TECHNISCHE UNIVERSITÄT DRESDEN

FORMALE SYSTEME

23. Vorlesung: Logisches Schließen

Markus Krötzsch
Professur für Wissensbasierte Systeme

TU Dresden, 15. Januar 2018

Logik: Glossar

Atom	kleinste mögliche Formel	$p \in \mathbf{P}$
Teilformel	Unterausdruck, der Formel ist	$\mathrm{Sub}(F)$
Wertzuweisung	Funktion	$w : \mathbf{P} \to \{1, 0\}$
Modell	erfüllende Wertzuweisung	$w \models F$ / $w(F) = 1$
Literal	Atom oder negiertes Atom	p oder $\neg p$
Klausel	Disjunktion von Literalen	$L_1 \vee \ldots \vee L_n$
Monom	Konjunktion von Literalen	$L_1 \wedge \ldots \wedge L_n$
logische Konsequenz	Teilmengenbeziehung der Modellmengen; entspricht $\to$	$F \models G$ oder $\mathcal{F} \models G$
semantische Äquivalenz	Gleichheit der Modellmengen; entspricht $\leftrightarrow$	$F \equiv G$ oder $\mathcal{F} \equiv \mathcal{G}$
Tautologie	allgemeingültige Formel	$\models F$

Vereinfachungen und Normalformen

Jede Formel ist semantisch äquivalent zu einer, die nur folgende Junktoren verwendet:

- $\wedge$ und $\neg$
- $\vee$ und $\neg$
- und verschiedene andere Kombinationen . . . (z.B. $\to$ und $\neg$; oder auch $\uparrow$)

Normalformen

- Negationsnormalform
  stets linear
- Konjunktive Normalform
  im schlimmsten Fall exponentiell; mit Hilfsatomen linear
- Disjunktive Normalform
  im schlimmsten Fall exponentiell

Logisches Schließen

Logisches Schließen

Allgemein umfasst logisches Schließen jede Berechnung, bei der semantische Eigenschaften von Formeln ermittelt werden.

Vielfach hat man es mit Entscheidungsproblemen zu tun, z.B.:

Schlussfolgerung:
- **Eingabe:** endliche Formelmenge $\mathcal{F}$, Formel G
- **Ausgabe:** „ja“ wenn $\mathcal{F} \models G$ gilt; sonst „nein.“

Allgemeingültigkeit:
- **Eingabe:** Formel F
- **Ausgabe:** „ja“ wenn F allgemeingültig ist; sonst „nein.“

Unerfüllbarkeit:
- **Eingabe:** Formel F
- **Ausgabe:** „ja“ wenn F unerfüllbar ist; sonst „nein.“

Viele logische Probleme – viele Algorithmen?

Oft kann man unterschiedliche Probleme leicht ineinander umwandeln:

- $\mathcal{F} \models G$ (für endliche $\mathcal{F}$) gdw. $\bigwedge_{F\in\mathcal{F}} F \to G$ allgemeingültig gdw. $\bigwedge_{F\in\mathcal{F}} F \wedge \neg G$ unerfüllbar
- F allgemeingültig gdw. $\emptyset \models F$ gdw. $\neg F$ unerfüllbar
- F unerfüllbar gdw. $F \models \bot$ gdw. $\neg F$ allgemeingültig

$\leadsto$ Jedes Verfahren, welches eines dieser Probleme lösen kann, lässt sich prinzipiell auch zur Lösung der anderen einsetzen.

(Es gibt auch Probleme des logischen Schließens, für die das höchstwahrscheinlich nicht zutrifft, z.B. „Gegeben eine Formel F, entscheide ob es eine kürzere Formel gibt, die zu F äquivalent ist.“)

Schließen mit Wahrheitswertetabellen

Die genannten Probleme sind mittels Wahrheitswertetabelle lösbar:

- Allgemeingültigkeit: Ist der Wert von F für jede Belegung 1?
- Unerfüllbarkeit: Ist der Wert von F für jede Belegung 0?
- Schlussfolgerung: Ist der Wert von G für jede Belegung 1, unter der alle Formeln von $\mathcal{F}$ den Wert 1 annehmen?

Nachteil: Die Tabelle hat 2^n Zeilen, wenn die betrachteten Formeln n Atome haben

Beispiel: Für die Modellierung von Sudoku nutzen wir $9 \times 9 \times 9$ Atome. Die entsprechende Tabelle hat also 2^{729} (ca. 2.8×10^{219}) Zeilen. Zum Vergleich:
- Anzahl der Neuronen im menschlichen Gehirn: ca. 1.5×10^{14}
- Jemals digital gespeicherte Daten in Bytes: $<< 10^{24}$
- Nanosekunden bis der Kern der Sonne ausbrennt: ca. 10^{29}
- Anzahl der Atome im beobachtbaren Universum: $10^{78} - 10^{82}$

Schließen mit DNF

Man kann (Un)Erfüllbarkeit leicht aus DNF ablesen:

Satz: Eine Formel in DNF ist genau dann erfüllbar, wenn eines ihrer Monome erfüllbar ist. Dies ist genau dann der Fall, wenn das Monom kein Atom gleichzeitig negiert und nichtnegiert enthält.

Mögliches Verfahren:
- Führe logisches Problem auf (Un)Erfüllbarkeit zurück
- Bilde die DNF und prüfe jedes Monom auf komplementäre Literale

Nachteil: Die DNF kann ebenfalls exponentiell groß werden (muss sie aber nicht in jedem Fall)

Schließen mit KNF

Man kann Widerlegbarkeit/Allgemeingültigkeit aus KNF ablesen:

Satz: Eine Formel in KNF ist genau dann widerlegbar, wenn eine ihrer Klauseln widerlegbar ist. Dies ist genau dann der Fall, wenn die Klausel kein Atom gleichzeitig negiert und nichtnegiert enthält.

Mögliches Verfahren:

- Führe logisches Problem auf Allgemeingültigkeit zurück
- Bilde die KNF und prüfe jede Klausel auf komplementäre Literale

Nachteil: Die KNF kann ebenfalls exponentiell groß werden (muss sie aber nicht in jedem Fall)

Können wir durch Hilfsatome eine kleinere KNF erzeugen?
Ja, aber dabei wird nur Erfüllbarkeit erhalten, nicht Allgemeingültigkeit!
$\rightsquigarrow$ nicht geeignet, um das Problem effizienter zu lösen

Resolution

Resolution

Ziel: Verfahren, um (Un)Erfüllbarkeit einer KNF zu bestimmen

Leider ist (Un)Erfüllbarkeit nicht an einzelnen Klauseln erkennbar.

Beispiel: Die Logelei aus Vorlesung 21 haben wir mit den Formeln $\mathcal{F} = \{A \leftrightarrow \neg B, B \leftrightarrow \neg C, C \leftrightarrow (\neg A \wedge \neg B)\}$ dargestellt. Wir wollen beweisen, dass Barbara die Wahrheit sagt, also $\mathcal{F} \models B$. Dazu kann man zeigen, dass $\mathcal{F} \cup \{\neg B\}$ unerfüllbar ist. Bei Anwendung des Distributivgesetztes erhalten wir die folgende KNF:

$$\begin{aligned}(\neg A \vee \neg B) \wedge (A \vee B) \wedge \\ (\neg B \vee \neg C) \wedge (B \vee C) \wedge \\ (\neg C \vee \neg A) \wedge (A \vee B \vee C) \wedge \\ \neg B\end{aligned}$$

Die Klauselform

Wenn man grundsätzlich mit KNF arbeitet, dann bieten sich einige syntaktische Vereinfachungen an:

- Eine Klausel $L_1 \vee \ldots \vee L_n$ wir dargestellt als Menge $\{L_1, \ldots, L_n\}$
- Eine Konjunktion von Klauseln $K_1 \wedge \ldots \wedge K_\ell$ wird dargestellt als Menge $\{K_1, \ldots, K_\ell\}$

Eine Menge von Mengen von Literalen unter dieser Interpretation heißt Klauselform.

Beispiel: Die KNF

$$\begin{aligned}(\neg A \vee \neg B) \wedge (A \vee B) \wedge \\ (\neg B \vee \neg C) \wedge (B \vee C) \wedge \\ (\neg C \vee \neg A) \wedge (A \vee B \vee C) \wedge \\ \neg B\end{aligned}$$

kann in Klauselform dargestellt werden als:

$$\{\{\neg A, \neg B\}, \{A, B\}, \{\neg B, \neg C\}, \{B, C\}, \{\neg C, \neg A\}, \{A, B, C\}, \{\neg B\}\}$$

Ableitungen mittels Resolution

Beim Resolutionsverfahren leiten wir schrittweise neue Klauseln aus den gegebenen ab.

Ein einzelner Resolutionschritt funktioniert wie folgt:

Gegeben seien zwei Klauseln K_1 und K_2 für die es ein Atom $p \in \mathbf{P}$ gibt mit $p \in K_1$ und $\neg p \in K_2$. Die Resolvente von K_1 und K_2 bezüglich p ist die Klausel $(K_1 \setminus \{p\}) \cup (K_2 \setminus \{\neg p\})$.

Eine Klausel R ist eine Resolvente einer Klauselmenge $\mathcal{K}$ wenn R Resolvente zweier Klauseln $K_1, K_2 \in \mathcal{K}$ ist.

Beispiel: Resolventen für die Menge $\{\{\neg A, \neg B\}, \{A, B\}, \{\neg B, \neg C\}, \{B, C\}, \{\neg C, \neg A\}, \{A, B, C\}, \{\neg B\}\}$ sind z.B.

- $\{B, \neg A\}$ aus den Klauseln $\{B, C\}$ und $\{\neg C, \neg A\}$
- $\{A\}$ aus den Klauseln $\{A, B\}$ und $\{\neg B\}$
- $\{B, \neg B\}$ aus den Klauseln $\{B, C\}$ und $\{\neg B, \neg C\}$

Bedeutung von Resolutionsschritten

Wir betrachten Klauseln $\{L_1, \ldots, L_n, p\}$ und $\{\neg p, M_1, \ldots, M_\ell\}$:

- $\{L_1, \ldots, L_n, p\} \equiv (L_1 \vee \ldots \vee L_n \vee p) \equiv (\neg L_1 \wedge \ldots \wedge \neg L_n) \to p$
- $\{\neg p, M_1, \ldots, M_\ell\} \equiv (\neg p \vee M_1 \vee \ldots \vee M_\ell) \equiv p \to (M_1 \vee \ldots \vee M_\ell)$

Daraus folgt unmittelbar $(\neg L_1 \wedge \ldots \wedge \neg L_n) \to (M_1 \vee \ldots \vee M_\ell)$
$\leadsto$ dies entspricht der Klausel $\{L_1, \ldots, L_n, M_1, \ldots, M_\ell\}$

Satz: Wenn R Resolvente der Klauseln K_1 und K_2 ist, dann gilt $\{K_1, K_2\} \models R$.

Resolutionsschritte produzieren also logische Schlüsse. Diese Eigenschaft einer Ableitungsregel wird Korrektheit genannt („jede Ableitung ist tatsächlich eine logische Konsequenz“).

Die leere Klausel

Die Resolvente kann eine leere Menge $\emptyset$ sein.

Beispiel: Die Resolvente der Klauseln $\{p\}$ und $\{\neg p\}$ ist leer.

Was bedeutet so eine leere Klausel?

- Klauseln sind Disjunktionen
- Die leere Klausel entspricht einer Disjunktion von 0 Literalen
- Dieser Ausdruck sollte das neutrale Element der Disjunktion sein, d.h. immer den Wert 0 annehmen

$\leadsto$ Wir bezeichnen die leere Klausel mit $\bot$.

Was bedeutet es, wenn $\bot$ in Klauselform auftaucht?

- Die Klauselform beschreibt eine Konjunktion von Klauseln
- Wenn eine der Klauseln $\bot$ ist, dann ist die gesamte Formel unerfüllbar

$\leadsto$ Die Ableitung von $\bot$ zeigt Unerfüllbarkeit.

Das Resolutionskalkül

Wir können damit das gesamte Verfahren angeben:

Resolution
Gegeben: Formel $\mathcal{F}$ in Klauselform
Gesucht: Ist $\mathcal{F}$ erfüllbar oder unerfüllbar?

(1) Finde ein Klauselpaar $K_1, K_2 \in \mathcal{F}$ mit Resolvente $R \notin \mathcal{F}$
(2) Setze $\mathcal{F} := \mathcal{F} \cup \{R\}$
(3) Wiederhole Schritt (1) und (2) bis keine neuen Resolventen gefunden werden können
(4) Falls $\bot \in \mathcal{F}$, dann gib „unerfüllbar“ aus; andernfalls gib „erfüllbar“ aus

Beobachtung: Unerfüllbarkeit steht fest, sobald $\bot$ abgeleitet wurde
$\leadsto$ dann kann man das Verfahren frühzeitig abbrechen

Erfüllbarkeit kann dagegen erst erkannt werden, wenn alle Resolventen erschöpfend gebildet worden sind

Beispiel

(1)	$\{\neg A, \neg B\}$		Anna oder Barbara lügen.
(2)	$\{A, B\}$		Anna oder Barbara sagen die Wahrheit.
(3)	$\{\neg B, \neg C\}$		Barbara oder Chris lügen.
(4)	$\{B, C\}$		Barbara oder Chris sagen die Wahrheit.
(5)	$\{\neg C, \neg A\}$		Chris oder Anna lügen.
(6)	$\{A, B, C\}$		Jemand sagt die Wahrheit.
(7)	$\{\neg B\}$		Barbara lügt.
(8)	$\{C\}$	(4)+(7)	Chris sagt die Wahrheit.
(9)	$\{\neg A\}$	(8)+(5)	Anna lügt.
(10)	$\{B\}$	(2)+(9)	Barbara sagt die Wahrheit.
(11)	$\bot$	(10)+(7)	Widerspruch.

$\leadsto$ Es gilt $\{\{\neg A, \neg B\}, \{A, B\}, \{\neg B, \neg C\}, \{B, C\}, \{\neg C, \neg A\}, \{A, B, C\}\} \models B$

Resolution: Eigenschaften

Resolution eignet sich tatsächlich zum logischen Schließen:

Satz: Das Resolutionskalkül hat die folgenden Eigenschaften:
- Korrektheit: Wenn $\bot$ aus $\mathcal{F}$ abgeleitet wird, dann gilt $\mathcal{F} \models \bot$.
- Vollständigkeit: Wenn $\mathcal{F} \models \bot$, dann kann $\bot$ aus $\mathcal{F}$ abgeleitet werden.
- Terminierung: Das Verfahren endet nach endlich vielen Schritten.

Beweis: Korrektheit haben wir schon für einzelne Resolutionsschritte gezeigt. Die Behauptung gilt daher auch, wenn beliebig viele Schritte nacheinander ausgeführt werden (Induktion mit Hypothese: „Die neu abgeleitete Klauselmenge folgt aus der ursprünglichen.“).

Resolution: Eigenschaften (2)

Resolution eignet sich tatsächlich zum logischen Schließen:

Satz: Das Resolutionskalkül hat die folgenden Eigenschaften:
- Korrektheit: Wenn $\bot$ aus $\mathcal{F}$ abgeleitet wird, dann gilt $\mathcal{F} \models \bot$.
- Vollständigkeit: Wenn $\mathcal{F} \models \bot$, dann kann $\bot$ aus $\mathcal{F}$ abgeleitet werden.
- Terminierung: Das Verfahren endet nach endlich vielen Schritten.

Beweis: Terminierung ist leicht zu sehen. Jede neu abgeleitete Klausel enthält nur Literale, die auch in der ursprünglichen Klauselmenge vorkamen. Die Zahl dieser Literale ist endlich; wir bezeichnen sie mit ℓ. Es gibt nur 2^ℓ Klauseln mit diesen Literalen. Man kann also weniger als 2^ℓ Resolventen hinzufügen bevor das Verfahren terminiert.

Resolution: Eigenschaften (3)

Resolution eignet sich tatsächlich zum logischen Schließen:

Satz: Das Resolutionskalkül hat die folgenden Eigenschaften:
- Korrektheit: Wenn $\bot$ aus $\mathcal{F}$ abgeleitet wird, dann gilt $\mathcal{F} \models \bot$.
- Vollständigkeit: Wenn $\mathcal{F} \models \bot$, dann kann $\bot$ aus $\mathcal{F}$ abgeleitet werden.
- Terminierung: Das Verfahren endet nach endlich vielen Schritten.

Beweis: Vollständigkeit ist etwas aufwändiger.

Vorgehen: Wir zeigen, wie man eine erfüllende Zuweisung finden kann, wenn die leere Klausel nicht abgeleitet wurde.

Widerspruchsfreie Resolution $\rightsquigarrow$ Modell

Für jedes Atom p, das in der Formel vorkommt, bestimmen wir einen Wahrheitswert $w(p)$:

Gegeben: widerspruchsfreie Klauselmenge $\mathcal{F}$ nach Resolution

Für jedes Atom p in $\mathcal{F}$ (in beliebiger Reihenfolge):

(a) Wenn $\{p\} \in \mathcal{F}$, definiere $w(p) := 1$

(b) Wenn $\{\neg p\} \in \mathcal{F}$, definiere $w(p) := 0$

(c) Wenn weder $\{p\} \in \mathcal{F}$ noch $\{\neg p\} \in \mathcal{F}$, dann setze $w(p) := 1$ und $\mathcal{F} := \mathcal{F} \cup \{\{p\}\}$ und führe nochmals Resolution durch, bis keine weiteren Klauseln abgeleitet werden.

Die so bestimmte Wertzuweisung w ist ein Modell für $\mathcal{F}$.

Beweisskizze:

(1) In Schritt (c) wird nie die leere Klausel abgeleitet (Intuition: Dazu wäre eine Klausel $\{\neg p\}$ nötig, die es aber in diesem Fall nicht geben darf und die auch nicht neu abgeleitet werden kann $\rightsquigarrow$ Induktion)

(2) Dank (1) enthält $\mathcal{F}$ auch nach Bestimmung aller Werte keine leere Klausel (einfache Induktion)

(3) Daher werden alle Klauseln in $\mathcal{F}$ erfüllt (Wenn eine nicht erfüllt wäre, dann könnte man aus ihr unter Verwendung der einelementigen Klauseln $\bot$ ableiten, was aber laut (2) nicht abgeleitet wird)

Resolution: Eigenschaften (4)

Resolution eignet sich tatsächlich zum logischen Schließen:

Satz: Das Resolutionskalkül hat die folgenden Eigenschaften:

- Korrektheit: Wenn $\bot$ aus $\mathcal{F}$ abgeleitet wird, dann gilt $\mathcal{F} \models \bot$.
- Vollständigkeit: Wenn $\mathcal{F} \models \bot$, dann kann $\bot$ aus $\mathcal{F}$ abgeleitet werden.
- Terminierung: Das Verfahren endet nach endlich vielen Schritten.

Beweis: Vollständigkeit ist etwas aufwändiger.

Vorgehen: Wir zeigen, wie man eine erfüllende Zuweisung finden kann, wenn die leere Klausel nicht abgeleitet wurde. □

Anmerkung: „Vollständigkeit" bezieht sich nur auf die Ableitung von $\bot$. Resolution kann nicht jede beliebige Klausel ableiten, die logisch folgt!

Beispiel: widerspruchsfreie Resolution

(1)	$\{\neg A, \neg B\}$		(25)	$\{A, \neg A, \neg B\}$	(16)+(3)
(2)	$\{A, B\}$		(26)	$\{\neg B, C, \neg C\}$	(17)+(1)
(3)	$\{\neg B, \neg C\}$		(27)	$\{B, \neg B, \neg C\}$	(18)+(5)
(4)	$\{B, C\}$		(28)	$\{A, \neg A, \neg C\}$	(19)+(3)
(5)	$\{\neg A, \neg C\}$		(29)	$\{\neg A, \neg B, \neg C\}$	(23)+(3)
(6)	$\{A, B, C\}$		(30)	$\{A, \neg A, B, \neg B\}$	(6)+(29)
(7)	$\{B, \neg B\}$	(2)+(1)	(31)	$\{A, \neg A, C, \neg C\}$	(6)+(29)
(8)	$\{B, \neg C\}$	(2)+(5)	(32)	$\{B, \neg B, C, \neg C\}$	(6)+(29)
(9)	$\{A, \neg A\}$	(2)+(1)	(33)	$\{A, \neg A, B, \neg B, C, \neg C\}$	(30)+(31)
(10)	$\{A, \neg C\}$	(2)+(3)	(34)	$\{\neg A, B, \neg B, C, \neg C\}$	(33)+(22)
(11)	$\{\neg A, C\}$	(4)+(1)	(35)	$\{A, \neg A, B, C, \neg C\}$	(20)+(33)
(12)	$\{C, \neg C\}$	(4)+(3)	(36)	$\{A, \neg A, B, \neg B, \neg C\}$	(33)+(21)
(13)	$\{\neg A, B\}$	(4)+(5)	(37)	$\{\neg A, B, C, \neg C\}$	(20)+(34)
(14)	$\{B, \neg B, C\}$	(6)+(1)	(38)	$\{\neg A, B, \neg B, \neg C\}$	(34)+(21)
(15)	$\{B, C, \neg C\}$	(6)+(5)	(39)	$\{A, \neg A, B, \neg C\}$	(35)+(21)
(16)	$\{A, \neg A, C\}$	(6)+(1)	(40)	$\{\neg A, B, \neg C\}$	(39)+(22)
(17)	$\{A, C, \neg C\}$	(6)+(3)	(41)	$\{A, B, \neg B, C, \neg C\}$	(2)+(33)
(18)	$\{A, B, \neg B\}$	(6)+(3)	(42)	$\{A, \neg A, \neg B, C, \neg C\}$	(33)+(3)
(19)	$\{A, \neg A, B\}$	(6)+(3)	(43)	$\{A, \neg A, B, \neg B, C\}$	(4)+(33)
(20)	$\{B\}$	(2)+(13)	(44)	... und so weiter	
(21)	$\{\neg C\}$	(8)+(3)			
(22)	$\{\neg A\}$	(13)+(1)			
(23)	$\{\neg A, B, \neg B\}$	(14)+(5)			
(24)	$\{\neg A, C, \neg C\}$	(15)+(1)			

Es ergibt sich ein Modell w mit $w(A) = 0$, $w(B) = 1$ und $w(C) = 0$.

Resolution: Komplexität

Leider können im schlimmsten Fall exponentiell viele Resolventen gebildet werden (wobei die Formel erfüllbar ist, so dass kein frühzeitiger Abbruch möglich ist)

(Die Verwendung von Hilfsatomen zur Ermittlung einer linear großen Klauselform löst dieses Problem nicht)

- Bezüglich der algorithmischen Komplexität ist Resolution schlimmstenfalls nicht besser als andere Verfahren.
- Das hier vorgestellte einfachste Verfahren ist hoffnungslos ineffizient.
- Praktische Implementierungen verwenden aber sehr viele weitere Optimierungen.

Aussagenlogisches Schließen in der Praxis

Systeme zum aussagenlogischen Schließen heißen SAT Solver.

Moderne SAT-Solver ...

... sind stark optimiert und können Probleme mit tausenden oder gar hunderttausenden von Atomen lösen

... verwenden in der Regel keine (reine) Resolution, bilden aber manchmal Resolventen

... werden auch jenseits der Aussagenlogik verwendet, indem Probleme zunächst in Aussagenlogik übersetzt werden

Resolution ist dennoch bedeutsam, da sie gut auf Prädikatenlogik und verwandte Logiken anwendbar ist, was für andere aussagenlogische Verfahren nicht zutrifft.

Horn-Logik

Horn-Logik

Bisher waren alle unsere Ansätze für aussagenlogisches Schließen exponentiell – geht es auch einfacher?

Idee: Beschränke die Form von Formeln

Eine Horn-Klausel* ist eine Klausel, die höchstens ein nicht-negiertes Literal enthält.
Eine Horn-Formel ist eine Formel in KNF, welche nur Horn-Klauseln enthält.

*) nach Alfred Horn, 1918–2001, US-amerikanischer Mathematiker

Beispiele:

- $\neg p \vee \neg q \vee r$ ist eine Horn-Klausel
- $q \vee p$ ist keine Horn-Klausel
- p ist eine Horn-Klausel
- $\neg p \vee \neg q$ ist eine Horn-Klausel
- $p \vee p$ ist keine Horn-Klausel (aber äquivalent zu einer)

Horn-Klauseln als Implikationen

Jede Horn-Klausel kann als eine Implikation ohne Negation und Disjunktion ausgedrückt werden

Beispiele:

$$\begin{array}{rcr} \neg p \vee \neg q \vee r & \equiv & (p \wedge q) \to r \\ p & \equiv & \top \to p \\ \neg p \vee \neg q & \equiv & (p \wedge q) \to \bot \end{array}$$

- Wir verwenden $\top$ für die leere Prämisse und $\bot$ für die leere Konsequenz (weil so die gewünschte logische Äquivalenz gilt)
- Als Implikationen geschriebene Horn-Klauseln werden oft als (Horn-)Regeln bezeichnet
- Es ist üblich, die Klammern der Konjunktion in der Prämisse wegzulassen

Ausblick

Nächste Vorlesung: Wir werden logisches Schließen für Horn-Regeln noch etwas genauer betrachten

- Resolution kann in diesem Fall spezialisiert werden
- Dadurch erhält man polynomielle Algorithmen

Zusammenfassung und Ausblick

Das logische Schließen umfasst viele Fragen (meist Entscheidungsprobleme), die aber oft auf Erfüllbarkeit zurückgeführt werden können

Resolution ist ein bekanntes Widerlegungskalkül, welches über die Aussagenlogik hinaus von Bedeutung ist

Die betrachteten Beweisverfahren für Aussagenlogik benötigen schlimmstenfalls exponentiell viel Zeit; bei Horn-Formeln gibt es dagegen polynomielle Schlussfolgerungsalgorithmen

Offene Fragen:

- Gibt es auch sub-exponentielle Algorithmen für aussagenlogisches Schließen?
- Was hat Aussagenlogik mit Sprachen, Berechnung und TMs zu tun?