The Foundations of Inductive Logic Programming

By Chongbing Liu

Outlines

- Resolution Based Proof Procedures
- ILP Problem Specification
- Generality Orders on Clauses
- Refinement Operators
- Conclusions

Resolution Based Proof Procedures

Proof Procedures

- Very often we need to prove that $\Sigma \models E$
- But this is in general undecidable
- When $\Sigma \models E$ is true, we could have some procedures to generate proofs
- Ideal properties: complete, sound,

  work mechanically, efficient and applicable to all Σ and E

Resolution Based Proof Procedures

Resolution

resolution rule

C=f,x   D= ¬f,y

R=x ∨ y

Resolvent

Σ: C, D, E, F, G, H

R, U, V, C, D, E, F, G, H

W, R, U, V, C, D, E, F, G, H

......

C, D from Σ or all the intermediate resolvents → **Unconstrained**

Incomplete
Inefficient

Resolution Based Proof Procedures

Resolution

resolution rule

C=f,x   D= ¬f,y

R=x ∨ y

Resolvent

Incompleteness Example

Σ:
h(x)←f(x),g(x).
f(x).
g(x).

resolution

Σ ⊨ h(x).

Σ ⊨ h(a).

C, D from Σ or all the intermediate resolvents

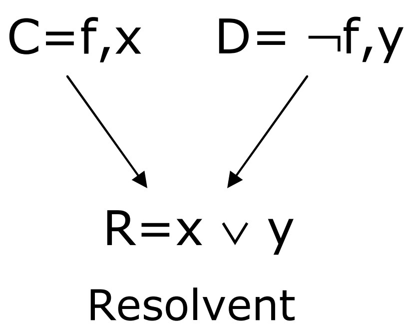
Unconstrained

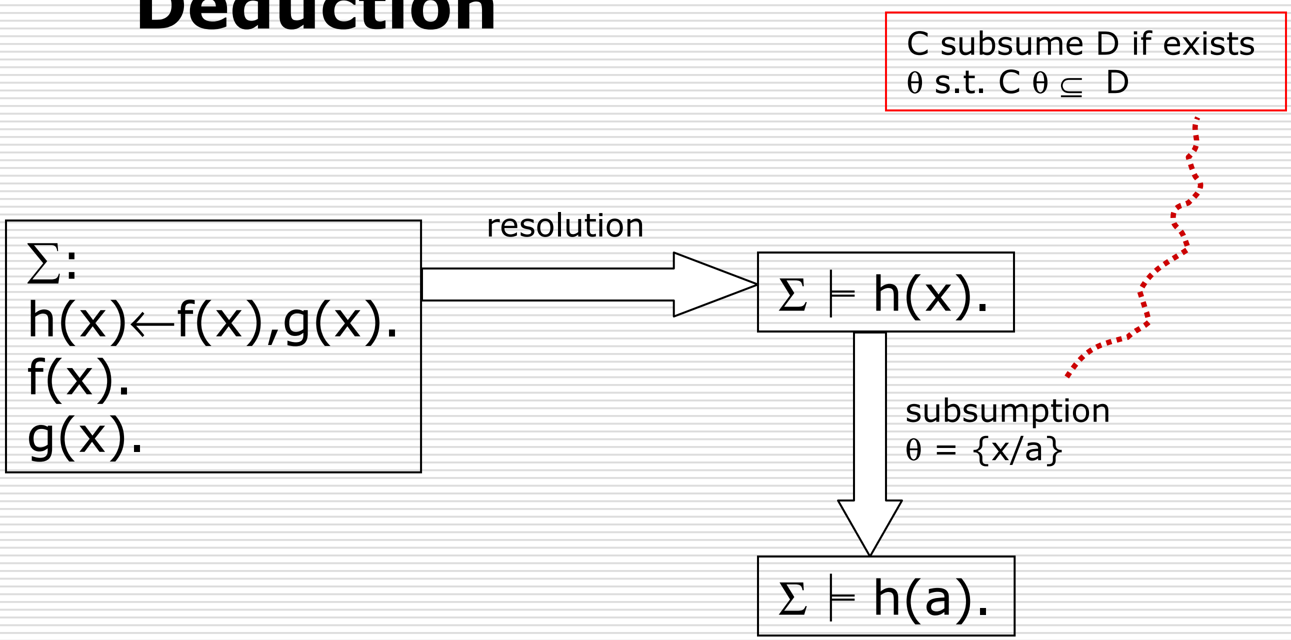
Incomplete
Inefficient

Resolution Based Proof Procedures

Deduction

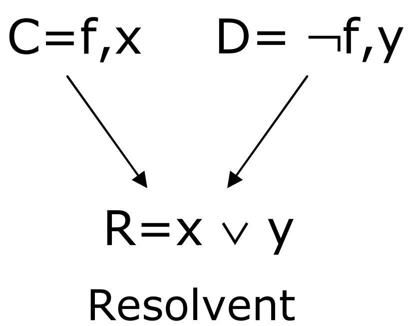
resolution rule



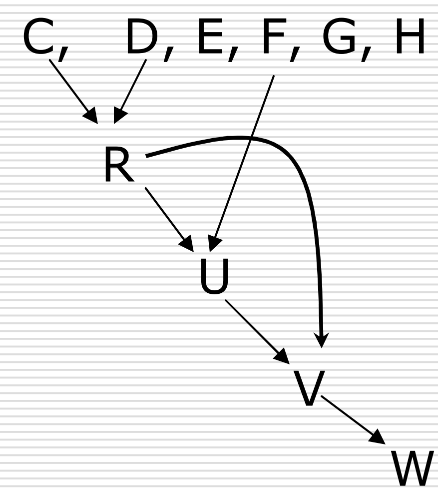


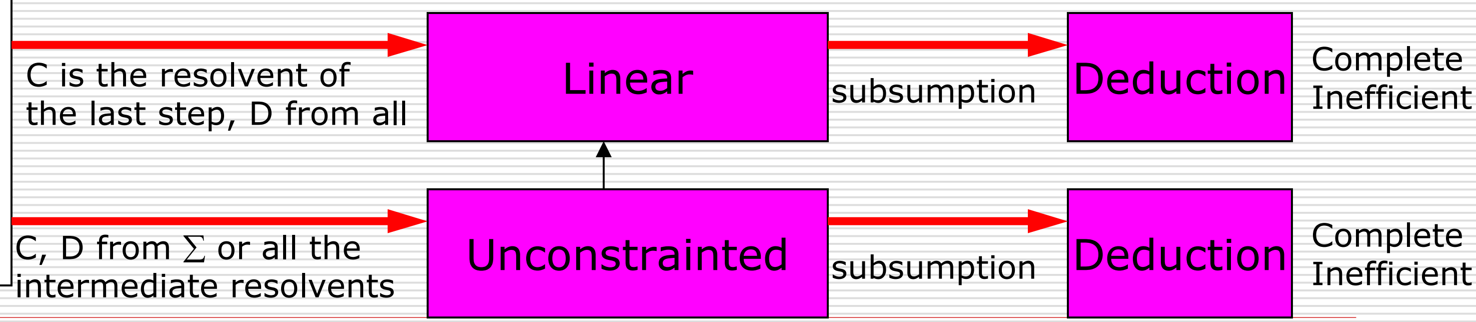
Resolution Based Proof Procedures

resolution rule

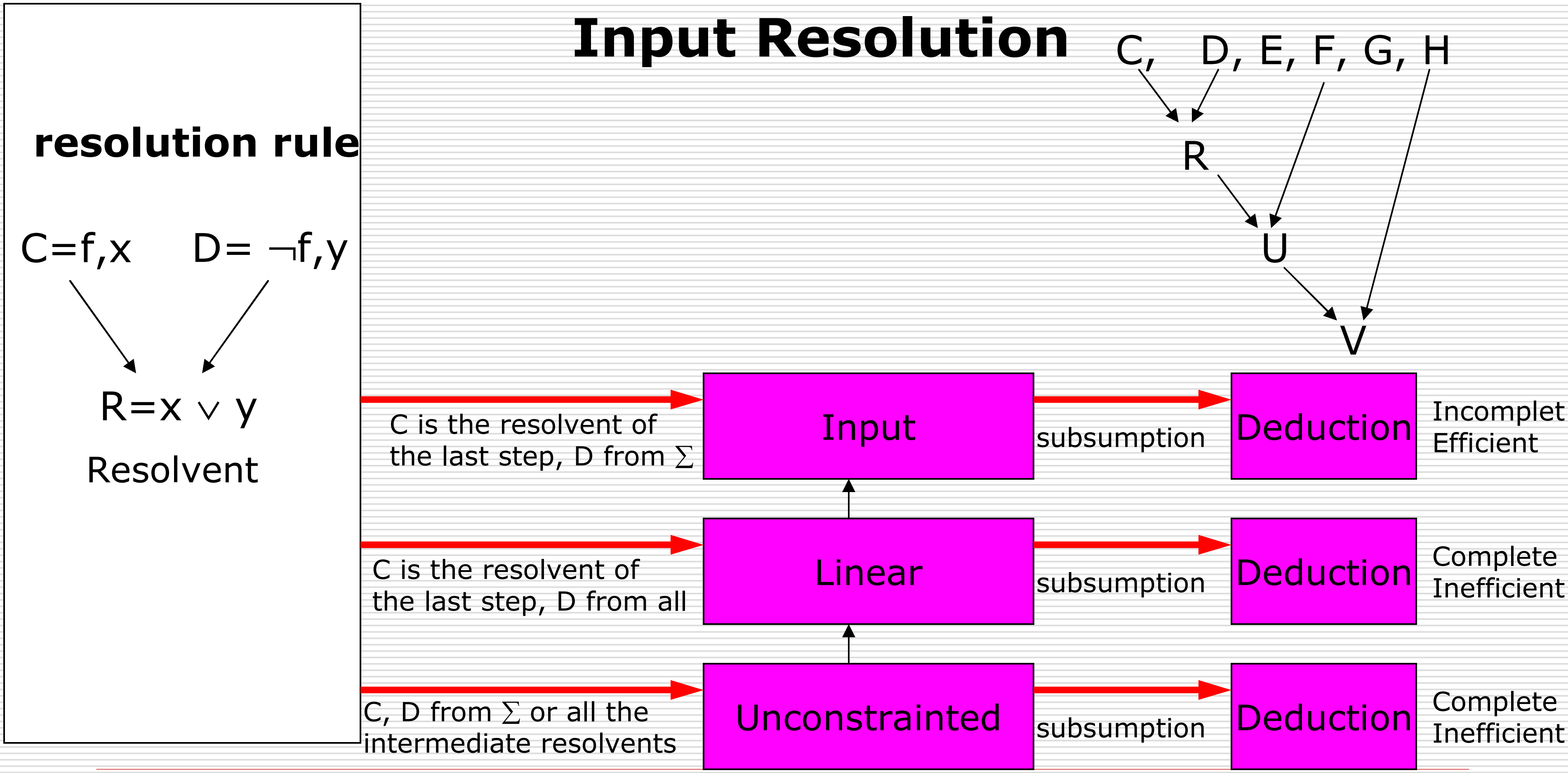


Linear Resolution



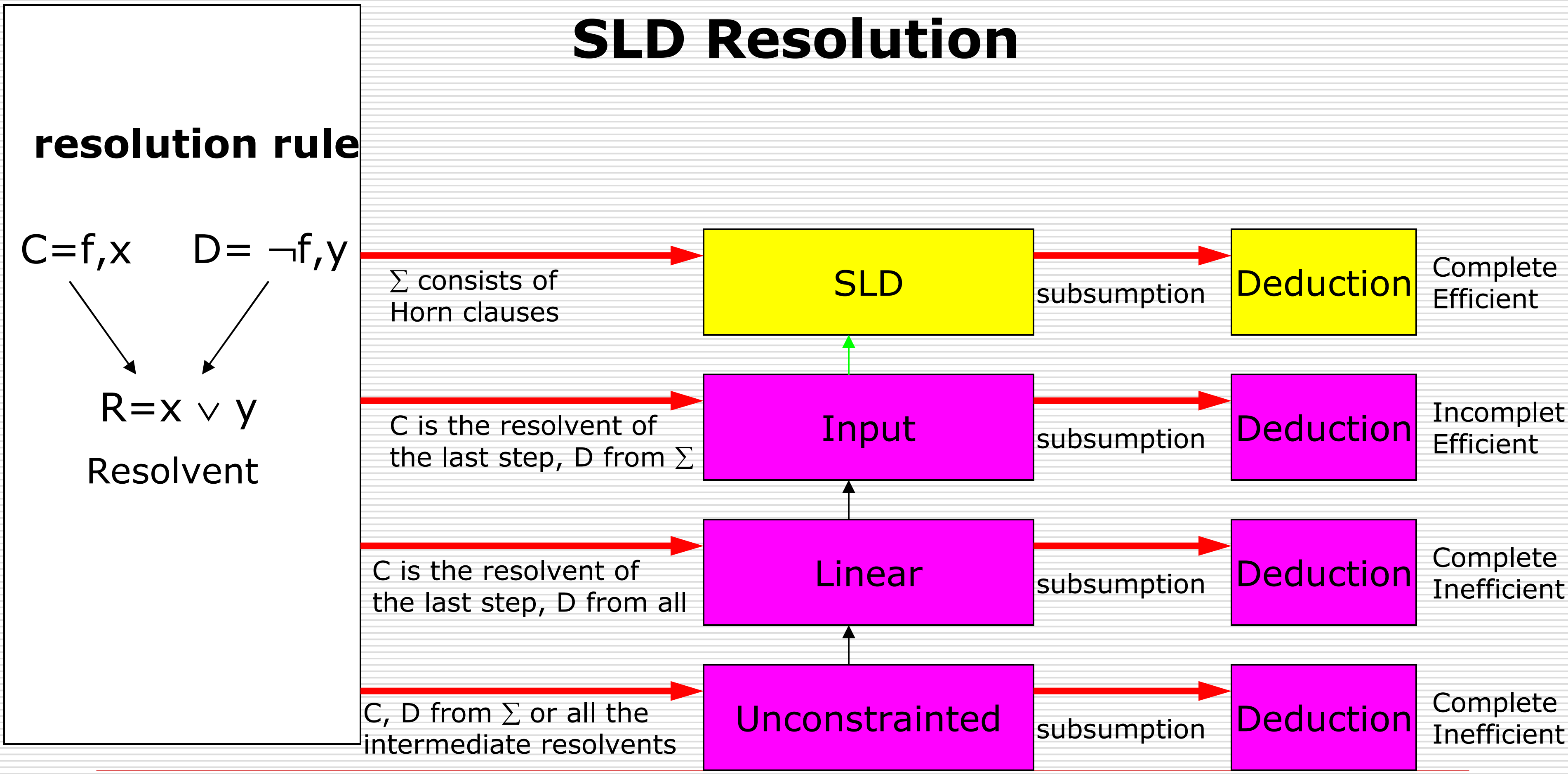


Resolution Based Proof Procedures

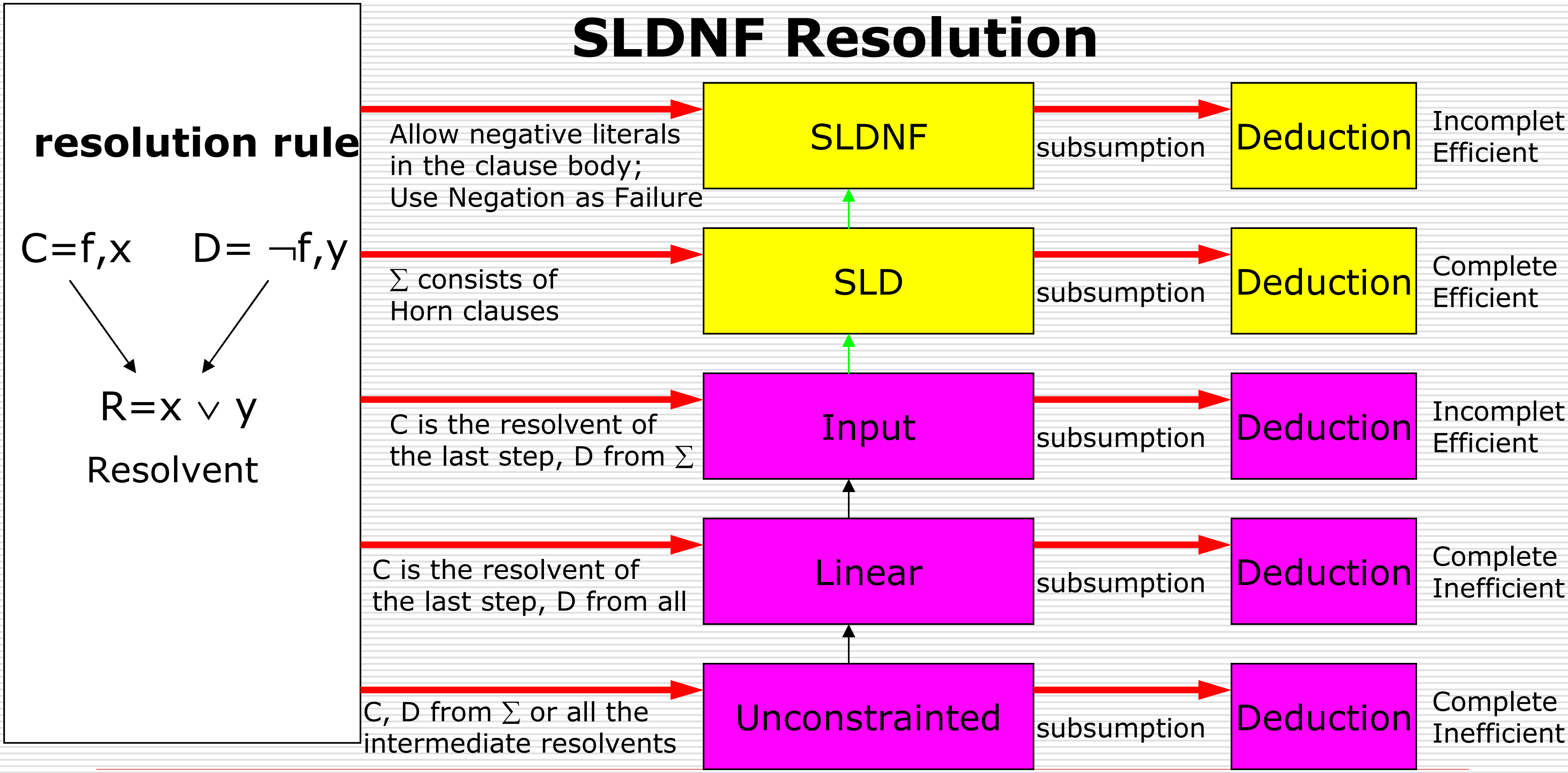


Resolution Based Proof Procedures

SLD Resolution



Resolution Based Proof Procedures



ILP Problem Specification

Given:

A finite set of clauses B (background knowledge),

and sets of clauses E^+ and E^-

Find:

A theory Σ , such that $\Sigma \cup B$ is correct with respect to E^+ and E^-

ILP Problem Specification

Correct theory

$\Sigma \cup B$ is correct with respect to E^+ and E^- if

1. $\Sigma \cup B \models E^+$ (*completeness*)

   and

2. $\Sigma \cup B \cup \neg E^-$ is satisfiable (*consistency*).

ILP Search all the clauses for correct Σ

ILP Problem Specification

Correct theory

$\Sigma \cup B$ is correct with respect to E^+ and E^- if

1. $\Sigma \cup B \models E^+$ (*completeness*)

   and

2. $\underline{\Sigma \cup B \cup \neg E^- \text{ is satisfiable}}$ (*consistency*).

⇕ ?

$(\Sigma \cup B)$ implies no $e \in E^-$ (easier, proof procedures)

ILP Problem Specification

Consistency Condition

$\Sigma \cup B$ is correct with respect to E^+ and E^- if

1. $\Sigma \cup B \models E^+$ (*completeness*)

   and

2. $\Sigma \cup B \cup \neg E^-$ is satisfiable (*consistency*).

$(\Sigma \cup B)$ implies no $e \in E^-$

(easier, proof procedures)

Example: (let $B=\varnothing$)

Σ = { P(a) ∨ P(b) }

E^- = { P(a), P(b) }

ILP Problem Specification

Admissibility

$\Sigma \cup B$ is correct with respect to E^+ and E^- if

1. $\Sigma \cup B \models E^+$ (*completeness*)

   and

2. $\Sigma \cup B \cup \neg E^-$ is satisfiable (*consistency*).

$\Updownarrow$

If $\langle \mathbf{E}, \Sigma \rangle$ are admissible:

$\langle$ground atoms, Horn clauses$\rangle$

$\langle$ground literals, clauses$\rangle$

$(\Sigma \cup B)$ implies no $e \in E^-$ (easier, proof procedures)

ILP Problem Specification

Correct theory

$\Sigma \cup B$ is correct with respect to E^+ and E^- if

1. $\Sigma \cup B \models E^+$ (*completeness*)

   and

If $\langle \mathbf{E}, \Sigma \rangle$ are admissible:

$\langle$ground atoms, Horn clauses$\rangle$

$\langle$ground literals, clauses$\rangle$

2. $(\Sigma \cup B)$ implies no $e \in E^-$ (*consistency*). (easier, proof procedures)

ILP Problem Specification

Correct theory

$\Sigma \cup B$ is correct with respect to E^+ and E^- if

1. $\Sigma \cup B \models E^+$ (*completeness*)

   and

Reduced Search Space ! (bias)

If $\langle$**E**, Σ $\rangle$ are admissible:

$\langle$ground atoms, Horn clauses$\rangle$

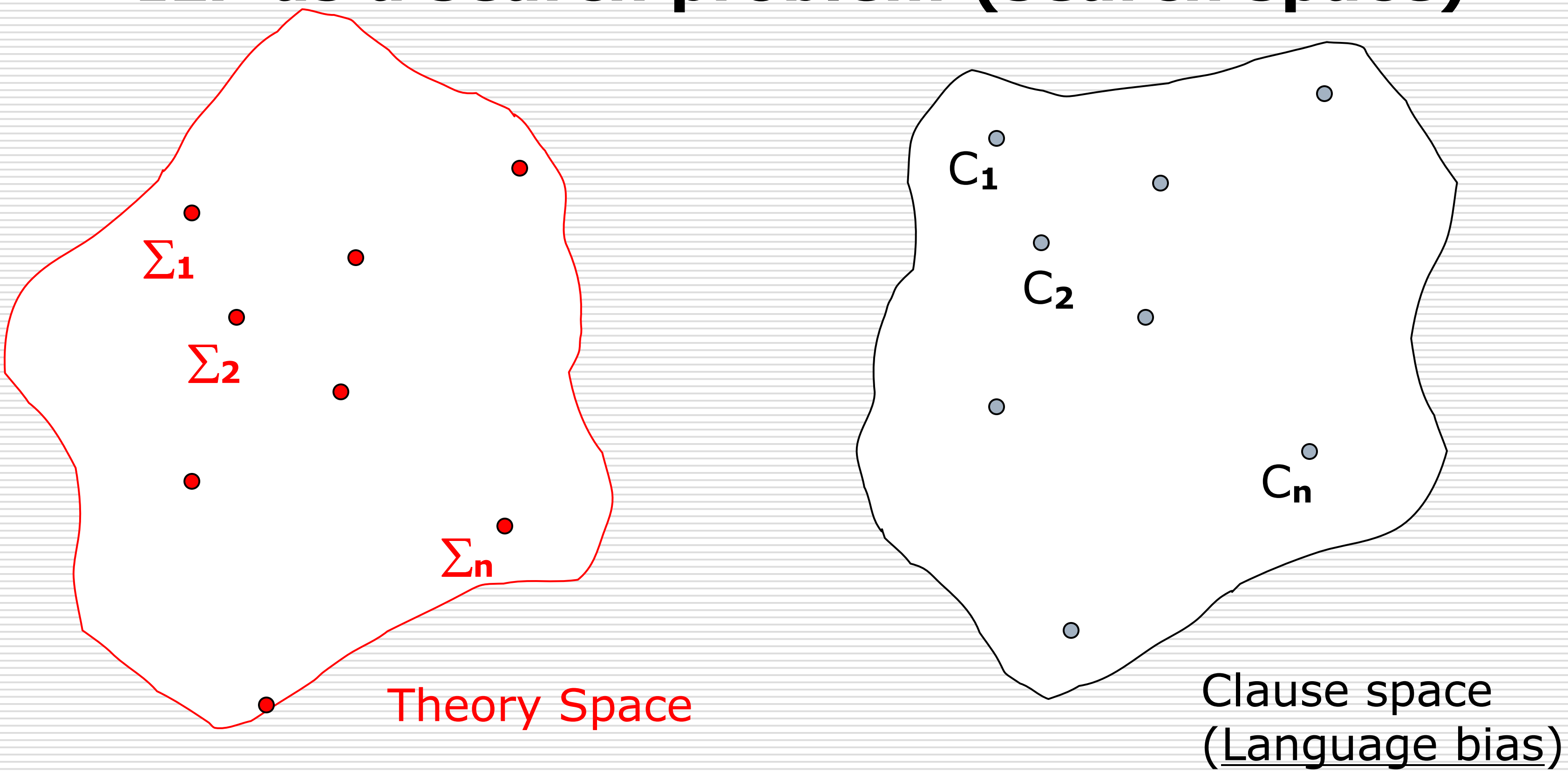
$\langle$ground literals, clauses$\rangle$

2. $(\Sigma \cup B)$ implies no $e \in E^-$ (*consistency*).

(easier, proof procedures)

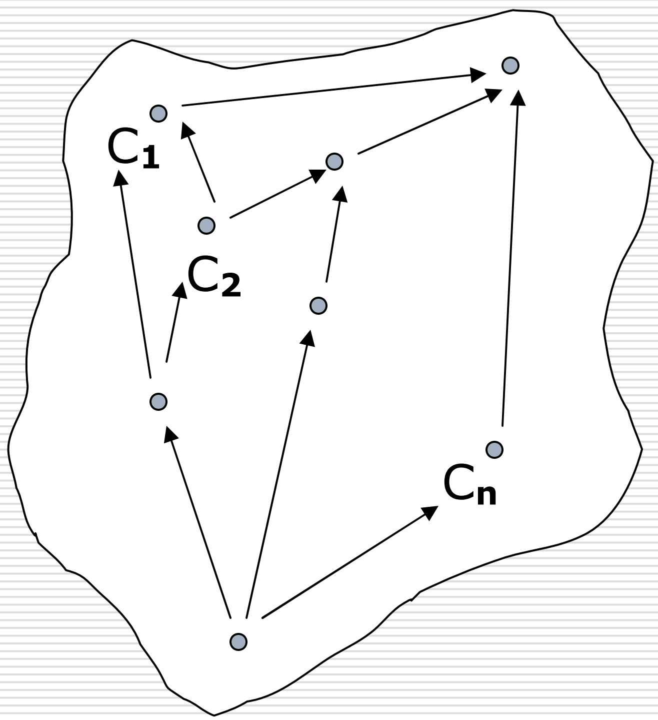
ILP Problem Specification

ILP as a search problem (search space)



ILP Problem Specification

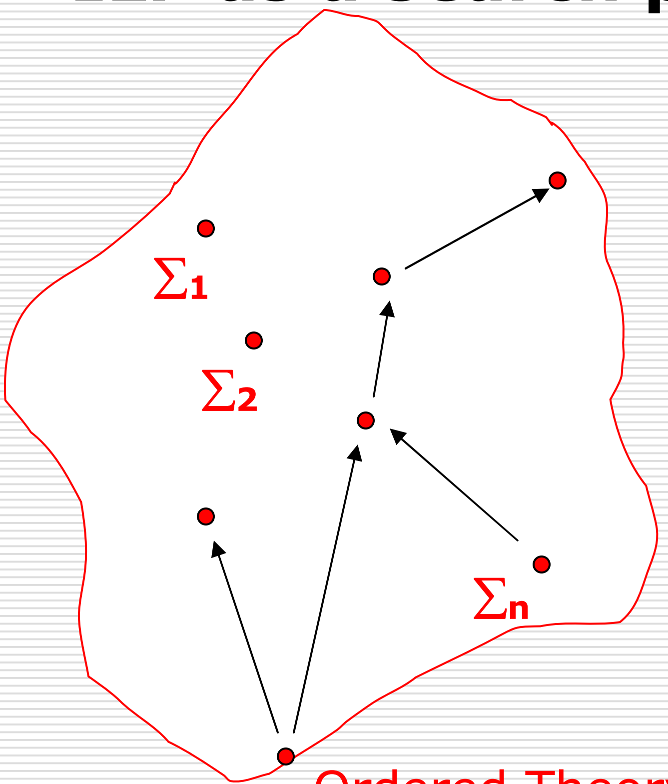
ILP as a search problem (generality orders)

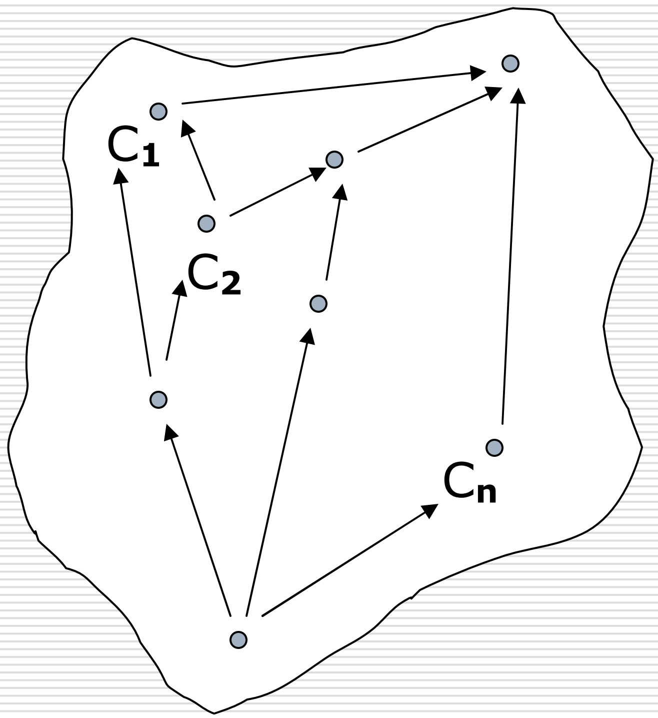


Ordered Clause space

ILP Problem Specification

ILP as a search problem (generality orders)





Ordered Theory Space

Ordered Clause space

ILP Problem Specification

ILP as a search problem (A General Scheme)

Start with some initial theory

Repeat

If Σ is too strong, specialize it

If Σ is too weak, generalize it

until $\Sigma \cup B$ is correct with respect to E^+ and E^-

ILP Problem Specification

Operations

Start with some initial theory

Repeat

If Σ is too strong, specialize it

If Σ is too weak, generalize it

} Refinement operators

until $\Sigma \cup B$ is correct with respect to E^{+} and E^{-}

Generality Orders on Clauses

Basic Concepts

- Quasi-order $\geqslant$ on set S: **Reflexive and transitive**
- Least generalization(S): **Least Upper Bound (lub)**
- Greatest specialization(S): **Greatest Lower Bound (glb)**
- Lattice: **Exist lub and glb for any S**
- Downward Cover(C): **$\{ D \mid C \geqslant D$, and no E s.t. $C > E > D \}$**
- Upward Cover(C): **$\{ D \mid D \geqslant C$, and no E s.t. $D > E > C \}$**

Generality Orders on Clauses

(no background knowledge)

- Subsumption order on atoms
- Subsumption order on clauses
- Implication order on clauses

Generality Orders on Clauses

(no background knowledge)

Subsumption order ($\propto$) on the set of **atoms**

- Definition : A $\propto$ B if A$\theta \subseteq$ B for some θ
- Existence Of Least Generalization : Yes
- Existence Of Greatest Specialization: Yes
- Upward covers : finite
- Downward cover: finite

Generality Orders on Clauses

(no background knowledge)

<u>Subsumption order ($\propto$) on the set of **clauses**</u>

- Definition : $A \propto B$ if $A\theta \subseteq B$ for some θ
- Existence Of Least Generalization : Yes
- Existence Of Greatest Specialization: Yes
- On Horn clauses : Lattice
- Upward covers : not always exist or finite
- Downward cover : not always exist or finite

Generality Orders on Clauses

(no background knowledge)

Implication order ($\models$) on the set of **clauses**

- Definition : logical consequence
- Existence Of Least Generalization :
- Existence Of Greatest Specialization: Yes
- On Horn clauses : NO
- Upward covers : not always exist or finite
- Downward cover : not always exist or finite

Only when S contains at least
One function-free clause

Generality Orders on Clauses

(with background knowledge)

- Relative Subsumption order
- Relative Implication order
- Generalized Subsumption order

Generality Orders on Clauses

(with background knowledge)

Relative Subsumption order ($\propto_B$)

- Definition: C $\propto_B$ D if **B** $\models \forall(C\theta \subseteq D)$ for some θ
- Existence Of Least Gneralization: Yes, when **B** is a set of ground literals
- On Horn clauses : Yes, when **B** is ground atoms
- Deduction : Exist a deduction of D from $\{C\} \cup$ **B** where C occurs at most once

Generality Orders on Clauses

(with background knowledge)

Relative Implication order ($\models_{\mathbf{B}}$)

- Definition: C $\models_{\mathbf{B}}$ D if ($\mathbf{B} \cup \{C\}$) $\models$ D
- Existence Of Least Generalization: Yes, when **B** is a set of function-free ground literals and S contains at least on function-free clause
- On Horn clauses : NO
- Deduction : Exist a deduction of D from $\{C\} \cup \mathbf{B}$

Generality Orders on Clauses

(with background knowledge)

Generalized Subsumption order ($\geqslant_B$)

- Definition: $C \geqslant_B D$ if with **B,** C can be used to prove at least as many results as D
- Existence Of Least Generalization: Yes, but if S is a set of atoms, or S and **B** are all function-free or **B** is ground
- On Horn clauses : Yes, e.g., if **B** is ground definite program and S is a set of definite program clause with same heads
- Deduction : Exist a SLD-deduction of D, where C is the top clause and members in **B** are input clauses

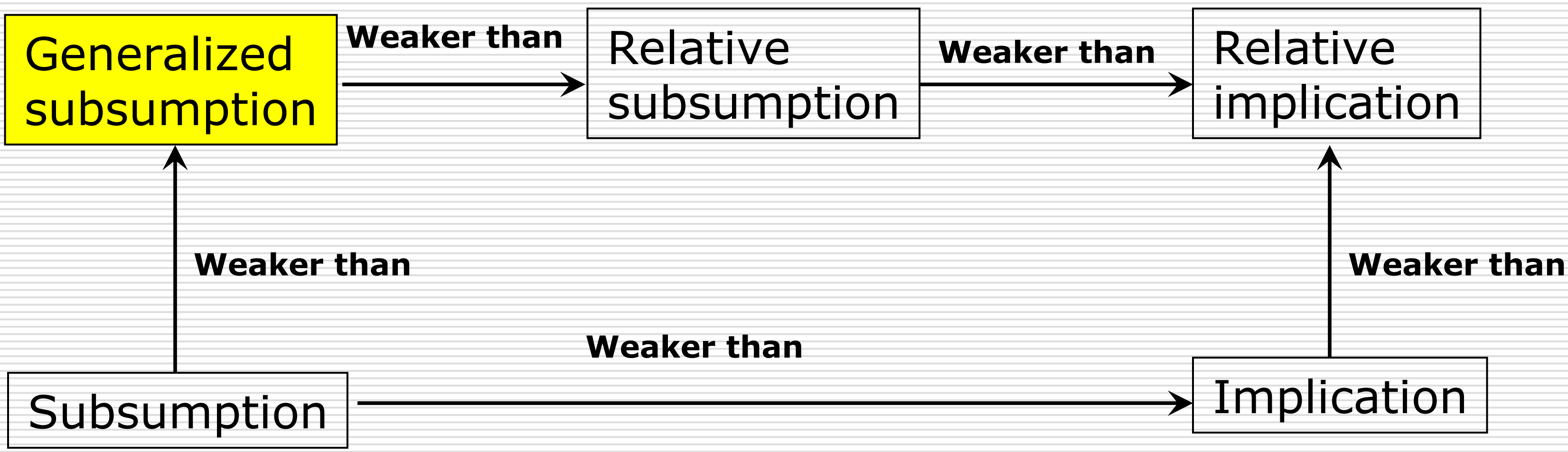
Generality Orders on Clauses

(with background knowledge)

- $C \geqslant_B D$ if there exists a SLD-deduction of D, with C as top clause and members of B as input clauses.
- $C \propto_B D$ if there exists a deduction of D from $\{C\} \cup B$ where C occurs at most once as a leaf.
- $C \models_B D$ if there exists a deduction of D from $\{C\} \cup B$.

Generality Orders on Clauses

summary



Refinement Operators

- functions from a single clause to a set of clauses:

  $\rho(C)$: downward refinement operators

  $\delta(C)$: upward refinement operators

- Ideal (downward) operators:

  Locally finite : $\rho(C)$ is finite

  Complete : $\forall\ C > D,\ \exists\ E \in \rho^*(c)$ s.t. $D \approx E$

  Proper : $\rho(C) \subseteq \{\ D \mid C > D\ \}$

Refinement Operators

- Ideal $\rho(C)$ exists $\Leftrightarrow$

  every C has a finite set of downward cover set
- Ideal $\delta(C)$ exists $\Leftrightarrow$

  every C has a finite set of upward cover set
- Only subsumption order on set of atoms has finite downward and upward cover sets. Others don't.
- So ideal operators do not exist for clauses structured by most practical orders.

Refinement Operators

- In practice we drop the properness, and use
- locally finite and complete operators.

- Such operators exist for clauses structured by
- subsumption order.

Conclusions

- Resolution based proof procedures are useful in ILP.
- ILP is a search problem.
- Different orders may be defined on the search space.
- The search could be achieve by applying refinement operators.

Thank you.