_j \in pr$. We are eliminating one model sub-branch already (unless pr = {} at root)

Process?

Model-elimination (IV)

Describe formally a search control for your model

$\boldsymbol{f_{leaf}} =$

1. 0 if (pr,?) contains P and ¬P′ such that there is a σ with σ(P) ≡ σ(P′) $(tie\ break\ by\ <_{Lit})$
2. $|pr|$ otherwise $(tie\ break\ by\ <_{Lit})$

$\boldsymbol{f_{trans}} =$

1. (pr, **yes**) if (pr,?) contains P and ¬P′ such that there is a σ with σ(P) ≡ σ(P′)
2. if out of unique $c_i \in C$ for more ***Div*** or fail unfication then backtrack (and remove backtracked $c_j \in C$ from future consideration for ***Div*** at that leaf)
3. select $c_i \in C$ that has most negations (tie break by $<_{Lit}$) for **Div**

UNIVERSITY OF CALGARY

Remarks

Remarks

- There are many optimization problems that can be solved by an **and-tree**-based search without backtracking!
- Backtracking is often used to reduce the memory needs for a search (it allows to store only one path of the tree).
- Backtracking can always be avoided by using **and-or-tree**-based search.
- Branch-and-bound, dynamic programming and a lot of other algorithm schemes are **and-tree**-based search! (Think about how standard code/functions work using a stack frame to store history!)

Onward to …
or-tree-based search

Jonathan Hudson, Ph.D.
jwhudson@ucalgary.ca
https://cspages.ucalgary.ca/~jwhudson/

UNIVERSITY OF CALGARY