Automated Theorem Proving

Georg Moser

Institute of Computer Science @ UIBK

Winter 2015

Summary of Last Lecture

Definition

$$\frac{\gamma}{\gamma(t)} \quad t \text{ term in } \mathcal{L}^+ \qquad \frac{\delta}{\delta(k)} \quad k \text{ fresh constant in } \mathcal{L}^+$$

1. $\mathcal{L}^+$ denotes extension of base language $\mathcal{L}$
2. new individual constants are introduced in δ rules
3. fresh means new to the branch of the tableau

Theorem

a sentence F is valid iff F has a tableau proof

Outline of the Lecture

Early Approaches in Automated Reasoning

Herbrand's theorem for dummies, Gilmore's prover, method of Davis and Putnam

Starting Points

resolution, tableau provers, Skolemisation, ordered resolution, redundancy and deletion

Automated Reasoning with Equality

paramodulation, ordered completion and proof orders, superposition

Applications of Automated Reasoning

Neuman-Stubblebinde Key Exchange Protocol, Robbins problem

Outline of the Lecture

Early Approaches in Automated Reasoning

Herbrand's theorem for dummies, Gilmore's prover, method of Davis and Putnam

Starting Points

resolution, tableau provers, Skolemisation, ordered resolution, redundancy and deletion

Automated Reasoning with Equality

paramodulation, ordered completion and proof orders, superposition

Applications of Automated Reasoning

Neuman-Stubblebinde Key Exchange Protocol, Robbins problem

First-Order Tableau

Example

consider the tableau proof of

$$\exists x \forall y R(x, y) \to \forall y \exists x R(x, y)$$

on the whiteboard

Free-Variable Semantic Tableaux

Definition (expansion rules)

$$\frac{\gamma}{\gamma(x)} \quad x \text{ a free variable} \qquad \frac{\delta}{\delta(f(x_1, \ldots, x_n))} \quad f \text{ a Skolem function}$$

- $x_1, \ldots, x_n$ denote all free variables of the formula δ
- Skolem function f must be new on the branch

Free-Variable Semantic Tableaux

Definition (expansion rules)

$$\frac{\gamma}{\gamma(x)} \quad x \text{ a free variable} \qquad \frac{\delta}{\delta(f(x_1, \ldots, x_n))} \quad f \text{ a Skolem function}$$

- $x_1, \ldots, x_n$ denote all free variables of the formula δ
- Skolem function f must be new on the branch

Remark

- δ-rule still leaves a lot of room for improvement
- requirement that f must be new on the branch forces the introduction of inefficiently many new Skolem functions
- prevented with cleverer notions of the δ-rule

Definition (atomic closure rule)

1. $\exists$ branch in tableau T that contains two literals A and $\neg B$
2. $\exists$ mgu σ of A and B
3. then $T\sigma$ is also a tableau

Definition (atomic closure rule)

1. $\exists$ branch in tableau T that contains two literals A and $\neg B$
2. $\exists$ mgu σ of A and B
3. then $T\sigma$ is also a tableau

Definition

consider the following tableau substitution rule:

1. T is a tableau for $\mathcal{G}$
2. σ is free for any sentence in $\mathcal{G}$
3. then $T\sigma$ is also a tableau

Definition (atomic closure rule)

1. $\exists$ branch in tableau T that contains two literals A and $\neg B$
2. $\exists$ mgu σ of A and B
3. then $T\sigma$ is also a tableau

Definition

consider the following tableau substitution rule:

1. T is a tableau for $\mathcal{G}$
2. σ is free for any sentence in $\mathcal{G}$
3. then $T\sigma$ is also a tableau

Remark

completeness of free-variable tableaux can (eventually) be proven via model existence

Example

consider the tableau proof of

$$\exists x \forall y R(x, y) \rightarrow \forall y \exists x R(x, y)$$

and

$$\forall x \forall y (P(x) \land P(y)) \rightarrow \forall x \forall y (P(x) \lor P(y))$$

on the whiteboard

Soundness of Free-Variable Tableaux

Definition

- a branch in a free-variable tableau is called satisfiable, if $\exists$ structure $\mathcal{A}$ and $\forall$ environment ℓ: $(\mathcal{A}, \ell) \models \mathcal{G}$
- a free-variable tableau is satisfiable, if there exists a satisfiable branch

Soundness of Free-Variable Tableaux

Definition

- a branch in a free-variable tableau is called satisfiable, if $\exists$ structure $\mathcal{A}$ and $\forall$ environment ℓ: $(\mathcal{A}, \ell) \models \mathcal{G}$
- a free-variable tableau is satisfiable, if there exists a satisfiable branch

Lemma

1. *T be a satisfiable (free-variable) tableau*
2. *propositional or (free-variable) first-order expansion rule is applied*
3. *then the result is satisfiable*

Soundness of Free-Variable Tableaux

Definition

- a branch in a free-variable tableau is called satisfiable, if $\exists$ structure $\mathcal{A}$ and $\forall$ environment ℓ: $(\mathcal{A}, \ell) \models \mathcal{G}$
- a free-variable tableau is satisfiable, if there exists a satisfiable branch

Lemma

1. *T be a satisfiable (free-variable) tableau*
2. *propositional or (free-variable) first-order expansion rule is applied*
3. *then the result is satisfiable*

Proof

the lemma follows by case-distinction on the applied expansion rule, it suffices to consider the δ-rule all other cases are similar

Proof (cont'd).

1. suppose B is a satisfiable branch in T such that δ occurs on B

Proof (cont'd).

1. suppose B is a satisfiable branch in T such that δ occurs on B
2. extend B with $\delta(f(x_1, \ldots, x_n))$ and call the result B'; T' denotes the corresponding tableau

Proof (cont'd).

1. suppose B is a satisfiable branch in T such that δ occurs on B
2. extend B with $\delta(f(x_1, \ldots, x_n))$ and call the result B'; T' denotes the corresponding tableau
3. $\mathcal{G}$ collects all formulas on B and assume $(\mathcal{A}, \ell) \models \mathcal{G}$

Proof (cont'd).

1. suppose B is a satisfiable branch in T such that δ occurs on B
2. extend B with $\delta(f(x_1, \dots, x_n))$ and call the result B'; T' denotes the corresponding tableau
3. $\mathcal{G}$ collects all formulas on B and assume $(\mathcal{A}, \ell) \models \mathcal{G}$
4. let x be the existentially bound variable replaced by $f(x_1, \dots, x_n)$

Proof (cont'd).

1. suppose B is a satisfiable branch in T such that δ occurs on B
2. extend B with $\delta(f(x_1, \dots, x_n))$ and call the result B'; T' denotes the corresponding tableau
3. $\mathcal{G}$ collects all formulas on B and assume $(\mathcal{A}, \ell) \models \mathcal{G}$
4. let x be the existentially bound variable replaced by $f(x_1, \dots, x_n)$
5. $\exists$ witness $a \in \mathcal{A}$ for x such that $(\mathcal{A}, \ell\{x \mapsto a\}) \models \delta(x)$

Proof (cont'd).

1. suppose B is a satisfiable branch in T such that δ occurs on B
2. extend B with $\delta(f(x_1, \dots, x_n))$ and call the result B'; T' denotes the corresponding tableau
3. $\mathcal{G}$ collects all formulas on B and assume $(\mathcal{A}, \ell) \models \mathcal{G}$
4. let x be the existentially bound variable replaced by $f(x_1, \dots, x_n)$
5. $\exists$ witness $a \in \mathcal{A}$ for x such that $(\mathcal{A}, \ell\{x \mapsto a\}) \models \delta(x)$
6. construct $\mathcal{A}'$ such that

$$f^{\mathcal{A}'}(\ell(x_1), \dots, \ell(x_n)) := a$$

Proof (cont'd).

1. suppose B is a satisfiable branch in T such that δ occurs on B
2. extend B with $\delta(f(x_1, \dots, x_n))$ and call the result B'; T' denotes the corresponding tableau
3. $\mathcal{G}$ collects all formulas on B and assume $(\mathcal{A}, \ell) \models \mathcal{G}$
4. let x be the existentially bound variable replaced by $f(x_1, \dots, x_n)$
5. $\exists$ witness $a \in \mathcal{A}$ for x such that $(\mathcal{A}, \ell\{x \mapsto a\}) \models \delta(x)$
6. construct $\mathcal{A}'$ such that

$$f^{\mathcal{A}'}(\ell(x_1), \dots, \ell(x_n)) := a$$

7. extendable to a total definition of $f^{\mathcal{A}'}$

Proof (cont'd).

1. suppose B is a satisfiable branch in T such that δ occurs on B
2. extend B with $\delta(f(x_1,\dots,x_n))$ and call the result B'; T' denotes the corresponding tableau
3. $\mathcal{G}$ collects all formulas on B and assume $(\mathcal{A},\ell) \models \mathcal{G}$
4. let x be the existentially bound variable replaced by $f(x_1,\dots,x_n)$
5. $\exists$ witness $a \in \mathcal{A}$ for x such that $(\mathcal{A},\ell\{x \mapsto a\}) \models \delta(x)$
6. construct $\mathcal{A}'$ such that
$$f^{\mathcal{A}'}(\ell(x_1),\dots,\ell(x_n)) := a$$
7. extendable to a total definition of $f^{\mathcal{A}'}$
8. we conclude
$$(\mathcal{A},\ell) \models \delta \quad\Longrightarrow\quad (\mathcal{A}',\ell) \models \delta(f(x_1,\dots,x_n))$$

Proof (cont'd).

1. suppose B is a satisfiable branch in T such that δ occurs on B
2. extend B with $\delta(f(x_1,\dots,x_n))$ and call the result B'; T' denotes the corresponding tableau
3. $\mathcal{G}$ collects all formulas on B and assume $(\mathcal{A},\ell) \models \mathcal{G}$
4. let x be the existentially bound variable replaced by $f(x_1,\dots,x_n)$
5. $\exists$ witness $a \in \mathcal{A}$ for x such that $(\mathcal{A},\ell\{x \mapsto a\}) \models \delta(x)$
6. construct $\mathcal{A}'$ such that

$$f^{\mathcal{A}'}(\ell(x_1),\dots,\ell(x_n)) := a$$

7. extendable to a total definition of $f^{\mathcal{A}'}$
8. we conclude

$$(\mathcal{A},\ell) \models \delta \quad \Longrightarrow \quad (\mathcal{A}',\ell) \models \delta(f(x_1,\dots,x_n))$$

∎

Lemma

if the atomic closure rule is applicable to a tableau T and T is satisfiable, then the result is also satisfiable

Lemma

if the atomic closure rule is applicable to a tableau T and T is satisfiable, then the result is also satisfiable

Proof.

1. we show a more general statement:
   if the substitution rule is applied to a satisfiable tableau T, then its result is satisfiable

Lemma

if the atomic closure rule is applicable to a tableau T and T is satisfiable, then the result is also satisfiable

Proof.

1. we show a more general statement:
   if the substitution rule is applied to a satisfiable tableau T, then its result is satisfiable
2. $\forall$ environments ℓ, $\exists$ environment ℓ' such that $t^{(\mathcal{A},\ell')} = t\sigma^{(\mathcal{A},\ell)}$

Lemma

if the atomic closure rule is applicable to a tableau T and T is satisfiable, then the result is also satisfiable

Proof.

1. we show a more general statement:
   if the substitution rule is applied to a satisfiable tableau T, then its result is satisfiable
2. $\forall$ environments ℓ, $\exists$ environment ℓ' such that $t^{(\mathcal{A},\ell')} = t\sigma^{(\mathcal{A},\ell)}$
3. we have to show that $T\sigma$ is satisfiable

Theorem

if the sentence F has a free-variable tableau proof, then F is valid

Lemma

if the atomic closure rule is applicable to a tableau T and T is satisfiable, then the result is also satisfiable

Proof.

1. we show a more general statement:
   if the substitution rule is applied to a satisfiable tableau T, then its result is satisfiable
2. $\forall$ environments ℓ, $\exists$ environment ℓ' such that $t^{(\mathcal{A},\ell')} = t\sigma^{(\mathcal{A},\ell)}$
3. we have to show that $T\sigma$ is satisfiable
4. this follows from the observation and definition of satisfiability

Theorem

if the sentence F has a free-variable tableau proof, then F is valid

Lemma

if the atomic closure rule is applicable to a tableau T and T is satisfiable, then the result is also satisfiable

Proof.

1. we show a more general statement:
   if the substitution rule is applied to a satisfiable tableau T, then its result is satisfiable
2. $\forall$ environments ℓ, $\exists$ environment ℓ' such that $t^{(\mathcal{A},\ell')} = t\sigma^{(\mathcal{A},\ell)}$
3. we have to show that $T\sigma$ is satisfiable
4. this follows from the observation and definition of satisfiability ∎

Theorem

if the sentence F has a free-variable tableau proof, then F is valid

Strong Completeness of Free-Variable Tableaux

NB: may consider a sequence of atomic closure rules that leads to an (atomically closed) tableau as one block

Strong Completeness of Free-Variable Tableaux

NB: may consider a sequence of atomic closure rules that leads to an (atomically closed) tableau as one block

Definition

- T be a tableau with branches $B_1, \dots, B_n$
- $\forall\ i\ A_i$ and $\neg B_i$ are literals on B_i
- if σ is a mgu of $A_1 = B_1, \dots, A_n = B_n$
- then σ is called most general atomic closure substitution

Strong Completeness of Free-Variable Tableaux

NB: may consider a sequence of atomic closure rules that leads to an (atomically closed) tableau as one block

Definition

- T be a tableau with branches $B_1, \dots, B_n$
- $\forall\ i\ A_i$ and $\neg B_i$ are literals on B_i
- if σ is a mgu of $A_1 = B_1, \dots, A_n = B_n$
- then σ is called most general atomic closure substitution

Lemma (Lifting Lemma)

1. *τ a substitution free for tableau T such that each branch in $T\tau$ is atomically closed*
2. *then $\exists$ a most general atomic closure substitution σ and*
3. *$T\sigma$ is closed by n applications of the atomic closure rule*

Definition

a strategy S details:

1. which expansion rule is supposed to be applied
2. or that no expansion rule can be applied

a strategy may use extra information which is updated

Definition

a strategy S details:

1. which expansion rule is supposed to be applied
2. or that no expansion rule can be applied

a strategy may use extra information which is updated

Definition

a strategy S is fair if for sequence of tableaux $T_1, T_2, \ldots$ following S :

Definition

a strategy S details:

1. which expansion rule is supposed to be applied
2. or that no expansion rule can be applied

a strategy may use extra information which is updated

Definition

a strategy S is fair if for sequence of tableaux $T_1, T_2, \ldots$ following S :

1. any non-literal formula in T_i is eventually expanded, and

Definition

a strategy S details:

1. which expansion rule is supposed to be applied
2. or that no expansion rule can be applied

a strategy may use extra information which is updated

Definition

a strategy S is fair if for sequence of tableaux $T_1, T_2, \ldots$ following S :

1. any non-literal formula in T_i is eventually expanded, and
2. any γ-formula occurrence in T_i has the γ-rule applied to it arbitrarily often

Definition

a strategy S details:

1. which expansion rule is supposed to be applied
2. or that no expansion rule can be applied

a strategy may use extra information which is updated

Definition

a strategy S is fair if for sequence of tableaux $T_1, T_2, \ldots$ following S :

1. any non-literal formula in T_i is eventually expanded, and
2. any γ-formula occurrence in T_i has the γ-rule applied to it arbitrarily often

Example

strategy employed in the implementation of free-variable tableaux is fair

Example

- for each tableau the extra information includes
  1. which formula occurrences have been used on which branch
  2. priority order for formula occurrences on each branch
  3. priority order for branches

Example

- for each tableau the extra information includes
  1. which formula occurrences have been used on which branch
  2. priority order for formula occurrences on each branch
  3. priority order for branches
- extra information for initial tableau
  1. $\neg F$ is not used
  2. $\neg F$ has top priority
  3. single branch has top priority

Example

- for each tableau the extra information includes
  1. which formula occurrences have been used on which branch
  2. priority order for formula occurrences on each branch
  3. priority order for branches
- extra information for initial tableau
  1. $\neg F$ is not used
  2. $\neg F$ has top priority
  3. single branch has top priority
- select branch of highest priority with unused formula
- select formula occurrence on this branch of highest priority
- apply expansion rule; give formula occurrence and branch lowest priority

Example

- for each tableau the extra information includes
  1. which formula occurrences have been used on which branch
  2. priority order for formula occurrences on each branch
  3. priority order for branches
- extra information for initial tableau
  1. $\neg F$ is not used
  2. $\neg F$ has top priority
  3. single branch has top priority
- select branch of highest priority with unused formula
- select formula occurrence on this branch of highest priority
- apply expansion rule; give formula occurrence and branch lowest priority
- if every non-literal formula has been used on any branch no continuation is possible

Example

- for each tableau the extra information includes
  1. which formula occurrences have been used on which branch
  2. priority order for formula occurrences on each branch
  3. priority order for branches
- extra information for initial tableau
  1. $\neg F$ is not used
  2. $\neg F$ has top priority
  3. single branch has top priority
- select branch of highest priority with unused formula
- select formula occurrence on this branch of highest priority
- apply expansion rule; give formula occurrence and branch lowest priority
- if every non-literal formula has been used on any branch no continuation is possible

this strategy is not fair

Theorem (Strong Completeness)

1. *S be a fair strategy*
2. *F be a valid sentence*
3. *F has a tableau proof with the following properties:*
   - *all tableau expansion rules are considered first and follow strategy S*
   - *a block of atomic closure rules closes the tableau*

Theorem (Strong Completeness)

1. *S be a fair strategy*
2. *F be a valid sentence*
3. *F has a tableau proof with the following properties:*
   - *all tableau expansion rules are considered first and follow strategy S*
   - *a block of atomic closure rules closes the tableau*

Proof Sketch.

1. we argue indirectly and suppose that a given formula F does not admit a tableau proof

Theorem (Strong Completeness)

1. *S be a fair strategy*
2. *F be a valid sentence*
3. *F has a tableau proof with the following properties:*
   - *all tableau expansion rules are considered first and follow strategy S*
   - *a block of atomic closure rules closes the tableau*

Proof Sketch.

1. we argue indirectly and suppose that a given formula F does not admit a tableau proof
2. $\exists$ open branch starting with $\neg F$

Theorem (Strong Completeness)

1. *S be a fair strategy*
2. *F be a valid sentence*
3. *F has a tableau proof with the following properties:*
   - *all tableau expansion rules are considered first and follow strategy S*
   - *a block of atomic closure rules closes the tableau*

Proof Sketch.

1. we argue indirectly and suppose that a given formula F does not admit a tableau proof
2. $\exists$ open branch starting with $\neg F$
3. based on syntactic properties (to be presented) we can conclude that all formula on the branch are satisfiable[a]

Theorem (Strong Completeness)

1. *S be a fair strategy*
2. *F be a valid sentence*
3. *F has a tableau proof with the following properties:*
   - *all tableau expansion rules are considered first and follow strategy S*
   - *a block of atomic closure rules closes the tableau*

Proof Sketch.

1. we argue indirectly and suppose that a given formula F does not admit a tableau proof
2. $\exists$ open branch starting with $\neg F$
3. based on syntactic properties (to be presented) we can conclude that all formula on the branch are satisfiable[a]
4. hence $\neg F$ is satisfiable, and we have found a counter model

Theorem (Strong Completeness)

1. *S be a fair strategy*
2. *F be a valid sentence*
3. *F has a tableau proof with the following properties:*
   - *all tableau expansion rules are considered first and follow strategy S*
   - *a block of atomic closure rules closes the tableau*

Proof Sketch.

1. we argue indirectly and suppose that a given formula F does not admit a tableau proof
2. $\exists$ open branch starting with $\neg F$
3. based on syntactic properties (to be presented) we can conclude that all formula on the branch are satisfiable[a]
4. hence $\neg F$ is satisfiable, and we have found a counter model

∎

[a] the formulas on the branch form a Hintikka set

Implementation of γ-Rule

γ-rule (simplified)

```
singlestep([OldBranch | Rest], NewTree) :-
   member(NotatedGamma, OldBranch),
   notation(NotatedGamma, Free),
   fmla(NotatedGamma, Gamma),
   is_universal(Gamma),
   remove(NotatedGamma, OldBranch, TempBranch),
   NewFree = [V | Free],
   instance(Gamma, V, GammaInstance),
   notation(NotatedGammaInstance, NewFree),
   fmla(NotatedGammaInstance, GammaInstance),
   append([NotatedGammaInstance | TempBranch],
          [NotatedGamma], NewBranch),
   append(Rest, [NewBranch], NewTree).
```