Logik für Informatiker

2. Aussagenlogik

Teil 7

27.05.2014

Viorica Sofronie-Stokkermans

Universität Koblenz-Landau

e-mail: sofronie@uni-koblenz.de

Organisatorisches

Hauptklausur:

- Montag, 11.08.2014, 13:00-15:00, Raum D 028
- Anmeldung: Details via E-Mail

Nachklausur:

- Dienstag, 30.09.2014, 13:30-15:30, Raum D 028
- Anmeldung: Details via E-Mail

Bis jetzt

- Normalformen: CNF/DNF
- Subsumption
- SAT-Problem (Erfüllbarkeitsproblem)

  SAT; k-SAT; 3-SAT vs. SAT

  2-SAT: heute

  Horn-Formeln (Erfüllbarkeitstest)
- Der aussagenlogische Resolutionkalkül

Resolutionskalkül

Definition: Resolutionsregel (einzige Regel des Kalküls)

$$\frac{C_1 \cup \{P\} \qquad \{\neg P\} \cup C_2}{C_1 \cup C_2}$$

wobei

- P eine aussagenlogische Variable
- C_1, C_2 Klauseln – in Mengenschreibweise (können leer sein)

Definition:

$C_1 \cup C_2$ heißt Resolvente von $C_1 \cup \{P\}$, $C_2 \cup \{\neg P\}$

Beispiel

Klauselnormalform:

$$M = \{\{\neg P, Q\}, \{\neg Q, R\}, \{P\}, \{\neg R\}\}$$

(1)	$\{\neg P, Q\}$	gegeben
(2)	$\{\neg Q, R\}$	gegeben
(3)	$\{P\}$	gegeben
(4)	$\{\neg R\}$	gegeben
(5)	$\{Q\}$	aus (1) und (3)
(6)	$\{R\}$	aus (2) und (5)
(7)	$\perp$	aus (4) und (6)

Beweis für $\perp$ aus M (Widerspruch)

$$M \vdash_{\mathrm{Res}} \perp$$

Resolution

Ziele:

- Formalisieren, was $M \vdash_{\mathsf{Res}} C$ bedeutet

- Zeigen, dass $M \vdash_{\mathsf{Res}} \perp$ gdw. M unerfüllbar.

  Korrektheit: Falls $M \vdash_{\mathsf{Res}} \perp$, so M unerfüllbar.

  Vollständigkeit: Falls M unerfüllbar, so $M \vdash_{\mathsf{Res}} \perp$

- Zeigen, dass das Verfahren terminiert.

Resolution

Ziele:

- Formalisieren, was $M \vdash_{\text{Res}} C$ bedeutet

- Zeigen, dass $M \vdash_{\text{Res}} \perp$ gdw. M unerfüllbar.

  Korrektheit: Falls $M \vdash_{\text{Res}} \perp$, so M unerfüllbar.

  Vollständigkeit: Falls M unerfüllbar, so $M \vdash_{\text{Res}} \perp$

- Zeigen, dass das Verfahren terminiert.

Resolution

Sei F eine Klauselmenge und

$$\text{Res}(F) = F \cup \{R \mid R \text{ ist eine Resolvente zweier Klauseln aus } F\}$$

$\text{Res}^0(F) = F$

$\text{Res}^{n+1}(F) = \text{Res}(\text{Res}^n(F))$

$\text{Res}^\star(F) = \bigcup_{n\in\mathbb{N}} \text{Res}^n(F)$

(bezeichnet die Vereinigung der Resolventen aus aller möglichen Resolutionsschritte auf F)

Notation: Falls $C \in \text{Res}^\star(F)$, so schreiben wir $F \vdash_{\text{Res}} C$.

Definition: Beweis für C (aus F): $C_1, \ldots C_n$, wobei:
$C_n = C$ und für alle $1 \leq i \leq n$: ($C_i \in F$ oder C_i Resolvente für $\frac{C_{j_1} \quad C_{j_2}}{C_i}$ mit $j_1, j_2 < i$).

Resolution

Ziele:

- Formalisieren, was $M \vdash_{\text{Res}} C$ bedeutet

- Zeigen, dass $M \vdash_{\text{Res}} \bot$ gdw. M unerfüllbar.

  Korrektheit: Falls $M \vdash_{\text{Res}} \bot$, so M unerfüllbar.

  Vollständigkeit: Falls M unerfüllbar, so $M \vdash_{\text{Res}} \bot$

- Zeigen, dass das Verfahren terminiert.

Resolution: Korrektheit und Vollständigkeit

Theorem (Korrektheit)
Für eine Menge M von Klauseln gilt: Falls $M \vdash_{\mathsf{Res}} \perp$, so M unerfüllbar.

Falls aus M die leere Klausel durch Resolution herleitbar ist, ist M unerfüllbar (es gibt keine Wertebelegung $\mathcal{A}$ in der alle Klauseln in M wahr sind).

Äquivalent:

Falls M erfüllbar ist, ist die leere Klausel durch Resolution nicht herleitbar.

Theorem (Vollständigkeit)
Für eine Menge M von Klauseln gilt: Falls M unerfüllbar, so $M \vdash_{\mathsf{Res}} \perp$.

Falls M unerfüllbar ist (d.h. es gibt keine Wertebelegung $\mathcal{A}$ in der alle Klauseln in M wahr sind), ist die leere Klausel aus M durch Resolution herleitbar.

Äquivalent:

Falls aus M die leere Klausel durch Resolution nicht herleitbar ist, ist M erfüllbar.

Resolution: Korrektheit

Theorem (Korrektheit)
Für eine Menge M von Klauseln gilt: Falls $M \vdash_{\mathsf{Res}} \perp$, so M unerfüllbar.

Resolution: Korrektheit

Theorem (Korrektheit)
Für eine Menge M von Klauseln gilt: Falls $M \vdash_{\text{Res}} \bot$, so M unerfüllbar.

Beweis

(1) Wir zeigen, dass falls $C \in \text{Res}^*(M)$, so $M \equiv M \cup \{C\}$.
(Theorem auf Seite 14). (NB: $M \cup \{C\}$ Notation für $M \wedge C$.)

(2) Es folgt, dass falls $M \vdash_{\text{Res}} \bot$, so $\bot \in \text{Res}^*(M)$, d.h. $M \equiv M \cup \{\bot\}$.

(3) Aber $M \cup \{\bot\}$ ist unerfüllbar, deshalb ist auch M unerfüllbar.

Resolution: Korrektheit

Theorem (Korrektheit)
Für eine Menge M von Klauseln gilt: Falls $M \vdash_{\text{Res}} \bot$, so M unerfüllbar.

Beweis

(1) Wir zeigen, dass falls $C \in \text{Res}^*(M)$, so $M \equiv M \cup \{C\}$.
(Theorem auf Seite 14). (NB: $M \cup \{C\}$ Notation für $M \wedge C$.)

(2) Es folgt, dass falls $M \vdash_{\text{Res}} \bot$, so $\bot \in \text{Res}^*(M)$, d.h. $M \equiv M \cup \{\bot\}$.

(3) Aber $M \cup \{\bot\}$ ist unerfüllbar, deshalb ist auch M unerfüllbar.

Erklärung 1: Falls $M = \{C_1, \dots, C_n\}$ so $M \cup \{\bot\}$ ist eine Notation für $C_1 \wedge \dots \wedge C_n \wedge \bot$. Aber $C_1 \wedge \dots \wedge C_n \wedge \bot \equiv \bot$, so ist $C_1 \wedge \dots \wedge C_n \wedge \bot$ unerfüllbar (also auch $M \cup \{\bot\}$). Da $M \equiv M \cup \{\bot\}$, ist auch M unerfüllbar.

Erklärung 2: Es gibt keine Wertebelegung $\mathcal{A}$ die alle Klauseln in $M \cup \{\bot\}$ wahr macht (d.h. so dass: $\mathcal{A}(D) = 1$ für alle Klauseln D in M und $\mathcal{A}(\bot) = 1$).

Resolution: Korrektheit

(1) Wir zeigen, dass falls $C \in \text{Res}^*(M)$, so $M \equiv M \cup \{C\}$.

Lemma: $C_1 \vee P, C_2 \vee \neg P \models C_1 \vee C_2$

Beweis:

Sei $\mathcal{A}$ Interpretation mit $\mathcal{A}(C_1 \vee P) = 1$ und $\mathcal{A}(C_2 \vee \neg P) = 1$.

Zu zeigen: $\mathcal{A}(C_1 \vee C_2) = 1$.

Fall 1: $\mathcal{A}(C_1) = 1$. Dann $\mathcal{A}(C_1 \vee C_2) = 1$.

Fall 2: $\mathcal{A}(C_1) = 0$. Dann $\mathcal{A}(P) = 1$.
Da $\mathcal{A}(C_2 \vee \neg P) = 1$, so $\mathcal{A}(C_2) = 1$, d.h. $\mathcal{A}(C_1 \vee C_2) = 1$.

Resolution: Korrektheit

Theorem: Falls $C \in \mathrm{Res}^\star(F)$, so $F \equiv F \cup \{C\}$.

Beweis: Annahme: $C \in \mathrm{Res}^n(F)$. Zu zeigen: $F \equiv F \wedge C$, i.e.:

Für alle $\mathcal{A} : \Pi \to \{0, 1\}$ gilt: $\mathcal{A}(F) = 1$ gdw. $(\mathcal{A}(F) = 1$ und $\mathcal{A}(C) = 1)$.

Resolution: Korrektheit

Theorem: Falls $C \in \text{Res}^\star(F)$, so $F \equiv F \cup \{C\}$.

Beweis: Annahme: $C \in \text{Res}^n(F)$. Zu zeigen: $F \equiv F \wedge C$, i.e.:
Für alle $\mathcal{A} : \Pi \to \{0,1\}$ gilt: $\mathcal{A}(F) = 1$ gdw. $(\mathcal{A}(F) = 1$ und $\mathcal{A}(C) = 1)$.

- Sei $\mathcal{A}{:}\Pi{\to}\{0,1\}$ mit $\mathcal{A}(F){=}1$ (d.h. $\mathcal{A}(C){=}1$ für jede Klausel $C{\in}F$).
  Zu zeigen: Für alle $m \in \mathbb{N}$ gilt p(m): Falls $D \in \text{Res}^m(F)$, so $\mathcal{A}(D) = 1$.
  Beweis durch Induktion:
  IB: $m = 0$: Sei $D \in \text{Res}^0(F) = F$. Dann $\mathcal{A}(D) = 1$.
  IV: Annahme: $p(m)$ is wahr. **IS:** Zu zeigen: $p(m+1)$ wahr.
  Sei $D \in \text{Res}^{m+1}(F)$. Fall 1: $D \in \text{Res}^m(F)$. Nach IV, $\mathcal{A}(D) = 1$.
  Fall 2: $D{=}C_1{\vee}C_2$ Resolvente von $C_1{\vee}P$, $C_2{\vee}\neg P \in \text{Res}^m(F)$. Nach IV:
  $\mathcal{A}(C_1{\vee}P){=}\mathcal{A}(C_2{\vee}\neg P){=}1$. Da $C_1{\vee}P, C_2{\vee}\neg P \models C_1{\vee}C_2$ folgt $\mathcal{A}(D){=}1$.

  Dann $\mathcal{A}(C) = 1$, so $(\mathcal{A}(F) = 1$ und $\mathcal{A}(C) = 1)$, d.h. $\mathcal{A}(F \wedge C) = 1$.
- Sei $\mathcal{A} : \Pi \to \{0,1\}$ mit $\mathcal{A}(F \wedge C) = 1$.
  Dann $\mathcal{A}(F) = 1$ und $\mathcal{A}(C) = 1$, so $\mathcal{A}(F) = 1$.

Resolution: Vollständigkeit

Theorem.
Für jede endliche Menge M von Klauseln gilt:
falls M unerfüllbar, so $M \vdash_{\mathsf{Res}} \bot$.

Resolution: Vollständigkeit

Theorem.
Für jede endliche Menge M von Klauseln gilt:
falls M unerfüllbar, so $M \vdash_{\mathsf{Res}} \bot$.

Beweis: Induktion:

$p(n)$: Sei M Menge von Klauseln mit n Aussagenvariablen.
Falls M unerfüllbar, so $M \vdash_{\mathsf{Res}} \bot$

Resolution: Vollständigkeit

Theorem.
Für jede endliche Menge M von Klauseln gilt:
falls M unerfüllbar, so $M \vdash_{\text{Res}} \perp$.

Beweis: Induktion:

$p(n)$: Sei M Menge von Klauseln mit n Aussagenvariablen.
Falls M unerfüllbar, so $M \vdash_{\text{Res}} \perp$

Induktionsbasis: $n = 0$.
Dann $M = \{\perp\}$, d.h. $M \vdash_{\text{Res}} \perp$

Resolution: Vollständigkeit

Theorem.
Für jede endliche Menge M von Klauseln gilt:
falls M unerfüllbar, so $M \vdash_{\mathsf{Res}} \bot$.

Beweis: Induktion:

$p(n)$: Sei M Menge von Klauseln mit n Aussagenvariablen.
Falls M unerfüllbar, so $M \vdash_{\mathsf{Res}} \bot$

Induktionsbasis: $n = 0$. Dann $M = \{\bot\}$, d.h. $M \vdash_{\mathsf{Res}} \bot$

Induktionsvoraussetzung: $p(n)$ gilt.
Induktionsschritt: Beweise, dass $p(n+1)$ gilt.

Sei M Menge von Klauseln mit Aussagenvariablen $\{P_1, \dots, P_n, P_{n+1}\}$.

M_0 sei aus M entstanden durch Ersetzung von P_{n+1} durch $\bot$
M_1 sei aus M entstanden durch Ersetzung von P_{n+1} durch $\top$

Resolution: Vollständigkeit

M_0 sei aus M entstanden durch Ersetzung von P_{n+1} durch $\bot$:

- P_{n+1} wird aus allen Klauseln gelöscht,
- Klauseln, die $\neg P_{n+1}$ enthalten werden ebenfalls gelöscht

M_1 sei aus M entstanden durch Ersetzung von P_{n+1} durch $\top$:

- $\neg P_{n+1}$ wird aus allen Klauseln gelöscht,
- Klauseln, die P_{n+1} enthalten werden ebenfalls gelöscht

Fakten:

- M_0, M_1 enthalten nur Aussagenvariablen $\{P_1, \ldots, P_n\}$
- M_0, M_1 unerfüllbar

Resolution: Vollständigkeit

M_0 sei aus M entstanden durch Ersetzung von P_{n+1} durch $\perp$:
- P_{n+1} wird aus allen Klauseln gelöscht,
- Klauseln, die $\neg P_{n+1}$ enthalten werden ebenfalls gelöscht

M_1 sei aus M entstanden durch Ersetzung von P_{n+1} durch $\top$:
- $\neg P_{n+1}$ wird aus allen Klauseln gelöscht,
- Klauseln, die P_{n+1} enthalten werden ebenfalls gelöscht

Fakten:

- M_0, M_1 enthalten nur Aussagenvariablen $\{P_1, \ldots, P_n\}$
- M_0, M_1 unerfüllbar

Induktionsvoraussetzung:

$M_0 \vdash_{\text{Res}} \perp$, i.e. es gibt $C_1, C_2, \ldots, C_m$ Beweis (aus M_0) für $\perp$

$M_1 \vdash_{\text{Res}} \perp$, i.e. es gibt $D_1, D_2, \ldots, D_k$ Beweis (aus M_1) für $\perp$

Resolution: Vollständigkeit

M_0 sei aus M entstanden durch Ersetzung von P_{n+1} durch $\bot$:
- P_{n+1} wird aus allen Klauseln gelöscht,
- Klauseln, die $\neg P_{n+1}$ enthalten werden ebenfalls gelöscht

M_1 sei aus M entstanden durch Ersetzung von P_{n+1} durch $\top$:
- $\neg P_{n+1}$ wird aus allen Klauseln gelöscht,
- Klauseln, die P_{n+1} enthalten werden ebenfalls gelöscht

Fakten:

- M_0, M_1 enthalten nur Aussagenvariablen $\{P_1, \dots, P_n\}$
- M_0, M_1 unerfüllbar

Induktionsvoraussetzung:

$M_0 \vdash_{\text{Res}} \bot$, i.e. es gibt $C_1, C_2, \dots, C_m$ Beweis (aus M_0) für $\bot$

P_{n+1} zurück: $C'_1, C'_2, \dots, C'_m$ Beweis (aus M) für $\bot$ oder P_{n+1}

$M_1 \vdash_{\text{Res}} \bot$, i.e. es gibt $D_1, D_2, \dots, D_k$ Beweis (aus M_1) für $\bot$

$\neg P_{n+1}$ zurück: $D'_1, D'_2, \dots, D'_k$ Beweis (aus M) für $\bot$ oder $\neg P_{n+1}$

Resolution: Vollständigkeit

M_0 sei aus M entstanden durch Ersetzung von P_{n+1} durch $\bot$:
- P_{n+1} wird aus allen Klauseln gelöscht,
- Klauseln, die $\neg P_{n+1}$ enthalten werden ebenfalls gelöscht

M_1 sei aus M entstanden durch Ersetzung von P_{n+1} durch $\top$:
- $\neg P_{n+1}$ wird aus allen Klauseln gelöscht,
- Klauseln, die P_{n+1} enthalten werden ebenfalls gelöscht

Fakten:

- M_0, M_1 enthalten nur Aussagenvariablen $\{P_1, \ldots, P_n\}$
- M_0, M_1 unerfüllbar

Induktionsvoraussetzung:

$$\left.\begin{array}{l}
M_0 \vdash_{\mathrm{Res}} \bot, \text{ i.e. es gibt } C_1, C_2, \ldots, C_m \text{ Beweis (aus } M_0\text{) für } \bot \\
\quad P_{n+1} \text{ zurück: } C'_1, C'_2, \ldots, C'_m \text{ Beweis (aus } M\text{) für } \bot \text{ oder } P_{n+1} \\
M_1 \vdash_{\mathrm{Res}} \bot, \text{ i.e. es gibt } D_1, D_2, \ldots, D_k \text{ Beweis (aus } M_1\text{) für } \bot \\
\quad \neg P_{n+1} \text{ zurück: } D'_1, D'_2, \ldots, D'_k \text{ Beweis (aus } M\text{) für } \bot \text{ oder } \neg P_{n+1}
\end{array}\right\} \Rightarrow \quad M \vdash_{\mathrm{Res}} \bot$$

Resolution: Vollständigkeit

Theorem.
Für jede endliche Menge M von Klauseln gilt:
falls M unerfüllbar, so $M \vdash_{\mathsf{Res}} \perp$.

Es gilt auch:

Theorem.
Für jede Menge M von Klauseln gilt:
falls M unerfüllbar, so $M \vdash_{\mathsf{Res}} \perp$.

Terminierung

Theorem.
Aussagenlogische Resolution (für Klauselmengen in Mengennotation) terminiert, für jede endliche Menge von Klauseln.

Terminierung

Theorem.
Aussagenlogische Resolution (für Klauselmengen in Mengennotation) terminiert, für jede endliche Menge von Klauseln.

Beweis: Es gibt nicht mehr als 2^{2n} Klauseln (in Mengennotation) mit n Aussagenvariablen.

$\mapsto$ nicht mehr als $(2^{2n})^2$ mögliche Anwendungen der Resolutionsregel.

2-SAT

Theorem

Erfüllbarkeit für Formeln in 2-KNF (2SAT) ist polynomiell entscheidbar

Beweis (Krom, 1967)

Fall 1: Für jede Aussagenvariable P, entweder enthalten alle Klauseln P oder $\neg P$: erfüllbar.

Fall 2: Es gibt Klauseln $C_1 = L_1 \vee P$, $C_2 = L_2 \vee \neg P$

Resolutionschritt; Resolvente $L_1 \vee L_2$ (Mengennotation)

Fakt: Resolventen sind immer auch in 2-KNF.

Wenn F n Aussagenvariablen enthält, gibt es $2n$ mögliche Literale, und $\leq 4n^2$ nicht-leere verschiedene 2-KNF Klauseln.

F ist erfüllbar gdw. $\perp \not\in \text{Res}^*(F)$

Wenn wir $\text{Res}^*(F)$ berechnen: nicht mehr als $(4n^2)^2$ Resolutionsschritte notwendig.

$\mapsto$ Erfüllbarkeit von F ist polynomiell entscheidbar.

1-Resolution

Die 1-Resolution (unit resolution) benutzt dieselbe Notation wie im Resolutionskalkül. Die 1-Resolutionsregel ist ein Spezialfall der allgemeinen Resolutionsregel:

$$\frac{\{P\} \quad \{\neg P\} \cup C}{C} \qquad \frac{\{\neg P\} \quad \{P\} \cup C}{C}$$

Der 1-Resolutionskalkül ist nicht vollständig.

Die Klauselmenge

$$M = \{\{P_1, P_2\}, \{P_1, \neg P_2\}, \{\neg P_1, P_2\}, \{\neg P_1, \neg P_2\}\}$$

ist nicht erfüllbar, aber mit 1-Resolution ist aus M nichts ableitbar, also auch nicht $\perp$.

Unser Ziel

Kalkül(e) zur systematischen Überprüfung von Erfüllbarkeit

(für Formeln und/oder Formelmengen)

1. Formeln in KNF (Mengen von Klauseln)

   Resolution — bis jetzt

2. Formelmengen

   Semantische Tableaux

Unser Ziel

Kalkül(e) zur systematischen Überprüfung von Erfüllbarkeit

(für Formeln und/oder Formelmengen)

1. Formeln in KNF (Mengen von Klauseln)

   Resolution

2. Formelmengen

   Semantische Tableaux

Der aussagenlogische Tableaukalkül

Wesentliche Eigenschaften

- Widerlegungskalkül: Testet auf Unerfüllbarkeit
- Beweis durch Fallunterscheidung
- Top-down-Analyse der gegebenen Formeln

Der aussagenlogische Tableaukalkül

Vorteile

- Intuitiver als Resolution
- Formeln müssen nicht in Normalform sein
- Falls Formelmenge erfüllbar ist (Test schlägt fehl), wird ein Gegenbeispiel (eine erfüllende Interpretation) konstruiert

Der aussagenlogische Tableaukalkül

Vorteile

- Intuitiver als Resolution
- Formeln müssen nicht in Normalform sein
- Falls Formelmenge erfüllbar ist (Test schlägt fehl), wird ein Gegenbeispiel (eine erfüllende Interpretation) konstruiert

Nachteile

- Mehr als eine Regel

Formeltypen

Konjunktive Formeln: Typ α

- $\neg\neg F$
- $F \land G$
- $\neg(F \lor G)$
- $\neg(F \to G)$

Formeltypen

Konjunktive Formeln: Typ α

- $\neg\neg F$
- $F \land G$
- $\neg(F \lor G)$
- $\neg(F \rightarrow G)$

Disjunktive Formeln: Typ β

- $\neg(F \land G)$
- $F \lor G$
- $F \rightarrow G$

Formeltypen

Konjunktive Formeln: Typ α

- $\neg\neg F$
- $F \wedge G$
- $\neg(F \vee G)$
- $\neg(F \rightarrow G)$

Zuordnungsregeln Formeln / Unterformeln

α	α_1	α_2
$F \wedge G$	F	G
$\neg(F \vee G)$	$\neg F$	$\neg G$
$\neg(F \rightarrow G)$	F	$\neg G$
$\neg\neg F$	F	

Formeltypen

Disjunktive Formeln: Typ β

- $\neg(F \wedge G)$
- $F \vee G$
- $F \rightarrow G$

Zuordnungsregeln Formeln / Unterformeln

β	β_1	β_2
$\neg(F \wedge G)$	$\neg F$	$\neg G$
$F \vee G$	F	G
$F \rightarrow G$	$\neg F$	G

Regeln des (aussagenlogischen) Tableaukalküls

$$\frac{\alpha}{\begin{array}{l}\alpha_1 \\ \alpha_2\end{array}}$$

Regeln des (aussagenlogischen) Tableaukalküls

$$\frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}}$$

Konjunktiv

$$\begin{array}{c} p \wedge q \\ | \\ p \\ | \\ q \end{array}$$

Regeln des (aussagenlogischen) Tableaukalküls

$$\frac{\alpha}{\begin{array}{l}\alpha_1 \\ \alpha_2\end{array}}$$

Konjunktiv

$$\begin{array}{c} p \wedge q \\ | \\ p \\ | \\ q \end{array}$$

$$\frac{\beta}{\beta_1 \mid \beta_2}$$

Disjunktiv

$$\begin{array}{ccc} & p \vee q & \\ / & & \backslash \\ p & & q \end{array}$$

Regeln des (aussagenlogischen) Tableaukalküls

$$\frac{\alpha}{\begin{array}{c}\alpha_1 \\ \alpha_2\end{array}}$$

Konjunktiv

$$\begin{array}{c} p \wedge q \\ | \\ p \\ | \\ q \end{array}$$

$$\frac{\beta}{\beta_1 \mid \beta_2}$$

Disjunktiv

$$\begin{array}{ccc} & p \vee q & \\ / & & \backslash \\ p & & q \end{array}$$

$$\frac{\begin{array}{c}\phi \\ \neg\phi\end{array}}{\perp}$$

Widerspruch

$$\begin{array}{c} \phi \\ \neg\phi \\ | \\ \perp \end{array}$$

Instanzen der α und β-Regel

Instanzen der α-Regel

$$\frac{P \wedge Q}{\begin{array}{c}P\\Q\end{array}} \qquad \frac{\neg(P \vee Q)}{\begin{array}{c}\neg P\\\neg Q\end{array}} \qquad \frac{\neg(P \rightarrow Q)}{\begin{array}{c}P\\\neg Q\end{array}} \qquad \frac{\neg\neg P}{P}$$

Instanzen der α und β-Regel

Instanzen der α-Regel

$$\frac{P \wedge Q}{\begin{array}{c}P\\Q\end{array}} \qquad \frac{\neg(P \vee Q)}{\begin{array}{c}\neg P\\\neg Q\end{array}} \qquad \frac{\neg(P \rightarrow Q)}{\begin{array}{c}P\\\neg Q\end{array}} \qquad \frac{\neg\neg P}{P}$$

Instanzen der β-Regel

$$\frac{P \vee Q}{P \mid Q} \qquad \frac{\neg(P \wedge Q)}{\neg P \mid \neg Q} \qquad \frac{P \rightarrow Q}{\neg P \mid Q}$$

Ein Tableau für $\{P \wedge \neg(Q \vee \neg R),\ \neg Q \vee \neg R\}$

1.(1) $P \wedge \neg(Q \vee \neg R)$

(2) $\neg Q \vee \neg R$

(3)	$\neg Q$	(4)	$\neg R$
(5)	P	(10)	P
(6)	$\neg(Q \vee \neg R)$	(11)	$\neg(Q \vee \neg R)$
(7)	$\neg Q$		
(8)	$\neg\neg R$		
(9)	R		

Dieses Tableau ist nicht "maximal", aber der erste "Ast" ist.
Dieser Ast ist nicht "geschlossen" (enthält keinen Widerspruch), also ist die Menge $\{1, 2\}$ erfüllbar.
(Diese Begriffe werden auf den nächsten Seiten erklärt.)

Determinismus von Kalkül und Regeln

Determinismus

- Die Regeln sind alle deterministisch

Determinismus von Kalkül und Regeln

Determinismus

- Die Regeln sind alle deterministisch
- Der Kalkül aber nicht:
  Auswahl der nächsten Formel, auf die eine Regel angewendet wird

Determinismus von Kalkül und Regeln

Determinismus

- Die Regeln sind alle deterministisch
- Der Kalkül aber nicht:
  Auswahl der nächsten Formel, auf die eine Regel angewendet wird

Heuristik

- Nicht-verzweigende Regeln zuerst: α vor β

Determinismus von Kalkül und Regeln

Determinismus

- Die Regeln sind alle deterministisch
- Der Kalkül aber nicht:
  Auswahl der nächsten Formel, auf die eine Regel angewendet wird

Heuristik

- Nicht-verzweigende Regeln zuerst: α vor β

Nota bene:

Dieselbe Formel kann mehrfach (auf verschiedenen Ästen) verwendet werden

Beispiel

Zu zeigen:

$$(P \to (Q \to R)) \to ((P \vee S) \to ((Q \to R) \vee S)) \text{ allgemeingültig}$$

Beispiel

Zu zeigen:

$$(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \vee S) \rightarrow ((Q \rightarrow R) \vee S)) \text{ allgemeingültig}$$

Wir zeigen, dass

$$\neg[(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \vee S) \rightarrow ((Q \rightarrow R) \vee S))]$$

unerfüllbar ist.

Beispiel

(1) $\neg[(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \vee S) \rightarrow ((Q \rightarrow R) \vee S))]$

(2) $(P \rightarrow (Q \rightarrow R))$ α $[1_1]$

(3) $\neg((P \vee S) \rightarrow ((Q \rightarrow R) \vee S))$ α $[1_2]$

(4) $P \vee S$ α $[3_1]$

(5) $\neg((Q \rightarrow R) \vee S))$ α $[3_2]$

(6) $\neg(Q \rightarrow R)$ α $[5_1]$

(7) $\neg S$ α $[5_2]$

(8) $\neg P$ β $[2_1]$ (9) $Q \rightarrow R$ β $[2_2]$

(10) P β $[4_1]$ (11) S β $[4_2]$

geschlossenes Tableau.

Zusammenfassung

- Resolution

  Korrektheit und Vollständigkeit

- Semantische Tableaux