Computational Logic

Automated Theorem Proving

Georg Moser

Institute of Computer Science @ UIBK

Winter 2015

Summary of Last Lecture

Method of Davis and Putnam in Pseudo-Code

```
if C does not contain function symbols
then apply DPLL(a)-DPLL(c) on C'_0
else {
  n := 0;
  contr := false;
  while (¬ contr) do {
    apply DPLL(a)-DPLL(c) on C'_n;
    if the decision tree proves unsatisfiability,
    then contr := true
    else contr := false;
    n := n + 1;
  }}
```

Definition

- individual constants
  $k_0, k_1, \ldots, k_j, \ldots$ denoted c, d, etc.
- function constants with i arguments
  $f_0^i, f_1^i, \ldots, f_j^i, \ldots$ denoted f, g, h, etc.
- predicate constants with i arguments
  $R_0^i, R_1^i, \ldots, R_j^i, \ldots$ denoted P, Q, R, etc.
- variables, collected in $\mathcal{V}$
  $x_0, x_1, \ldots, x_j, \ldots$ denoted x, y, z, etc.

Definition

- propositional connectives $\neg$, $\vee$
- equality sign $=$

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

Resolution Calculus for First-Order Logic

Definition

resolution

$$\frac{C \vee A \quad D \vee \neg B}{(C \vee D)\sigma}$$

factoring

$$\frac{C \vee A \vee B}{(C \vee A)\sigma}$$

σ is a mgu of the atomic formulas A and B

Resolution Calculus for First-Order Logic

restricted to atoms

Definition

resolution

$$\frac{C \vee A \quad D \vee \neg B}{(C \vee D)\sigma}$$

factoring

$$\frac{C \vee A \vee B}{(C \vee A)\sigma}$$

σ is a mgu of the atomic formulas A and B

Resolution Calculus for First-Order Logic

Definition

resolution

$$\frac{C \vee A \quad D \vee \neg B}{(C \vee D)\sigma}$$

factoring

$$\frac{C \vee A \vee B}{(C \vee A)\sigma}$$

σ is a mgu of the atomic formulas A and B

let $\mathcal{C}$ be a set of clauses; define resolution operator $\mathrm{Res}(\mathcal{C})$

Resolution Calculus for First-Order Logic

Definition

resolution

$$\frac{C \vee A \quad D \vee \neg B}{(C \vee D)\sigma}$$

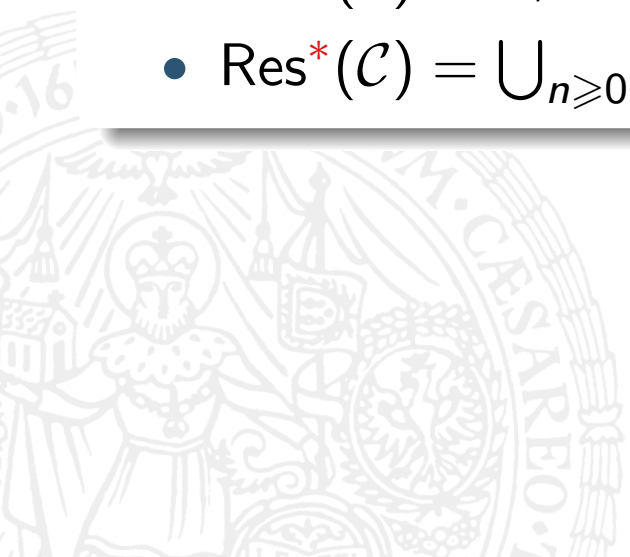
factoring

$$\frac{C \vee A \vee B}{(C \vee A)\sigma}$$

σ is a mgu of the atomic formulas A and B

let $\mathcal{C}$ be a set of clauses; define resolution operator $\mathrm{Res}(\mathcal{C})$

- $\mathrm{Res}(\mathcal{C}) = \{D \mid D \text{ is resolvent or factor with premises in } \mathcal{C}\}$
- $\mathrm{Res}^0(\mathcal{C}) = \mathcal{C}$; $\mathrm{Res}^{n+1}(\mathcal{C}) = \mathrm{Res}^n(\mathcal{C}) \cup \mathrm{Res}(\mathrm{Res}^n(\mathcal{C}))$
- $\mathrm{Res}^*(\mathcal{C}) = \bigcup_{n \geqslant 0} \mathrm{Res}^n(\mathcal{C})$



Resolution Calculus for First-Order Logic

Definition

resolution

$$\frac{C \vee A \quad D \vee \neg B}{(C \vee D)\sigma}$$

factoring

$$\frac{C \vee A \vee B}{(C \vee A)\sigma}$$

σ is a mgu of the atomic formulas A and B

let $\mathcal{C}$ be a set of clauses; define resolution operator $\mathrm{Res}(\mathcal{C})$

- $\mathrm{Res}(\mathcal{C}) = \{D \mid D \text{ is resolvent or factor with premises in } \mathcal{C}\}$
- $\mathrm{Res}^0(\mathcal{C}) = \mathcal{C}$; $\mathrm{Res}^{n+1}(\mathcal{C}) = \mathrm{Res}^n(\mathcal{C}) \cup \mathrm{Res}(\mathrm{Res}^n(\mathcal{C}))$
- $\mathrm{Res}^*(\mathcal{C}) = \bigcup_{n \geqslant 0} \mathrm{Res}^n(\mathcal{C})$

Example

$$\frac{\mathrm{P}(x) \vee \mathrm{Q}(\mathrm{f}(x, \mathrm{g}(y), x)) \quad \mathrm{R}(\mathrm{a}, \mathrm{b}) \vee \neg \mathrm{Q}(\mathrm{f}(z, \mathrm{g}(x'), \mathrm{h}(x')))}{\mathrm{P}(\mathrm{h}(x')) \vee \mathrm{R}(\mathrm{a}, \mathrm{b})} \{x \mapsto \mathrm{h}(x')\}$$

Soundness and Completeness of Resolution

Theorem

resolution is sound: if F a sentence and $\mathcal{C}$ its clause form such that $\square \in \mathrm{Res}^*(\mathcal{C})$, then F is unsatisfiable

Soundness and Completeness of Resolution

Theorem

resolution is sound: if F a sentence and $\mathcal{C}$ its clause form such that $\square \in \mathrm{Res}^*(\mathcal{C})$, then F is unsatisfiable

Proof.

on the whiteboard ■

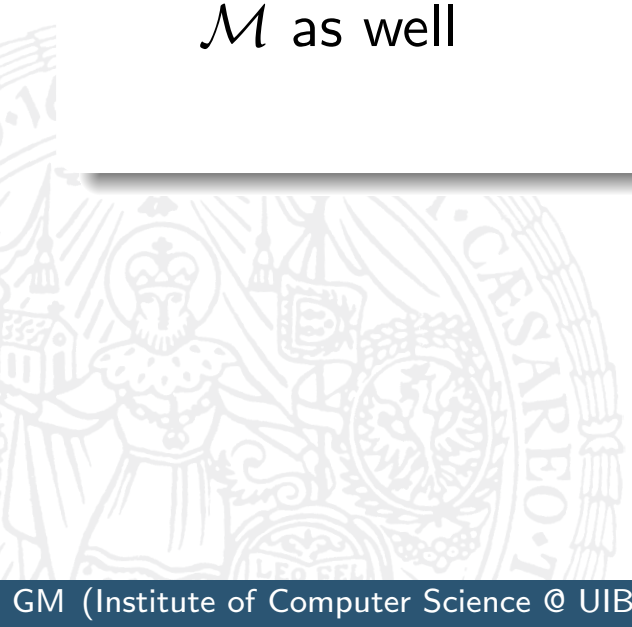
Soundness and Completeness of Resolution

Theorem

resolution is sound: if F a sentence and $\mathcal{C}$ its clause form such that $\square \in \mathrm{Res}^*(\mathcal{C})$, then F is unsatisfiable

Proof.

- the theorem follows by case-distinction on the inferences
- for each inference one verifies that if the assumptions (as formulas) are modelled by an interpretation $\mathcal{M}$, then the consequence holds in $\mathcal{M}$ as well



Soundness and Completeness of Resolution

Theorem

resolution is sound: if F a sentence and $\mathcal{C}$ its clause form such that $\square \in \mathrm{Res}^*(\mathcal{C})$, then F is unsatisfiable

Proof.

- the theorem follows by case-distinction on the inferences
- for each inference one verifies that if the assumptions (as formulas) are modelled by an interpretation $\mathcal{M}$, then the consequence holds in $\mathcal{M}$ as well

Theorem

resolution is (refutationally) complete; if F a sentence and $\mathcal{C}$ its clause form, then $\square \in \mathrm{Res}^*(\mathcal{C})$ if F is unsatisfiable

Tableau Expansion Rules

Definition (uniform notation)

conjunctive			disjunctive		
α	α_1	α_2	β	β_1	β_2
$A \wedge B$	A	B	$\neg(A \wedge B)$	$\neg A$	$\neg B$
$\neg(A \vee B)$	$\neg A$	$\neg B$	$A \vee B$	A	B
$\neg(A \to B)$	A	$\neg B$	$A \to B$	$\neg A$	B

Tableau Expansion Rules

Definition (uniform notation)

conjunctive			disjunctive		
α	α_1	α_2	β	β_1	β_2
$A \wedge B$	A	B	$\neg(A \wedge B)$	$\neg A$	$\neg B$
$\neg(A \vee B)$	$\neg A$	$\neg B$	$A \vee B$	A	B
$\neg(A \to B)$	A	$\neg B$	$A \to B$	$\neg A$	B

Definition (tableau expansion rules)

$$\frac{\neg\neg A}{A} \qquad \frac{\alpha}{\alpha_1 \\ \alpha_2} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

Reminder: Propositional Semantic Tableaux

Computational Logic: week 3

Definition

let $\{A_1, \ldots, A_n\}$ be propositional formulas

- the following tree T is a tableau for $\{A_1, \ldots, A_n\}$:

$$
\begin{array}{c}
A_1 \\
A_2 \\
\vdots \\
A_n
\end{array}
$$

Reminder: Propositional Semantic Tableaux

Computational Logic: week 3

Definition

let $\{A_1, \ldots, A_n\}$ be propositional formulas

- the following tree T is a tableau for $\{A_1, \ldots, A_n\}$:

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}$$

- suppose T is a tableau for $\{A_1, \ldots, A_n\}$ and T^* is obtained by applying a tableau expansion rule to T, then T^* is a tableau for $\{A_1, \ldots, A_n\}$

Example

consider the tableau proof of $(P \to (Q \to R)) \to (P \vee S \to (Q \to R) \vee S)$

$$\neg((P \to (Q \to R)) \to (P \vee S \to (Q \to R) \vee S))$$

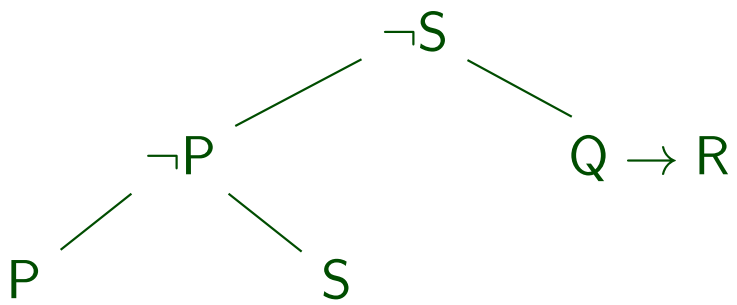
$$P \to (Q \to R)$$

$$\neg(P \vee S \to (Q \to R) \vee S)$$

$$P \vee S$$

$$\neg((Q \to R) \vee S)$$

$$\neg((Q \to R)$$

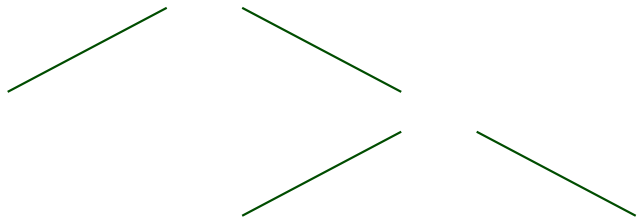


Heuristics Matters

Example

consider $P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q$ and the following tableau proof

$$\neg (P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q)$$



Heuristics Matters

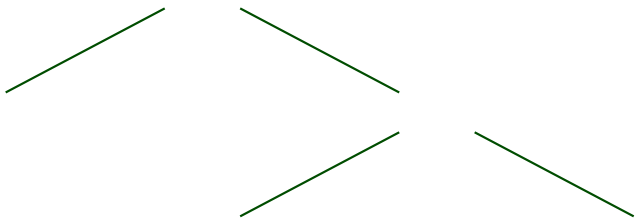
Example

consider $P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q$ and the following tableau proof

$$\neg (P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q)$$

$$P \wedge (Q \rightarrow R \vee S)$$

$$\neg (P \vee Q)$$



Heuristics Matters

Example

consider $P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q$ and the following tableau proof

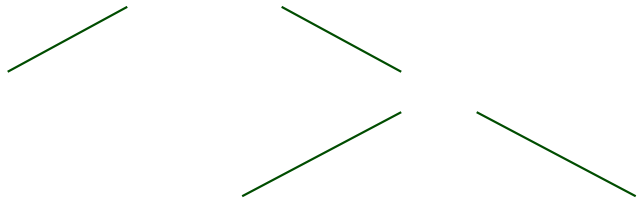
$$\neg (P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q)$$

$$P \wedge (Q \rightarrow R \vee S)$$

$$\neg (P \vee Q)$$

$$P$$

$$Q \rightarrow R \vee S$$



Heuristics Matters

Example

consider $P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q$ and the following tableau proof

$$\neg (P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q)$$

$$P \wedge (Q \rightarrow R \vee S)$$

$$\neg(P \vee Q)$$

$$P$$

$$Q \rightarrow R \vee S$$

$$\neg Q \qquad\qquad R \vee S$$

Heuristics Matters

Example

consider $P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q$ and the following tableau proof

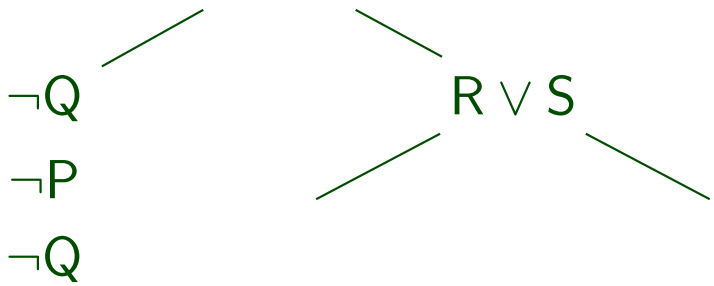
$$\neg (P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q)$$

$$P \wedge (Q \rightarrow R \vee S)$$

$$\neg(P \vee Q)$$

$$P$$

$$Q \rightarrow R \vee S$$



Heuristics Matters

Example

consider $P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q$ and the following tableau proof

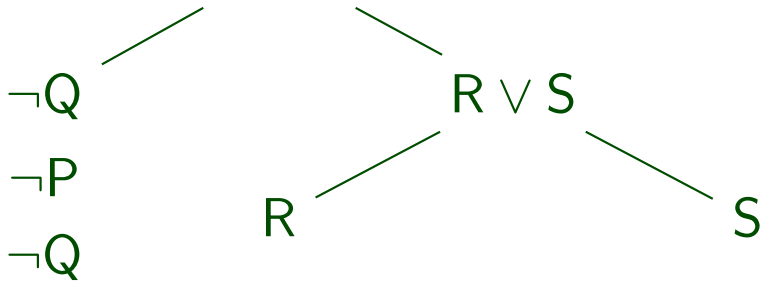
$$\neg (P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q)$$

$$P \wedge (Q \rightarrow R \vee S)$$

$$\neg(P \vee Q)$$

$$P$$

$$Q \rightarrow R \vee S$$



Heuristics Matters

Example

consider $P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q$ and the following tableau proof

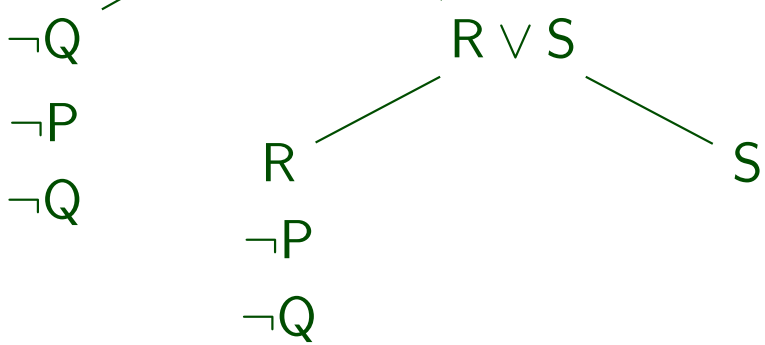
$$\neg (P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q)$$

$$P \wedge (Q \rightarrow R \vee S)$$

$$\neg(P \vee Q)$$

$$P$$

$$Q \rightarrow R \vee S$$



Heuristics Matters

Example

consider $P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q$ and the following tableau proof

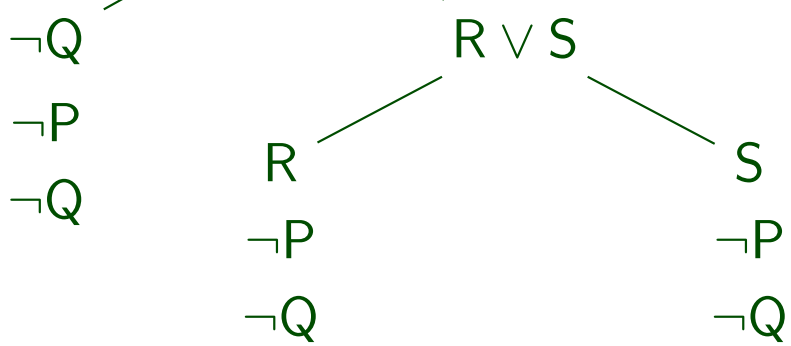
$$\neg (P \wedge (Q \rightarrow R \vee S) \rightarrow P \vee Q)$$

$$P \wedge (Q \rightarrow R \vee S)$$

$$\neg(P \vee Q)$$

$$P$$

$$Q \rightarrow R \vee S$$



Example (cont'd)

now consider the following tableau proof

$$\neg((P \wedge (Q \rightarrow R \vee S)) \rightarrow P \vee Q)$$

$$P \wedge (Q \rightarrow R \vee S)$$

$$\neg(P \vee Q)$$

$$P$$

$$Q \rightarrow R \vee S$$

$$\neg P$$

$$\neg Q$$

Soundness and Completeness

Definitions

- a branch is closed if the formulas F and $\neg F$ occur on it
- if F is atomic, then the branch is said to be atomically closed
- a tableau is closed if every branch is closed
- a tableau proof of F is a closed tableau for $\neg F$
- in a strict tableau no formula is expanded twice on the same branch

Soundness and Completeness

Definitions

- a branch is closed if the formulas F and $\neg F$ occur on it
- if F is atomic, then the branch is said to be atomically closed
- a tableau is closed if every branch is closed
- a tableau proof of F is a closed tableau for $\neg F$
- in a strict tableau no formula is expanded twice on the same branch

Theorem

the tableau procedure is sound and complete:

$$F \text{ is a tautology} \iff F \text{ has a tableau proof}$$

Soundness and Completeness

Definitions

- a branch is closed if the formulas F and $\neg F$ occur on it
- if F is atomic, then the branch is said to be atomically closed
- a tableau is closed if every branch is closed
- a tableau proof of F is a closed tableau for $\neg F$
- in a strict tableau no formula is expanded twice on the same branch

Theorem

the tableau procedure is sound and complete:

$$F \text{ is a tautology} \iff F \text{ has a tableau proof}$$

Proof.

use next two lemmas; alternative proof of completeness: propositional model existence lemma ∎

Strong Propositional Completeness

Lemma

any application of a tableau expansion rule to a satisfiable tableau yields another satisfiable tableau

Strong Propositional Completeness

Lemma

any application of a tableau expansion rule to a satisfiable tableau yields another satisfiable tableau

Lemma

suppose F is a valid; a strict tableau construction for $\neg F$ that is continued as long as possible must terminate in an atomically closed tableau

Strong Propositional Completeness

Lemma

any application of a tableau expansion rule to a satisfiable tableau yields another satisfiable tableau

Lemma

suppose F is a valid; a strict tableau construction for $\neg F$ that is continued as long as possible must terminate in an atomically closed tableau

Proof.

see Computational Logic, this week ■

Implementation of Semantic Tableaux

Naive Approach

```
tableau_prover(X) :-
        expand([[neg X]],Y),
        closed(Y).
```

Implementation of Semantic Tableaux

Naive Approach

```
tableau_prover(X) :-
        expand([[neg X]],Y),
        closed(Y).
```

Slightly More Efficient

```
tableau_prover2(X) :-
        expand([[neg X]],Y),
        !,
        closed(Y).
```

Implementation of Semantic Tableaux

Naive Approach

```
tableau_prover(X) :-
        expand([[neg X]],Y),
        closed(Y).
```

Slightly More Efficient

```
tableau_prover2(X) :-
        expand([[neg X]],Y),
        !,
        closed(Y).
```

A Bit More Efficient

```
tableau_prover3(X) :-
        expand_and_close([[neg X]]).
```

First-Order Semantic Tableaux

Definition (uniform notation)

universal		existential	
γ	$\gamma(t)$	δ	$\delta(t)$
$\forall x A(x)$	$A(t)$	$\exists x A(x)$	$A(t)$
$\neg\exists x A(x)$	$\neg A(t)$	$\neg\forall x A(x)$	$\neg A(t)$

First-Order Semantic Tableaux

Definition (uniform notation)

universal		existential	
γ	$\gamma(t)$	δ	$\delta(t)$
$\forall x A(x)$	$A(t)$	$\exists x A(x)$	$A(t)$
$\neg\exists x A(x)$	$\neg A(t)$	$\neg\forall x A(x)$	$\neg A(t)$

Definition (tableau expansion rules)

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

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

Example

consider $\forall x(P(x) \vee Q(x)) \rightarrow \exists x P(x) \vee \forall x Q(x)$
we give a tableau proof

$$\neg\,(\forall x(P(x) \vee Q(x)) \rightarrow \exists x P(x) \vee \forall x Q(x)))$$

$$\forall x(P(x) \vee Q(x))$$

$$\neg(\exists x P(x) \vee \forall x Q(x))$$

$$\neg\exists x P(x)$$

$$\neg\forall x Q(x)$$

$$\neg Q(c)$$

$$\neg P(c)$$

$$P(c) \vee Q(c)$$

$$P(c) \qquad\qquad Q(c)$$

Soundness and Completeness of Tableau

Definitions

- a tableau proof of a sentence F is a closed tableau for $\neg F$
- a tableau branch is satisfiable if the set $\mathcal{G}$ of sentences on it is satisfiable, i.e., there exists a model of $\mathcal{G}$; a tableau is satisfiable if some branch is satisfiable

Soundness and Completeness of Tableau

Definitions

- a tableau proof of a sentence F is a closed tableau for $\neg F$
- a tableau branch is satisfiable if the set $\mathcal{G}$ of sentences on it is satisfiable, i.e., there exists a model of $\mathcal{G}$; a tableau is satisfiable if some branch is satisfiable

Theorem

if sentence F has a tableau proof, then F is valid

Proof.

if any tableau expansion rule is applied to a satisfiable tableau, the result is satisfiable ■

Soundness and Completeness of Tableau

Definitions

- a tableau proof of a sentence F is a closed tableau for $\neg F$
- a tableau branch is satisfiable if the set $\mathcal{G}$ of sentences on it is satisfiable, i.e., there exists a model of $\mathcal{G}$; a tableau is satisfiable if some branch is satisfiable

Theorem

if sentence F has a tableau proof, then F is valid

Proof.

if any tableau expansion rule is applied to a satisfiable tableau, the result is satisfiable ■

Theorem

if a sentence F is valid, then F has a tableau proof