Horn Clauses and SLD Resolution

Marco Piastra

Horn Clauses (in L_P)

- Definition

  A ***Horn Clause*** is a wff in CF
  that contains at most <u>one</u> literal in positive form

- Three types of *Horn Clauses*:

  Rule: two or more literals, one positive

  Examples: $\{B, \neg D, \neg A, \neg C\}$, $\{A, \neg B\}$ (equivalent to: $(D \wedge A \wedge C) \rightarrow B$, $B \rightarrow A$)

  Facts: just one positive literal

  Examples: $\{B\}$, $\{A\}$

  Goal: one or more literals, all negative

  Examples: $\{\neg B\}$, $\{\neg A, \neg B\}$

  More terminology:

  Rules and facts are also called *definite clauses*

  Goals are allo called *negative clauses*

Lost in Translation…

Many wffs can be translated into Horn clauses:

$(A \wedge B) \rightarrow C$
$\neg(A \wedge B) \vee C$ (rewriting $\rightarrow$)
$\neg A \vee \neg B \vee C$ (De Morgan - CF – it is a rule)

$A \rightarrow (B \wedge C)$
$\neg A \vee (B \wedge C)$ (rewriting $\rightarrow$)
$(\neg A \vee B) \wedge (\neg A \vee C)$ (distributing $\vee$)
$(\neg A \vee B), (\neg A \vee C)$ (CF – two rules)

$(A \vee B) \rightarrow C$
$\neg(A \vee B) \vee C$ (rewriting $\rightarrow$)
$(\neg A \wedge \neg B) \vee C$ (De Morgan)
$(\neg A \vee C) \wedge (\neg B \vee C)$ (distributing $\vee$)
$(\neg A \vee C), (\neg B \vee C)$ (CF – two rules)

But not all of them:

$(A \wedge \neg B) \rightarrow C$
$\neg(A \wedge \neg B) \vee C$ (rewriting $\rightarrow$)
$\neg A \vee B \vee C$ (De Morgan)

$A \rightarrow (B \vee C)$
$\neg A \vee B \vee C$ (rewriting $\rightarrow$)

SLD Resolution (in L_P)

Linear resolution with Selection function for Definite clauses

- Algorithm

  Starts from a set of *definite clauses* (also the *program*) + a *goal*

  1) At each step, the *selection function* identifies a *literal* in the *goal* (i.e. *subgoal*)
  2) All *definite clause* applicable to the *subgoal* is selected
  3) The resolution rule is applied generating the *resolvent*

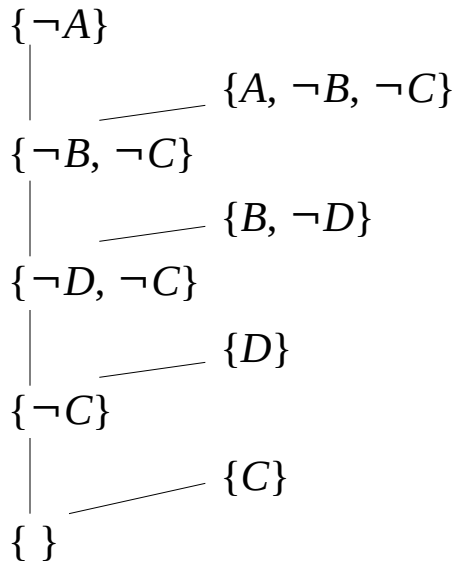
  Termination: either the empty clause { } is obtained or step 2) fails.

Example:

Selection function: ***leftmost subgoal first***

Definite clauses: $\{C\}$, $\{D\}$, $\{B, \neg D\}$, $\{A, \neg B, \neg C\}$

Goal: $\{\neg A\}$



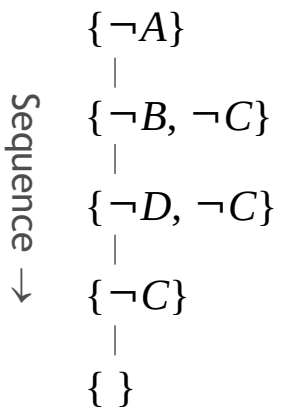
SLD trees (in L_P)

SLD derivations

Example: $\{C\}$, $\{D\}$, $\{B, \neg D\}$, $\{A, \neg B, \neg C\}$ goal $\{\neg A\}$

In this example each subgoal can be resolved in one mode only

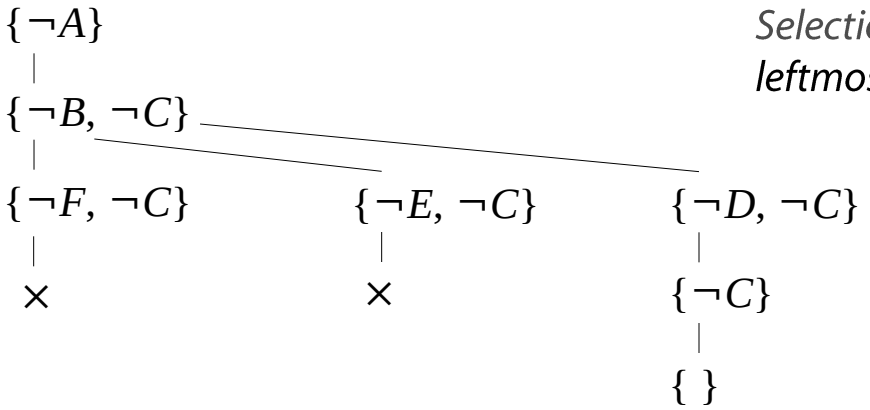
This is not true in general



- SLD trees (= trace of all SLD derivations from a goal)

  Example: $\{C\}$, $\{D\}$, $\{B, \neg F\}$, $\{B, \neg E\}$, $\{B, \neg D\}$, $\{A, \neg B, \neg C\}$ goal $\{\neg A\}$

  A few new rules have been added: there are now different possibilities



Selection function:
leftmost subgoal first

Each branch correspond to a possible resolution for a *subgoal*

SLD Resolution (in L_P)

- A resolution method <u>for Horn clauses</u> in L_P

  It always terminates

  It is *correct*: $\Gamma \vdash \varphi \Rightarrow \Gamma \models \varphi$

  It is *complete*: $\Gamma \models \varphi \Rightarrow \Gamma \vdash \varphi$

- Computationally efficient

  It has polynomial time complexity (w.r.t the # of propositional symbols occurring in Γ and φ)

- Limitations

  Not all problems can be translated into Horn clauses

  The "Harry is happy" problem does not translate

  Γ : only a set of *rules* and *facts*

  φ : only a conjunction of *facts*

Horn Clauses in L_{FO}

The definition is very similar to the propositional case

- Horn Clauses (of the skolemization of a set *sentences*)

  Each clause contains at most one literal in positive form

  Facts, rules and goals

  Fact: a clause with just an individual *atom*

  $\{Human(socrates)\}$, $\{Pyramid(x)\}$, $\{Sister(sally, motherOf(paul))\}$

  Rule: a clause with at least two literals, exactly one in positive form

  $\{Human(x), \neg Philosopher(x)\}$,
  $\forall x\ (Philospher(x) \rightarrow Human(x))$

  $\{\neg Female(x), \neg Parent(k(x),x), \neg Parent(k(y),y), Sister(x,y)\}$
  $\forall x \forall y\ ((Female(x) \wedge \exists z\ (Parent(z,x) \wedge Parent(z,y))) \rightarrow Sister(x,y))$

  $\{\neg Above(x,y), On(x,k(x))\}$, $\{\neg Above(x,y), On(j(y),y)\}$
  $\forall x \forall y\ (Above(x,y) \rightarrow (\exists z\ On(x,z) \wedge \exists v\ On(v,y)))$

  Goal: a clause containing negative literals only

  $\{\neg Human(socrates)\}$

  $\{\neg Sister(sally,x), \neg Sister(x,paul)\}$
  Negation of $\exists x\ (Sorella(sally,x) \wedge Sorella(x,paul))$

SLD Resolution in L_{FO}

Linear resolution with Selection function for Definite clauses

- Description

  Program (a set of *definite clauses:* rules + facts):

  Rule: $\beta \vee \neg\gamma_1 \vee \neg\gamma_2 \vee \ldots \vee \neg\gamma_n$

  Fact: δ

  Goal (a conjunction of facts in negated form:

  Goal: $\neg\alpha_1 \vee \neg\alpha_2 \vee \ldots \vee \neg\alpha_k$

  Procedure:

  - Starting point: a program Π and a goal ϕ
  - The subgoals are considered according to the *selection function* of choice
  - For each subgoal $\neg\alpha_i$ the resolution (with unification) is attempted with <u>all</u> rules and facts in Π whose positive literal is compatible

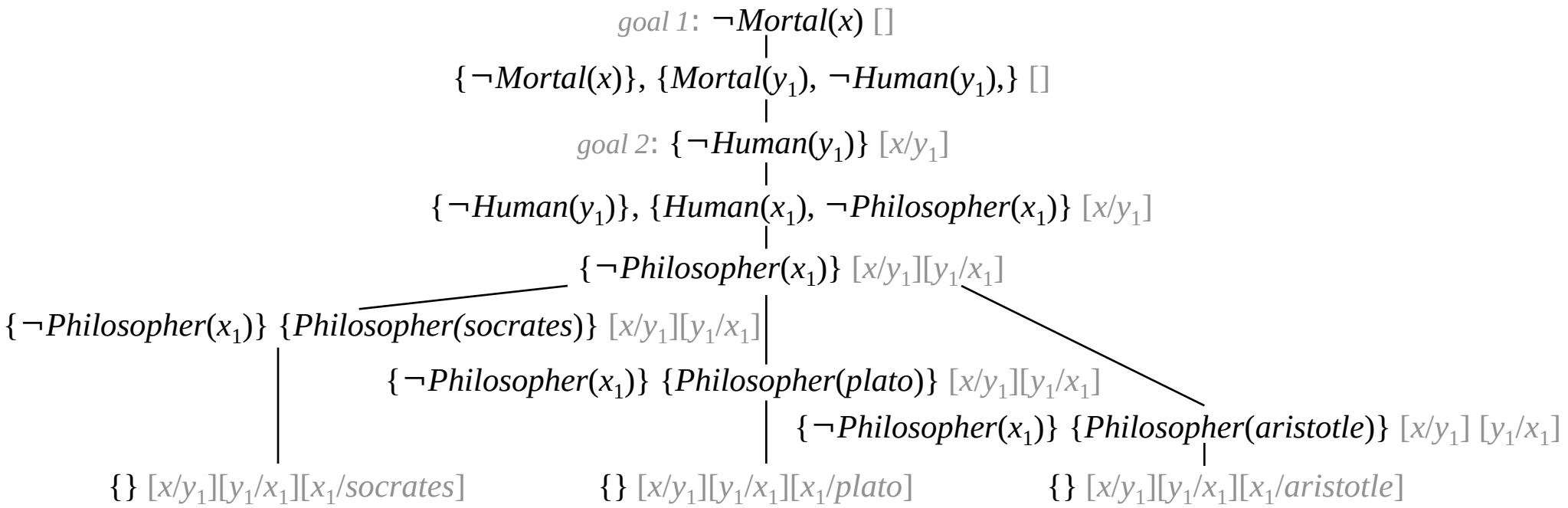
SLD Trees

- Example:

$\Pi \equiv \{\{Human(x), \neg Philosopher(x)\}, \{Mortal(y), \neg Human(y)\},$
$\{Philosopher(socrates)\}, \{Philosopher(plato)\}, \{Philosopher(aristotle)\}\}$

$goal \equiv \{\neg Mortal(x), \neg Human(x)\}$

"Is there anyone who is both human and mortal?"



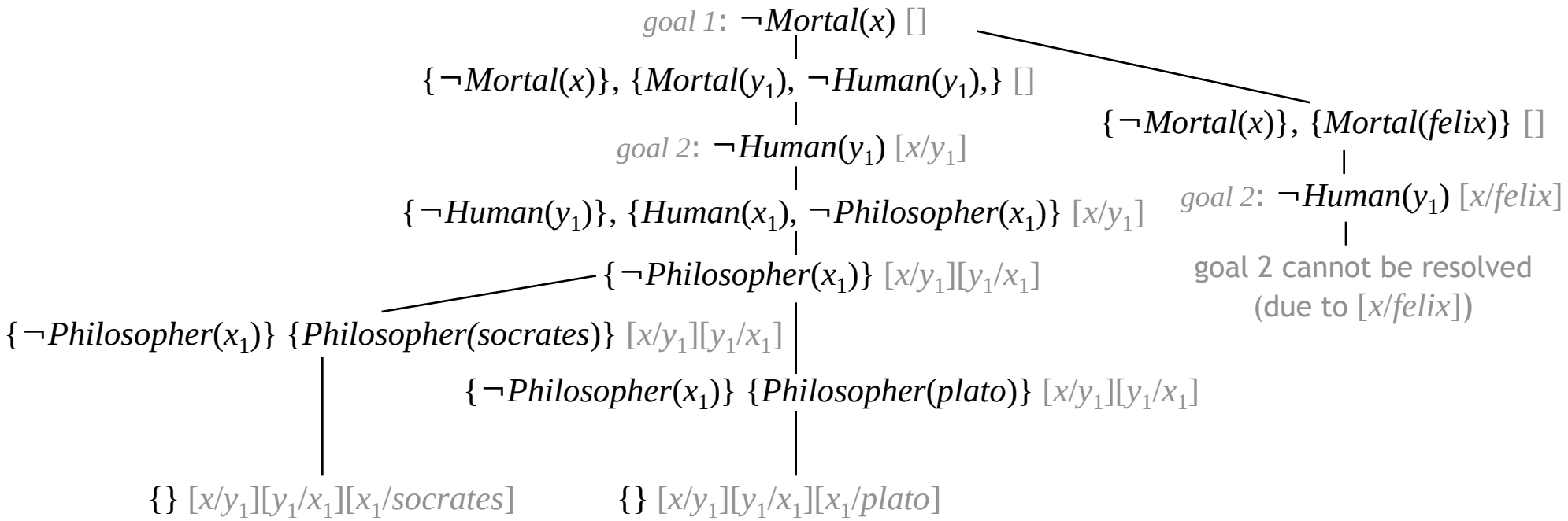
SLD Trees

- Another example

$\Pi \equiv \{\{Human(x), \neg Philosopher(x)\}, \{Mortal(y), \neg Human(y)\},$
$\{Philosopher(socrates)\}, \{Philosopher(plato)\}, \{Mortal(felix)\}\}$

$goal \equiv \{\neg Mortal(x), \neg Human(x)\}$

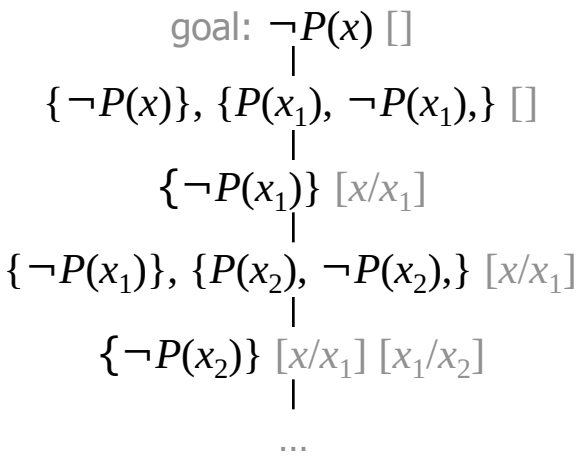
"Is there anyone who is both human and mortal?"



Infinite SLD Trees

- A first example:

$$\Pi \equiv \{\{P(x), \neg P(x)\}\}$$
$$\neg\phi \equiv \{\neg P(x)\}$$

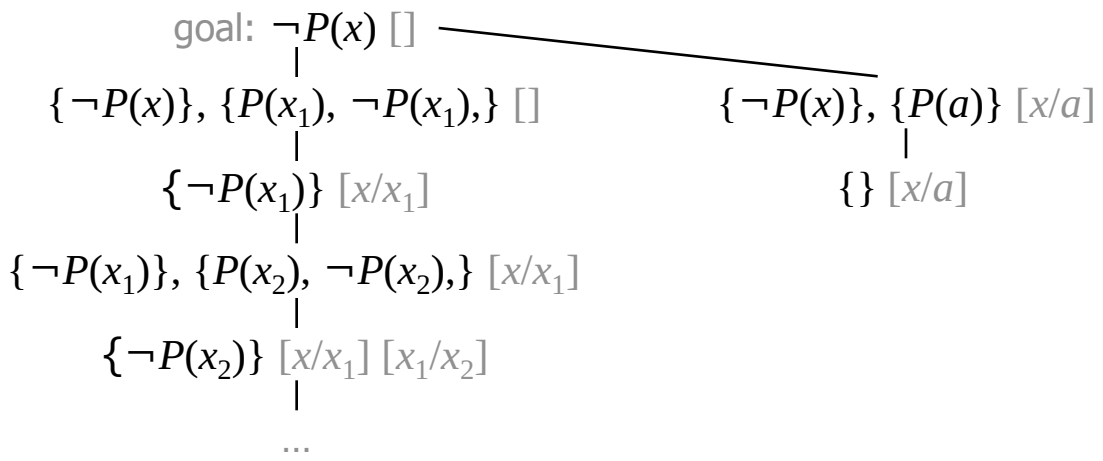


Since $\Pi \not\models \phi$, the method can *diverge* (and it does…)

Infinite SLD Trees

- A second example:

$$\Pi \equiv \{\{P(x), \neg P(x)\}, \{P(a)\}\}$$
$$\neg\phi \equiv \{\neg P(x)\}$$



In this case $\Pi \models \phi$, so the method should *not* diverge.

However, when a *depth-first* selection function is used, the infinite branch in the SLD-tree makes the method diverge anyway.

A ***fair*** selection function is such that no possible resolution will be postponed indefinitely: that is, any possible resolution will be performed, eventually.