UNIWERSYTET WARSZAWSKI

Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski

Artificial Intelligence in Theorem Proving

(Sztuczna inteligencja w dowodzeniu)

Cezary Kaliszyk

MIMUW

Overview

Last Lecture

- How do we characterize the prover / proof state?
- Syntactic and semantic features
- Unification-based features
- ML-evaluation

Today

- How does learning work in different prover foundations?
- First-order ATPs foundations

Different Prover Foundations

FOL, HOL, ...

Set Theory

- sets and membership
- semantic information
- "collections of things"
- membership is undecidable
- extensional; talk about things that exist

Type Theory

- typing judgement
- syntactic information
- what objects can be constructed
- intentional
- type checking (and sometimes inference) is decidable

Typed λ-calculus as a foundation

Basis for a Proof Assistant

- Terms: Programs and Proofs
- Types: Specifications and Formulas

Brings together

- Programming
- Proving

Extensions

- Dependent Types
- Dependent Propositions
- Dependency on Proofs
- Reflection ...

Typed set theory (Mizar)

- x comes from a certain domain
  - We should have a unique type for any object
- Proper Frankel operator:

$$\{x : D | P(x)\} \text{ or } \{x \in D | P(x)\}$$

Definition [membership]

$$P(y) \iff y \in \{x : D | P(x)\}$$

Notation

$$\{x : A | P(x)\} \text{ means } \{x : D | x \in A \wedge P(x)\}$$

Features, Labels and Training examples

In FOL / HOL (also set theory)

- Proved theorems $\rightarrow$ Training examples
- Dependencies (including theorems) $\rightarrow$ Labels
- Characteristics of conjectures (eg: constants) $\rightarrow$ Features

Type Theory: Types, Terms, Thms merged

- Each defined constant consists of a term and a type
- Features are the constants present in the type
- Dependencies are the constants present in the term

Approaches

- Do we need to?
- Only `Prop` is insufficient
- Heuristic

How does an ATP work

Proof by contradiction

- Assume that the conjecture does not hold
- Derive that axioms and negated conjecture imply $\bot$

Saturation

- Convert problem to CNF (conjunctive normal form)
- Enumerate the consequences of the available clauses
- Goal: get to the empty clause

Redundancies

Simplify or eliminate some clauses (contract)

We will next focus on the foundations needed to transform the problem to CNF

Quantifier Equivalences

Theorem

$$\begin{array}{rcl} \neg\forall x\,\phi & \dashv\vdash & \exists x\,\neg\phi \\ \forall x\,\phi \wedge \forall x\,\psi & \dashv\vdash & \forall x\,(\phi \wedge \psi) \\ \forall x\,\forall y\,\phi & \dashv\vdash & \forall y\,\forall x\,\phi \end{array} \qquad \begin{array}{rcl} \neg\exists x\,\phi & \dashv\vdash & \forall x\,\neg\phi \\ \exists x\,\phi \vee \exists x\,\psi & \dashv\vdash & \exists x\,(\phi \vee \psi) \\ \exists x\,\exists y\,\phi & \dashv\vdash & \exists y\,\exists x\,\phi \end{array}$$

if x is not free in ψ then

$$\begin{array}{rcl} \forall x\,\phi \wedge \psi & \dashv\vdash & \forall x\,(\phi \wedge \psi) \\ \exists x\,\phi \wedge \psi & \dashv\vdash & \exists x\,(\phi \wedge \psi) \\ \forall x\,(\psi \to \phi) & \dashv\vdash & \psi \to \forall x\,\phi \\ \exists x\,(\psi \to \phi) & \dashv\vdash & \psi \to \exists x\,\phi \end{array} \qquad \begin{array}{rcl} \forall x\,\phi \vee \psi & \dashv\vdash & \forall x\,(\phi \vee \psi) \\ \exists x\,\phi \vee \psi & \dashv\vdash & \exists x\,(\phi \vee \psi) \\ \forall x\,(\phi \to \psi) & \dashv\vdash & \exists x\,\phi \to \psi \\ \exists x\,(\phi \to \psi) & \dashv\vdash & \forall x\,\phi \to \psi \end{array}$$

Skolemization

Definitions

- prenex normal form is predicate logic formula

$$Q_1 x_1 \; Q_2 x_2 \ldots Q_n x_n \; \phi$$

with $Q_i \in \{\forall, \exists\}$ and ϕ quantifier free

Skolemization

Definitions

- prenex normal form is predicate logic formula

$$Q_1 x_1 \; Q_2 x_2 \ldots Q_n x_n \; \phi$$

  with $Q_i \in \{\forall, \exists\}$ and ϕ quantifier free

- Skolem normal form is closed (no free variables) prenex normal form

$$\forall x_1 \; \forall x_2 \ldots \forall x_n \; \phi$$

  with ϕ in CNF

Skolemization

Definitions

- prenex normal form is predicate logic formula

$$Q_1 x_1 \; Q_2 x_2 \ldots Q_n x_n \; \phi$$

  with $Q_i \in \{\forall, \exists\}$ and ϕ quantifier free

- Skolem normal form is closed (no free variables) prenex normal form

$$\forall x_1 \; \forall x_2 \ldots \forall x_n \; \phi$$

  with ϕ in CNF

Example

$$\forall x \, \forall y \, ((P(f(x)) \vee \neg P(g(y)) \vee Q(g(y))) \wedge (\neg Q(g(y)) \vee \neg P(g(y)) \vee Q(g(x))))$$

Skolemization

Definitions

- prenex normal form is predicate logic formula

$$Q_1 x_1 \; Q_2 x_2 \ldots Q_n x_n \; \phi$$

  with $Q_i \in \{\forall, \exists\}$ and ϕ quantifier free

- Skolem normal form is closed (no free variables) prenex normal form

$$\forall x_1 \; \forall x_2 \ldots \forall x_n \; \phi$$

  with ϕ in CNF

Example

$$\forall x \, \forall y \, ((P(f(x)) \vee \neg P(g(y)) \vee Q(g(y))) \wedge (\neg Q(g(y)) \vee \neg P(g(y)) \vee Q(g(x))))$$

$$\{\{P(f(x)), \neg P(g(y)), Q(g(y))\}, \{\neg Q(g(y)), \neg P(g(y)), Q(g(x))\}\}$$

Theorem

$\forall$ *formula* ϕ $\exists$ *prenex normal form* ψ *such that* $\phi \equiv \psi$

Theorem

$\forall$ *formula* ϕ $\exists$ *prenex normal form* ψ *such that* $\phi \equiv \psi$

Proof.

1. rename all bound variables such that every quantifier binds unique variable (which does not occur free)

Theorem

$\forall$ *formula* ϕ $\exists$ *prenex normal form* ψ *such that* $\phi \equiv \psi$

Proof.

1. rename all bound variables such that every quantifier binds unique variable (which does not occur free)
2. push logical connectives through quantifiers:

$$\begin{array}{cc}
\neg\forall x\, \phi \equiv \exists x\, \neg\phi & \neg\exists x\, \phi \equiv \forall x\, \neg\phi \\
\forall x\, \phi \wedge \psi \equiv \forall x\, (\phi \wedge \psi) & \phi \wedge \forall x\, \psi \equiv \forall x\, (\phi \wedge \psi) \\
\exists x\, \phi \wedge \psi \equiv \exists x\, (\phi \wedge \psi) & \phi \wedge \exists x\, \psi \equiv \exists x\, (\phi \wedge \psi) \\
\forall x\, \phi \vee \psi \equiv \forall x\, (\phi \vee \psi) & \phi \vee \forall x\, \psi \equiv \forall x\, (\phi \vee \psi) \\
\exists x\, \phi \vee \psi \equiv \exists x\, (\phi \vee \psi) & \phi \vee \exists x\, \psi \equiv \exists x\, (\phi \vee \psi) \\
\forall x\, \phi \rightarrow \psi \equiv \exists x\, (\phi \rightarrow \psi) & \phi \rightarrow \forall x\, \psi \equiv \forall x\, (\phi \rightarrow \psi) \\
\exists x\, \phi \rightarrow \psi \equiv \forall x\, (\phi \rightarrow \psi) & \phi \rightarrow \exists x\, \psi \equiv \exists x\, (\phi \rightarrow \psi)
\end{array}$$

Definition

formulas ϕ and ψ are equisatisfiable ($\phi \approx \psi$) if

$$\phi \text{ is satisfiable} \iff \psi \text{ is satisfiable}$$

Definition

formulas ϕ and ψ are equisatisfiable ($\phi \approx \psi$) if

$$\phi \text{ is satisfiable} \iff \psi \text{ is satisfiable}$$

Theorem

$\forall$ *sentence* ϕ $\exists$ *Skolem normal form* ψ *such that* $\phi \approx \psi$

Definition

formulas ϕ and ψ are equisatisfiable ($\phi \approx \psi$) if

$$\phi \text{ is satisfiable} \iff \psi \text{ is satisfiable}$$

Theorem

$\forall$ *sentence* ϕ $\exists$ *Skolem normal form* ψ *such that* $\phi \approx \psi$

Proof (Skolemization).

1. transform ϕ into closed prenex normal form $Q_1 x_1\ Q_2 x_2 \ldots Q_n x_n\ \chi$ with χ in CNF

Definition

formulas ϕ and ψ are equisatisfiable ($\phi \approx \psi$) if

$$\phi \text{ is satisfiable} \iff \psi \text{ is satisfiable}$$

Theorem

$\forall$ *sentence* ϕ $\exists$ *Skolem normal form* ψ *such that* $\phi \approx \psi$

Proof (Skolemization).

1. transform ϕ into closed prenex normal form $Q_1 x_1\ Q_2 x_2 \ldots Q_n x_n\ \chi$ with χ in CNF
2. repeatedly replace

$$\forall x_1\ \ldots \forall x_{i-1}\ \exists x_i\ Q_{i+1} x_{i+1}\ \ldots Q_n x_n\ \psi$$

by

$$\forall x_1\ \ldots \forall x_{i-1}\ Q_{i+1} x_{i+1}\ \ldots Q_n x_n\ \psi[f(x_1, \ldots, x_{i-1})/x_i]$$

where f is new function symbol of arity $i-1$

Examples

- $\forall z\ \exists x\ \exists y\ ((P(x) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(x) \vee \neg P(y) \vee Q(z)))$

Examples

- $\forall z\ \exists x\ \exists y\ ((P(x) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(x) \vee \neg P(y) \vee Q(z)))$

Examples

- $\forall z\ \exists x\ \exists y\ ((P(x) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(x) \vee \neg P(y) \vee Q(z)))$

  $[f(z)/x]$

Examples

- $\forall z\ \exists x\ \exists y\ ((P(x) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(x) \vee \neg P(y) \vee Q(z)))$

  $\approx\ [f(z)/x]$

  $\forall z\ \exists y\ ((P(f(z)) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(y) \vee Q(z)))$

Examples

- $\forall z\ \exists x\ \exists y\ ((P(x) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(x) \vee \neg P(y) \vee Q(z)))$

  $\approx\ [f(z)/x]$

  $\forall z\ \exists y\ ((P(f(z)) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(y) \vee Q(z)))$

Examples

- $\forall z\ \exists x\ \exists y\ ((P(x) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(x) \vee \neg P(y) \vee Q(z)))$

  $\approx\ [f(z)/x]$

  $\forall z\ \exists y\ ((P(f(z)) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(y) \vee Q(z)))$

  $[g(z)/y]$

Examples

- $\forall z\ \exists x\ \exists y\ ((P(x) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(x) \vee \neg P(y) \vee Q(z)))$

  $\approx\ [f(z)/x]$

  $\forall z\ \exists y\ ((P(f(z)) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(y) \vee Q(z)))$

  $\approx\ [g(z)/y]$

  $\forall z\ ((P(f(z)) \vee \neg P(g(z)) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(g(z)) \vee Q(z)))$

Examples

- $\forall z\, \exists x\, \exists y\, ((P(x) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(x) \vee \neg P(y) \vee Q(z)))$

  $\approx\ [f(z)/x]$

  $\forall z\, \exists y\, ((P(f(z)) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(y) \vee Q(z)))$

  $\approx\ [g(z)/y]$

  $\forall z\, ((P(f(z)) \vee \neg P(g(z)) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(g(z)) \vee Q(z)))$

- $\forall x\, \exists y\, (\exists z\, P(z) \wedge \forall u\, (Q(x,u) \rightarrow \exists v\, Q(y,v)))$

Examples

- $\forall z\, \exists x\, \exists y\, ((P(x) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(x) \vee \neg P(y) \vee Q(z)))$

  $\approx \quad [f(z)/x]$

  $\forall z\, \exists y\, ((P(f(z)) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(y) \vee Q(z)))$

  $\approx \quad [g(z)/y]$

  $\forall z\, ((P(f(z)) \vee \neg P(g(z)) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(g(z)) \vee Q(z)))$

- $\forall x\, \exists y\, (\exists z\, P(z) \wedge \forall u\, (Q(x,u) \rightarrow \exists v\, Q(y,v)))$

  $\equiv$

  $\forall x\, \exists y\, \exists z\, \forall u\, \exists v\, (P(z) \wedge (\neg Q(x,u) \vee Q(y,v)))$

Examples

- $\forall z\, \exists x\, \exists y\, ((P(x) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(x) \vee \neg P(y) \vee Q(z)))$

  $\approx\ [f(z)/x]$

  $\forall z\, \exists y\, ((P(f(z)) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(y) \vee Q(z)))$

  $\approx\ [g(z)/y]$

  $\forall z\, ((P(f(z)) \vee \neg P(g(z)) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(g(z)) \vee Q(z)))$

- $\forall x\, \exists y\, (\exists z\, P(z) \wedge \forall u\, (Q(x,u) \rightarrow \exists v\, Q(y,v)))$

  $\equiv$

  $\forall x\, \exists y\, \exists z\, \forall u\, \exists v\, (P(z) \wedge (\neg Q(x,u) \vee Q(y,v)))$

  $\approx\ [f(x)/y]\ [g(x)/z]$

  $\forall x\, \forall u\, \exists v\, (P(g(x)) \wedge (\neg Q(x,u) \vee Q(f(x),v)))$

Examples

- $\forall z\, \exists x\, \exists y\, ((P(x) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(x) \vee \neg P(y) \vee Q(z)))$

  $\approx\ [f(z)/x]$

  $\forall z\, \exists y\, ((P(f(z)) \vee \neg P(y) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(y) \vee Q(z)))$

  $\approx\ [g(z)/y]$

  $\forall z\, ((P(f(z)) \vee \neg P(g(z)) \vee Q(z)) \wedge (\neg Q(f(z)) \vee \neg P(g(z)) \vee Q(z)))$

- $\forall x\, \exists y\, (\exists z\, P(z) \wedge \forall u\, (Q(x,u) \rightarrow \exists v\, Q(y,v)))$

  $\equiv$

  $\forall x\, \exists y\, \exists z\, \forall u\, \exists v\, (P(z) \wedge (\neg Q(x,u) \vee Q(y,v)))$

  $\approx\ [f(x)/y]\ [g(x)/z]$

  $\forall x\, \forall u\, \exists v\, (P(g(x)) \wedge (\neg Q(x,u) \vee Q(f(x),v)))$

  $\approx\ [h(x,u)/v]$

  $\forall x\, \forall u\, (P(g(x)) \wedge (\neg Q(x,u) \vee Q(f(x),h(x,u))))$

Resolution (propositional)

Definition

- literal is atom p or negation of atom $\neg p$

Resolution (propositional)

Definition

- literal is atom p or negation of atom $\neg p$
- clause is set of literals $\{l_1, \ldots, l_n\}$, representing $l_1 \vee \cdots \vee l_n$

Resolution (propositional)

Definition

- literal is atom p or negation of atom $\neg p$
- clause is set of literals $\{l_1, \ldots, l_n\}$, representing $l_1 \vee \cdots \vee l_n$
- clausal form is set of clauses $\{C_1, \ldots, C_m\}$, representing $C_1 \wedge \cdots \wedge C_m$

Resolution (propositional)

Definition

- literal is atom p or negation of atom $\neg p$
- clause is set of literals $\{l_1, \ldots, l_n\}$, representing $l_1 \vee \cdots \vee l_n$
- clausal form is set of clauses $\{C_1, \ldots, C_m\}$, representing $C_1 \wedge \cdots \wedge C_m$
- literals l_1 and l_2 are complementary if $l_1 = l_2^c$

Resolution (propositional)

Definition

- literal is atom p or negation of atom $\neg p$
- clause is set of literals $\{l_1, \ldots, l_n\}$, representing $l_1 \vee \cdots \vee l_n$
- clausal form is set of clauses $\{C_1, \ldots, C_m\}$, representing $C_1 \wedge \cdots \wedge C_m$
- literals l_1 and l_2 are complementary if $l_1 = l_2^c$
- clauses C_1 and C_2 clash if $\exists$ literal l with $l \in C_1$ and $l^c \in C_2$

Resolution (propositional)

Definition

- literal is atom p or negation of atom $\neg p$
- clause is set of literals $\{l_1, \ldots, l_n\}$, representing $l_1 \vee \cdots \vee l_n$
- clausal form is set of clauses $\{C_1, \ldots, C_m\}$, representing $C_1 \wedge \cdots \wedge C_m$
- literals l_1 and l_2 are complementary if $l_1 = l_2^c$
- clauses C_1 and C_2 clash if $\exists$ literal l with $l \in C_1$ and $l^c \in C_2$
- resolvent of clashing clauses C_1 and C_2 is clause

$$(C_1 \setminus \{l\}) \cup (C_2 \setminus \{l^c\})$$

Resolution (predicate)

Definition

- literal is atom $P(t_1, \ldots, t_n)$ or negation of atom $\neg P(t_1, \ldots, t_n)$

Resolution (predicate)

Definition

- literal is atom $P(t_1, \ldots, t_n)$ or negation of atom $\neg P(t_1, \ldots, t_n)$
- clause is set of literals $\{l_1, \ldots, l_n\}$, representing $l_1 \vee \cdots \vee l_n$

Resolution (predicate)

Definition

- literal is atom $P(t_1, \ldots, t_n)$ or negation of atom $\neg P(t_1, \ldots, t_n)$
- clause is set of literals $\{l_1, \ldots, l_n\}$, representing $l_1 \vee \cdots \vee l_n$
- clausal form is set of clauses $\{C_1, \ldots, C_m\}$, representing $\forall (C_1 \wedge \cdots \wedge C_m)$

Resolution (predicate)

Definition

- literal is atom $P(t_1, \ldots, t_n)$ or negation of atom $\neg P(t_1, \ldots, t_n)$
- clause is set of literals $\{l_1, \ldots, l_n\}$, representing $l_1 \vee \cdots \vee l_n$
- clausal form is set of clauses $\{C_1, \ldots, C_m\}$, representing $\forall (C_1 \wedge \cdots \wedge C_m)$
- clauses C_1 and C_2 without common variables clash if $\exists$ literals $l_1 \in C_1$ and $l_2 \in C_2$ such that l_1 and l_2^c are unifiable

Resolution (predicate)

Definition

- literal is atom $P(t_1, \ldots, t_n)$ or negation of atom $\neg P(t_1, \ldots, t_n)$
- clause is set of literals $\{l_1, \ldots, l_n\}$, representing $l_1 \vee \cdots \vee l_n$
- clausal form is set of clauses $\{C_1, \ldots, C_m\}$, representing $\forall (C_1 \wedge \cdots \wedge C_m)$
- clauses C_1 and C_2 without common variables clash if $\exists$ literals $l_1 \in C_1$ and $l_2 \in C_2$ such that l_1 and l_2^c are unifiable

Resolution (predicate)

Definition

- literal is atom $P(t_1, \ldots, t_n)$ or negation of atom $\neg P(t_1, \ldots, t_n)$
- clause is set of literals $\{l_1, \ldots, l_n\}$, representing $l_1 \vee \cdots \vee l_n$
- clausal form is set of clauses $\{C_1, \ldots, C_m\}$, representing $\forall (C_1 \wedge \cdots \wedge C_m)$
- clauses C_1 and C_2 without common variables clash if $\exists$ literals $l_1 \in C_1$ and $l_2 \in C_2$ such that l_1 and l_2^c are unifiable
- binary resolvent of clashing clauses C_1 and C_2 is clause

$$((C_1 \setminus \{l_1\}) \cup (C_2 \setminus \{l_2\}))\theta$$

where θ is mgu of l_1 and l_2^c

Example (1)

1. $\{\neg P(x), Q(x), R(x, f(x))\}$
2. $\{\neg P(x), Q(x), S(f(x))\}$
3. $\{T(a)\}$
4. $\{P(a)\}$
5. $\{\neg R(a, y), T(y)\}$
6. $\{\neg T(x), \neg Q(x)\}$
7. $\{\neg T(x), \neg S(x)\}$

Example (1)

1. $\{\neg P(x), Q(x), R(x, f(x))\}$
2. $\{\neg P(x), Q(x), S(f(x))\}$
3. $\{T(a)\}$
4. $\{P(a)\}$
5. $\{\neg R(a, y), T(y)\}$
6. $\{\neg T(x), \neg Q(x)\}$
7. $\{\neg T(x), \neg S(x)\}$
8. $\{\neg Q(a)\}$ resolve 3, 6 $\{a/x\}$

Example (1)

1. $\{\neg P(x), Q(x), R(x, f(x))\}$
2. $\{\neg P(x), Q(x), S(f(x))\}$
3. $\{T(a)\}$
4. $\{P(a)\}$
5. $\{\neg R(a, y), T(y)\}$
6. $\{\neg T(x), \neg Q(x)\}$
7. $\{\neg T(x), \neg S(x)\}$
8. $\{\neg Q(a)\}$ resolve 3, 6 $\{a/x\}$
9. $\{Q(a), S(f(a))\}$ resolve 2, 4 $\{a/x\}$

Example (1)

1.	$\{\neg P(x), Q(x), R(x, f(x))\}$		
2.	$\{\neg P(x), Q(x), S(f(x))\}$		
3.	$\{T(a)\}$		
4.	$\{P(a)\}$		
5.	$\{\neg R(a, y), T(y)\}$		
6.	$\{\neg T(x), \neg Q(x)\}$		
7.	$\{\neg T(x), \neg S(x)\}$		
8.	$\{\neg Q(a)\}$	resolve 3, 6	$\{a/x\}$
9.	$\{Q(a), S(f(a))\}$	resolve 2, 4	$\{a/x\}$
10.	$\{Q(a), R(a, f(a))\}$	resolve 1, 4	$\{a/x\}$

Example (1)

1.	$\{\neg P(x), Q(x), R(x, f(x))\}$		
2.	$\{\neg P(x), Q(x), S(f(x))\}$		
3.	$\{T(a)\}$		
4.	$\{P(a)\}$		
5.	$\{\neg R(a, y), T(y)\}$		
6.	$\{\neg T(x), \neg Q(x)\}$		
7.	$\{\neg T(x), \neg S(x)\}$		
8.	$\{\neg Q(a)\}$	resolve 3, 6	$\{a/x\}$
9.	$\{Q(a), S(f(a))\}$	resolve 2, 4	$\{a/x\}$
10.	$\{Q(a), R(a, f(a))\}$	resolve 1, 4	$\{a/x\}$
11.	$\{S(f(a))\}$	resolve 8, 9	

Example (1)

1.	$\{\neg P(x), Q(x), R(x, f(x))\}$		
2.	$\{\neg P(x), Q(x), S(f(x))\}$		
3.	$\{T(a)\}$		
4.	$\{P(a)\}$		
5.	$\{\neg R(a, y), T(y)\}$		
6.	$\{\neg T(x), \neg Q(x)\}$		
7.	$\{\neg T(x), \neg S(x)\}$		
8.	$\{\neg Q(a)\}$	resolve 3, 6	$\{a/x\}$
9.	$\{Q(a), S(f(a))\}$	resolve 2, 4	$\{a/x\}$
10.	$\{Q(a), R(a, f(a))\}$	resolve 1, 4	$\{a/x\}$
11.	$\{S(f(a))\}$	resolve 8, 9	
12.	$\{R(a, f(a))\}$	resolve 8, 10	

Example (1)

1.	$\{\neg P(x), Q(x), R(x, f(x))\}$		
2.	$\{\neg P(x), Q(x), S(f(x))\}$		
3.	$\{T(a)\}$		
4.	$\{P(a)\}$		
5.	$\{\neg R(a, y), T(y)\}$		
6.	$\{\neg T(x), \neg Q(x)\}$		
7.	$\{\neg T(x), \neg S(x)\}$		
8.	$\{\neg Q(a)\}$	resolve 3, 6	$\{a/x\}$
9.	$\{Q(a), S(f(a))\}$	resolve 2, 4	$\{a/x\}$
10.	$\{Q(a), R(a, f(a))\}$	resolve 1, 4	$\{a/x\}$
11.	$\{S(f(a))\}$	resolve 8, 9	
12.	$\{R(a, f(a))\}$	resolve 8, 10	
13.	$\{T(f(a))\}$	resolve 5, 12	$\{f(a)/y\}$

Example (1)

1.	$\{\neg P(x), Q(x), R(x, f(x))\}$		
2.	$\{\neg P(x), Q(x), S(f(x))\}$		
3.	$\{T(a)\}$		
4.	$\{P(a)\}$		
5.	$\{\neg R(a, y), T(y)\}$		
6.	$\{\neg T(x), \neg Q(x)\}$		
7.	$\{\neg T(x), \neg S(x)\}$		
8.	$\{\neg Q(a)\}$	resolve 3, 6	$\{a/x\}$
9.	$\{Q(a), S(f(a))\}$	resolve 2, 4	$\{a/x\}$
10.	$\{Q(a), R(a, f(a))\}$	resolve 1, 4	$\{a/x\}$
11.	$\{S(f(a))\}$	resolve 8, 9	
12.	$\{R(a, f(a))\}$	resolve 8, 10	
13.	$\{T(f(a))\}$	resolve 5, 12	$\{f(a)/y\}$
14.	$\{\neg S(f(a))\}$	resolve 7, 13	$\{f(a)/x\}$

Example (1)

1.	$\{\neg P(x), Q(x), R(x, f(x))\}$		
2.	$\{\neg P(x), Q(x), S(f(x))\}$		
3.	$\{T(a)\}$		
4.	$\{P(a)\}$		
5.	$\{\neg R(a, y), T(y)\}$		
6.	$\{\neg T(x), \neg Q(x)\}$		
7.	$\{\neg T(x), \neg S(x)\}$		
8.	$\{\neg Q(a)\}$	resolve 3, 6	$\{a/x\}$
9.	$\{Q(a), S(f(a))\}$	resolve 2, 4	$\{a/x\}$
10.	$\{Q(a), R(a, f(a))\}$	resolve 1, 4	$\{a/x\}$
11.	$\{S(f(a))\}$	resolve 8, 9	
12.	$\{R(a, f(a))\}$	resolve 8, 10	
13.	$\{T(f(a))\}$	resolve 5, 12	$\{f(a)/y\}$
14.	$\{\neg S(f(a))\}$	resolve 7, 13	$\{f(a)/x\}$
15.	$\square$	resolve 11,14	

Example (2)

1. $\{\neg P(x,y), P(y,x)\}$
2. $\{\neg P(x,y), \neg P(y,z), P(x,z)\}$
3. $\{P(x, f(x))\}$
4. $\{\neg P(x,x)\}$

$$\forall x\, \forall y\, \forall z \left(\begin{array}{l} (\neg P(x,y) \vee P(y,x)) \wedge (\neg P(x,y) \vee \neg P(y,z) \vee P(x,z)) \\ \wedge\, P(x, f(x)) \wedge \neg P(x,x) \end{array} \right)$$

Example (2)

1. $\{\neg P(x,y), P(y,x)\}$
2. $\{\neg P(x,y), \neg P(y,z), P(x,z)\}$
3. $\{P(x, f(x))\}$
4. $\{\neg P(x,x)\}$
3′. $\{P(x', f(x'))\}$ rename 3

$$\forall x\, \forall y\, \forall z \left(\begin{array}{c} (\neg P(x,y) \vee P(y,x)) \wedge (\neg P(x,y) \vee \neg P(y,z) \vee P(x,z)) \\ \wedge\, P(x, f(x)) \wedge \neg P(x,x) \end{array} \right)$$

Example (2)

1. $\{\neg P(x,y), P(y,x)\}$
2. $\{\neg P(x,y), \neg P(y,z), P(x,z)\}$
3. $\{P(x, f(x))\}$
4. $\{\neg P(x,x)\}$
3′. $\{P(x', f(x'))\}$ rename 3
5. $\{P(f(x), x)\}$ resolve 1, 3′ $\{f(x)/y, x/x'\}$

$$\forall x\, \forall y\, \forall z \left(\begin{array}{l} (\neg P(x,y) \vee P(y,x)) \wedge (\neg P(x,y) \vee \neg P(y,z) \vee P(x,z)) \\ \wedge\, P(x, f(x)) \wedge \neg P(x,x) \end{array} \right)$$

Example (2)

1. $\{\neg P(x,y), P(y,x)\}$
2. $\{\neg P(x,y), \neg P(y,z), P(x,z)\}$
3. $\{P(x, f(x))\}$
4. $\{\neg P(x,x)\}$
3′. $\{P(x', f(x'))\}$ rename 3
5. $\{P(f(x), x)\}$ resolve 1, 3′ $\{f(x)/y, x/x'\}$
6. $\{\neg P(f(x), z), P(x,z)\}$ resolve 2, 3′ $\{f(x)/y, x/x'\}$

$$\forall x\, \forall y\, \forall z \left(\begin{array}{l} (\neg P(x,y) \vee P(y,x)) \wedge (\neg P(x,y) \vee \neg P(y,z) \vee P(x,z)) \\ \wedge\, P(x, f(x)) \wedge \neg P(x,x) \end{array} \right)$$

Example (2)

1. $\{\neg P(x,y), P(y,x)\}$
2. $\{\neg P(x,y), \neg P(y,z), P(x,z)\}$
3. $\{P(x, f(x))\}$
4. $\{\neg P(x,x)\}$
3′. $\{P(x', f(x'))\}$ rename 3
5. $\{P(f(x), x)\}$ resolve 1, 3′ $\{f(x)/y, x/x'\}$
6. $\{\neg P(f(x), z), P(x,z)\}$ resolve 2, 3′ $\{f(x)/y, x/x'\}$
5′. $\{P(f(x'), x')\}$ rename 5

$$\forall x\, \forall y\, \forall z \left(\begin{array}{l} (\neg P(x,y) \vee P(y,x)) \wedge (\neg P(x,y) \vee \neg P(y,z) \vee P(x,z)) \\ \wedge\, P(x, f(x)) \wedge \neg P(x,x) \end{array} \right)$$

Example (2)

1.	$\{\neg P(x,y), P(y,x)\}$		
2.	$\{\neg P(x,y), \neg P(y,z), P(x,z)\}$		
3.	$\{P(x, f(x))\}$		
4.	$\{\neg P(x,x)\}$		
3′.	$\{P(x', f(x'))\}$	rename 3	
5.	$\{P(f(x), x)\}$	resolve 1, 3′	$\{f(x)/y, x/x'\}$
6.	$\{\neg P(f(x), z), P(x,z)\}$	resolve 2, 3′	$\{f(x)/y, x/x'\}$
5′.	$\{P(f(x'), x')\}$	rename 5	
7.	$\{P(z,z)\}$	resolve 6, 5′	$\{z/x, z/x'\}$

$$\forall x\, \forall y\, \forall z \left(\begin{array}{c} (\neg P(x,y) \vee P(y,x)) \wedge (\neg P(x,y) \vee \neg P(y,z) \vee P(x,z)) \\ \wedge\, P(x, f(x)) \wedge \neg P(x,x) \end{array} \right)$$

Example (2)

1. $\{\neg P(x,y), P(y,x)\}$
2. $\{\neg P(x,y), \neg P(y,z), P(x,z)\}$
3. $\{P(x, f(x))\}$
4. $\{\neg P(x,x)\}$
3′. $\{P(x', f(x'))\}$ rename 3
5. $\{P(f(x), x)\}$ resolve 1, 3′ $\{f(x)/y, x/x'\}$
6. $\{\neg P(f(x), z), P(x,z)\}$ resolve 2, 3′ $\{f(x)/y, x/x'\}$
5′. $\{P(f(x'), x')\}$ rename 5
7. $\{P(z,z)\}$ resolve 6, 5′ $\{z/x, z/x'\}$
8. $\square$ resolve 4, 7 $\{z/x\}$

$$\forall x\, \forall y\, \forall z \left(\begin{array}{c} (\neg P(x,y) \vee P(y,x)) \wedge (\neg P(x,y) \vee \neg P(y,z) \vee P(x,z)) \\ \wedge\, P(x, f(x)) \wedge \neg P(x,x) \end{array} \right)$$

Theorem

binary resolution is sound:

clausal form S is unsatisfiable if S admits refutation

Theorem

binary resolution is sound:

clausal form S is unsatisfiable if S admits refutation

Problem

binary resolution is incomplete

Theorem

binary resolution is sound:

clausal form S is unsatisfiable if S admits refutation

Problem

binary resolution is incomplete

Example

1. $\{P(x), P(y)\}$
2. $\{\neg P(x'), \neg P(y')\}$

unsatisfiable

Theorem

binary resolution is sound:

clausal form S is unsatisfiable if S admits refutation

Problem

binary resolution is incomplete

Example

1. $\{P(x), P(y)\}$
2. $\{\neg P(x'), \neg P(y')\}$
3. $\{P(y), \neg P(y')\}$ resolve 1, 2 $\{x'/x\}$

unsatisfiable

Theorem

binary resolution is sound:

clausal form S is unsatisfiable if S admits refutation

Problem

binary resolution is incomplete

Example

1. $\{P(x), P(y)\}$
2. $\{\neg P(x'), \neg P(y')\}$
3. $\{P(y), \neg P(y')\}$ resolve 1, 2 $\{x'/x\}$

unsatisfiable but no refutation

Solution

incorporate factoring: $C\sigma$ is a factor of C if two or more literals in C have mgu σ

Solution

incorporate factoring: $C\sigma$ is a factor of C if two or more literals in C have mgu σ

Example

1. $\{P(x), P(y)\}$
2. $\{\neg P(x'), \neg P(y')\}$
3. $\{P(y), \neg P(y')\}$ resolve 1, 2 $\{x'/x\}$

unsatisfiable but no refutation

Solution

incorporate factoring: $C\sigma$ is a factor of C if two or more literals in C have mgu σ

Example

1. $\{P(x), P(y)\}$ factor $\{P(x)\}$
2. $\{\neg P(x'), \neg P(y')\}$ factor $\{\neg P(x')\}$
3. $\square$

refutation

ATP summary

This already allows building a reasonable saturation-based ATP for first-order logic

Next time we will see how to make it more efficient and how to add learning to it

Summary

This Lecture

- learning in different prover foundations
- transformation to CNF
- resolution

Next

- Equality handling in automated proof
- Machine learning in E-Prover
- Machine learning for tableaux