3.4 Unentscheidbarkeit

In der Aussagenlogik gibt es für jede Formel eine endliche Menge von Belegungen.
In Prädikatenlogik Beschränkung auf endliche Menge von Strukturen nicht möglich.

Sogar Strukturen mit unendlichem Universum müssen betrachtet werden:

Beispiel: $\forall x\ P(x,f(x)) \wedge \forall y\ \neg P(y,y) \wedge \forall u\ \forall v\ \forall w\ ((P(u,v) \wedge P(v,w)) \rightarrow P(u,w))$

nur unendliche Modelle, etwa: $A = (U_A, I_A)$ mit
$U_A = \{0,1,2,...\}$
$P^A = \{(m,n) \mid m < n\}$
$f^A(n) = n + 1$

Das bedeutet, dass Entscheidungsverfahren wie in der Aussagenlogik, die einfach alle Interpretationen der Reihe nach prüfen, ausgeschlossen sind.

Ein Problem P heißt entscheidbar, wenn es ein Berechnungsverfahren gibt, das
a) für jede Eingabe terminiert,
b) für jede Eingabe die richtige JA/NEIN-Antwort liefert.

Ein Problem P heißt semi-entscheidbar, wenn es ein Berechnungsverfahren gibt, das
a) für jede Eingabe terminiert, für die die korrekte Antwort JA lautet,
b) im Falle der Terminierung die richtige Antwort liefert.

Gültigkeitsproblem der Prädikatenlogik: gegeben: Formel F gefragt: ist F gültige Formel?	Erfüllbarkeitsproblem der Prädikatenlogik: gegeben: Formel F gefragt: ist F erfüllbare Formel?

Satz: (Church) Das Gültigkeitsproblem der Prädikatenlogik ist unentscheidbar.

Bew.-skizze: durch Reduktion eines Problems, von dem man schon weiß, dass es unentscheidbar ist, auf das Gültigkeitsproblem (z.B., Korrespondenzproblem, siehe Vorlesung Berechenbarkeit, 4. Semester). Intuitiv: man übersetzt ein Problem A in ein Problem B. Wenn A in B übersetzt werden kann, dann beinhaltet eine Lösung des Entscheidungsproblems B eine Lösung von A. Es ist also mindestens so schwer, B zu entscheiden, wie A zu entscheiden. Wenn A unentscheidbar ist, dann muss es auch B sein.

Korollar: Das Erfüllbarkeitsproblem der Prädikatenlogik ist unentscheidbar.

Bew.: Wäre Erfüllbarkeit entscheidbar, so auch Gültigkeit, denn eine Formel ist genau dann gültig, wenn ihre Negation nicht erfüllbar ist.

Korollar: Folgerbarkeit in der Prädikatenlogik ist unentscheidbar.

Bew.: Wäre Folgerbarkeit entscheidbar, so auch Gültigkeit, denn eine Formel ist genau dann gültig, wenn sie aus einer Tautologie folgerbar ist.

3.5 Herbrand-Theorie

Die Definition von Strukturen lässt beliebige Mengen als Universum zu.
Algorithmische Suche nach Modellen kann auf kanonische Weise erfolgen.

Ausgangspunkt: geschlossene Formeln in Skolemform.

Def.: (Herbrand-Universum)
Das Herbrand-Universum D(F) einer geschlossenen Formel F in Skolemform ist die Menge aller variablenfreien Terme, die aus in F vorkommenden Symbolen gebildet werden können. Falls in F keine Konstante vorkommt, wird als Symbol zusätzlich eine neue Konstante a verwendet.

Induktiv:

1. Alle in F vorkommenden Konstanten sind in D(F). Falls keine Konstante vorkommt, so ist a in D(F).
2. Falls f mit Stelligkeit n in F vorkommt und $t_1, ..., t_n$ in D(F), so ist $f(t_1,...,t_n)$ in D(F).

Beispiel: $\forall x\ (\neg P(a,f(x)) \vee Q(g(b)))$
D(F) = {a, b, f(a), f(b), g(a), g(b), f(f(a)), f(f(b)), f(g(a)), f(g(b)), g(f(a)), ...}

Def.: (Herbrand-Strukturen)
Sei F geschlossene Formel in Skolemform. Eine zu F passende Struktur $A = (U_A, I_A)$ heißt Herbrand-Struktur, falls gilt:

1. $U_A = D(F)$,
2. für jedes n-stellige in F vorkommende Funktionssymbol f und $t_1, ..., t_n$ in D(F) gilt: $f^A(t_1,...,t_n) = f(t_1,...,t_n)$.

Herbrand-Strukturen lassen also nur die Wahl von P^A für jedes Prädikatensymbol P offen. Terme werden "durch sich selbst" interpretiert.

Satz: Sei F eine geschlossene Formel in Skolemform. Dann ist F erfüllbar gdw. F ein Herbrand-Modell besitzt.

Def.: (Herbrand-Expansion)
Sei $F = \forall y_1 ...\forall y_n F'$ eine geschlossene Formel in Skolemform. Die Herbrand-Expansion von F, E(F), ist wie folgt definiert:

$$E(F) = \{ F'[y_1/t_1]...[y_n/t_n] \mid t_1,...,t_n \text{ in } D(F)\}$$

Def.: (Instanz, Grundinstanz)
Sei A ein Ausdruck (Term oder Formel) mit den freien Variablen $x_1, x_2, ..., x_n$. Für Terme $t_1, t_2, ..., t_n$ heißt der Ausdruck $A[x_1/t_1][x_2/t_2]...[x_n/t_n]$, in dem jedes Auftreten von x_i durch t_i ersetzt ist,

- Instanz von A (dabei können die t_i beliebige Terme sein)
- Grundinstanz von A, wenn alle $t_1, t_2, ..., t_n$ variablenfrei sind

Die Herbrand-Expansion ist also die Menge aller Grundinstanzen von F'.

Satz: (Gödel, Herbrand, Skolem)
Für jede geschlossene Formel F in Skolemform gilt: F ist erfüllbar gdw. E(F) erfüllbar ist.

Daraus und mit dem Endlichkeitssatz der Aussagenlogik folgt:

Satz: (Herbrand)
Für jede geschlossene Formel F in Skolemform gilt: F ist unerfüllbar gdw. es eine endliche Teilmenge von E(F) gibt, die unerfüllbar ist.

Wir können nun ein Semi-Entscheidungsverfahren für die Unerfüllbarkeit von Formeln formulieren:

Algorithmus von Gilmore:

Eingabe: geschlossene Formel F in Skolemform;
Sei $E(F) = \{F_1, F_2, F_3, ...\}$
n := 0;
repeat n := n+1
until $\{F_1, ..., F_n\}$ unerfüllbar; (getestet z.B. mit Wahrheitstafeln)
stoppe mit Ausgabe "unerfüllbar";

Es handelt sich um ein Semi-Entscheidungsverfahren, denn falls F unerfüllbar, so terminiert der Algorithmus mit der korrekten die Antwort (Algorithmus terminiert nicht für erfüllbare Formeln F, also kein Entscheidungsverfahren).

Satz: a) Das Unerfüllbarkeitsproblem der Prädikatenlogik ist semi-entscheidbar.
b) Das Gültigkeitsproblem der Prädikatenlogik ist semi-entscheidbar.
c) Folgerbarkeit in der Prädikatenlogik ist semi-entscheidbar.

Bew.: a) siehe Algorithmus, b) wende Algorithmus auf $\neg F$ an, c) teste $F \wedge \neg C$, wobei F die Prämisse ist, C die Formel, die folgern soll, auf Unerfüllbarkeit.

3.6 Prädikatenlogische Resolution

Unerfüllbarkeitstests in Gilmore-Algorithmus können per aussagenlogischer Resolution durchgeführt werden:

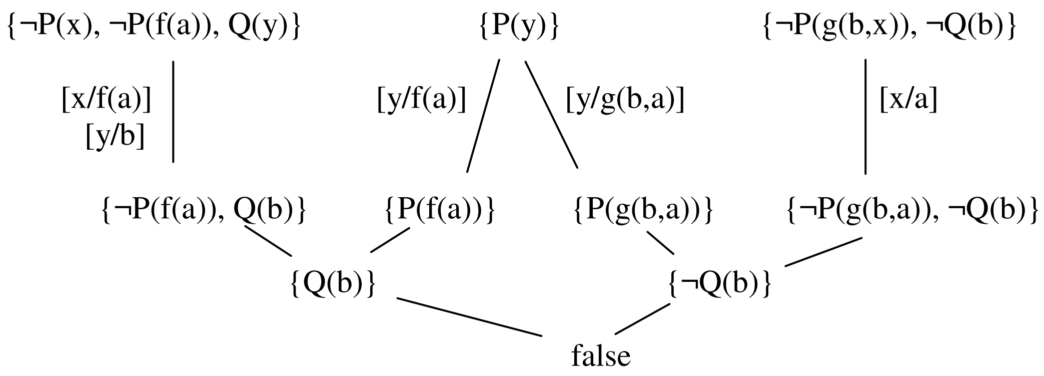
> Grundresolutionsalgorithmus:
>
> Eingabe: geschlossene Formel F in Skolemform mit Matrix F* in KNF.
> Sei $E(F) = \{F_1, F_2, F_3, ...\}$
> $i := 0$;
> $M := \{\}$;
> **repeat** $i := i+1$; $M := M \cup \{F_i\}$; $M := Res^*(M)$
> **until** leere Klausel $\in$ M;
> stoppe mit Ausgabe "unerfüllbar";

Aus Satz von Herbrand und aussagenlogischem Resolutionssatz folgt:

Satz: Bei Eingabe einer geschlossenen Formel in Skolemform mit Matrix F* in KNF stoppt der Grundresolutionsalgorithmus nach endlich vielen Schritten mit Ausgabe "unerfüllbar" gdw. F unerfüllbar ist.

Bemerkung: Algorithmus erzeugt wesentlich mehr Formeln, als für Beweis von [] nötig.

Beispiel:



Grundresolutionssatz:
Eine Aussage in Skolemform F mit Matrix F* in KNF ist unerfüllbar gdw. es eine Folge von Klauseln $K_1, ..., K_n$ gibt mit:
1) K_n leere Klausel, und
2) für alle i in $\{1,...,n\}$ gilt:
    entweder K_i ist Grundinstanz einer Klausel aus F*, oder
    K_i ist (aussagenlogische) Resolvente zweier Klauseln K_j, K_k mit $j,k < i$.

Suche nach Grundinstanzen oft ineffizient, weil es i. A. sehr viele Instanzierungsmöglichkeiten gibt. Deshalb wird in prädikatenlogischer Resolution Instanzierung so zurückhaltend wie möglich vorgenommen: nur insoweit, wie es für Resolutionsschritt erforderlich ist.

Bsp.: {P(x), ¬Q(g(x)}, {¬P(f(y))}
Es genügt die Substitution [x/f(y)] und man erhält: {¬Q(g(f(y)))}. Nicht nötig, Substitution für y zu finden.
[x/f(y)] ist Unifikator (unifier) von P(x) und P(f(y))
ebenso: [x/f(a)] [y/a], aber: durch letztere Unifikation entsteht speziellere Formel.

Def.:
Eine Substitution sub = $[x_1/t_1, ..., x_k/t_k]$ ist eine Abbildung von Variablen x_i auf Terme t_i. Sei A ein Ausdruck (Term oder Formel). A sub bezeichnet den Ausdruck, der entsteht, indem jedes Vorkommen von x_i in A durch t_i ersetzt wird.
sub heißt Unifikator einer Menge von Literalen L = $\{L_1,..., L_n\}$, falls L_1 sub = ... = L_n sub.
sub heißt allgemeinster Unifikator (most general unifier, mgu) von L falls für jeden Unifikator sub' von L gilt: sub' = sub sub'' für eine geeignete Substitution sub''.

Unifikationssatz: (Robinson)
Jede unifizierbare Menge von Literalen besitzt auch einen allgemeinsten Unifikator.

```
Unifikationsalgorithmus:

Eingabe: nicht-leere Literalmenge L
sub := []
while |L sub| > 1 do
      begin
      durchsuche Literale in L sub von links nach rechts bis zur ersten Position, an der
      sich zwei Literale unterscheiden;
      if keines der sich unterscheidenden Zeichen ist Variable then stoppe mit "fail"
      else   begin
             Sei x die Variable und t der Term, der im anderen Literal beginnt;
             if x kommt in t vor then stoppe mit "fail" (occur check)
             else sub := sub [x/t] (Hintereinanderausführung der Substitutionen)
             end;
      end;
gib mgu sub aus;
```

Beispiel: P($\underline{a}$, f(x,b)),
P($\underline{y}$, f(g(y,z)))

sub = [y/a]

P(a, f($\underline{x}$,b)),
P(a, f($\underline{g}$(a), z))

sub = [y/a] [x/g(a)]

P(a, f(g(a),$\underline{b}$)),
P(a, f(g(a),$\underline{z}$))

sub = [y/a] [x/g(a)] [z/b]

P(a, f(g(a),b)),
P(a, f(g(a),b))

in diesem Fall: [y/a] [x/g(a)] [z/b] = [y/a, x/g(a), z/b] (parallele Substitution)
im allgemeinen Fall müssen spätere Substitutionen beim Parallelmachen auf vorherige angewendet werden:

$[y/f(x,b)]\ [x/g(a)] = [y/f(g(a),b),\ x/g(a)] \neq [y/f(x,b),\ x/g(a)]$

Zum occur check: notwendig für Korrektheit (bzw Terminierung)

$P(\underline{x},x)$
$P(\underline{y}, f(y))$ $\quad$ [x/y]

$P(y,\underline{y})$
$P(y, \underline{f(y)})$ $\quad$ [x/y] [y/f(y)]

$P(f(y),f(\underline{y}))$
$P(f(y), f(\underline{f(y)}))$ [x/y] [y/f(y)] [y/f(y)] ...

Def.: (prädikatenlogische Resolution)
Seien K_1, K_2 und R Klauseln, s_1 und s_2 Substitutionen, so dass K_1s_1 und K_2s_2 keine Variablen gemeinsam haben. R heißt Resolvente von K_1 und K_2 wenn folgendes gilt:

1. Es gibt Literale $L_1,\ldots, L_m$ in K_1s_1 und $L'_1,\ldots, L'_n$ in K_2s_2, so dass $\{-L_1,\ldots, -L_m, L'_1,\ldots, L'_n\}$ durch mgu sub unifizierbar ist.
2. $R = ((K_1s_1 - \{L_1,\ldots, L_m\}) \cup (K_2s_2 - \{L'_1,\ldots, L'_n\}))$ sub.

Beispiel: $\{P(g(y)), S(h(a,y))\}$ $\qquad$ $\{Q(y), \neg S(h(y,b))\}$ $\qquad$ $s_2 = [y/z]$, sub = [z/a, y/b]

$\{P(g(b)), Q(a)\}$

Wie in der Aussagenlogik definieren wir:

$$\begin{aligned}
\text{Res}(F) &= F \cup \{R \mid R \text{ ist Resolvente zweier Klauseln in } F\}\\
\text{Res}^0(F) &= F\\
\text{Res}^{n+1}(F) &= \text{Res}(\text{Res}^n(F))\\
\text{Res}^*(F) &= \cup_{n\geq 0} \text{Res}^n(F)
\end{aligned}$$

Resolutionssatz der Prädikatenlogik: Sei F eine geschlossene Formel in Skolemform mit Matrix F* in KNF. F ist unerfüllbar gdw. $[] \in \text{Res}^*(F)$.

Resolutionsbeispiel:

Normale Vögel fliegen.
Strausse fliegen nicht.
Pinguine fliegen nicht.
Pinguine und Strausse sind Vögel
Tweety ist ein Pinguin oder ein Strauss.

Tweety ist nicht normal.

$\forall x$ (Vogel(x) $\wedge$ Normal(x) $\rightarrow$ Fliegt(x))	$\neg$Vogel(x) $\vee$ $\neg$Normal(x) $\vee$ Fliegt(x)	(1)
$\forall x$ (Strauss(x) $\rightarrow$ $\neg$Fliegt(x))	$\neg$Strauss(x) $\vee$ $\neg$Fliegt(x)	(2)
$\forall x$ (Pinguin(x) $\rightarrow$ $\neg$Fliegt(x))	$\neg$Pinguin(x) $\vee$ $\neg$Fliegt(x)	(3)
$\forall x$ (Pinguin(x) $\rightarrow$ Vogel(x))	$\neg$Pinguin(x) $\vee$ Vogel(x)	(4)
$\forall x$ (Strauss(x) $\rightarrow$ Vogel(x))	$\neg$Strauss(x) $\vee$ Vogel(x)	(5)
Strauss(Tweety) $\vee$ Pinguin(Tweety)	Strauss(Tweety) $\vee$ Pinguin(Tweety)	(6)
neg. Ziel: Normal(Tweety)	Normal(Tweety)	(7)
6,4	Strauss(Tweety) $\vee$ Vogel(Tweety)	(8)
8,5	Vogel(Tweety)	(9)
6,2	Pinguin(Tweety) $\vee$ $\neg$Fliegt(Tweety)	(10)
10,3	$\neg$Fliegt(Tweety)	(11)
9,1	$\neg$Normal(Tweety) $\vee$ Fliegt(Tweety)	(12)
11,12	$\neg$Normal(Tweety)	(13)
13,7	[]	

Effizientes Finden von Resolutionsherleitungen erschwert durch kombinatorische Explosion.
Restriktionen der Resolution: schränken Wahlfreiheit bei Resolventenbildung ein

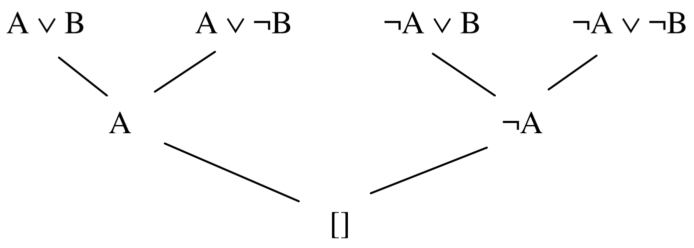
1) P-Restriktion: eine der Elternklauseln muss positiv sein

2) N-Restriktion: eine der Elternklauseln muss negativ sein.

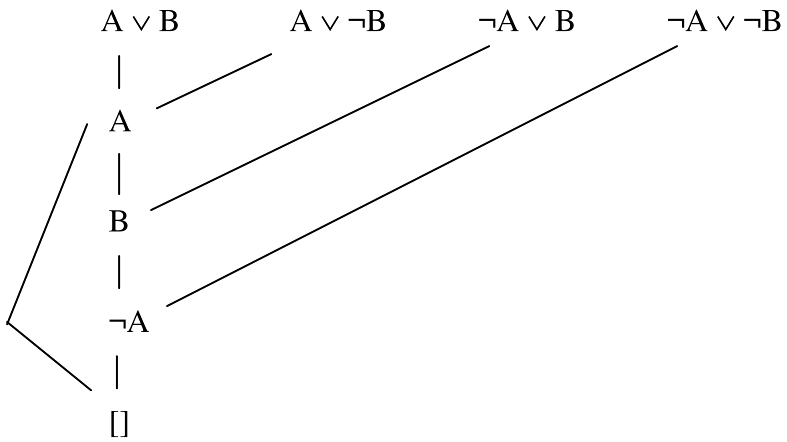
3) lineare Resolution: jeweils eine der Elternklauseln ist zuletzt erzeugte Klausel

Beispiel: $A \vee B$, $A \vee \neg B$, $\neg A \vee B$, $\neg A \vee \neg B$

Beweisbaum:



Linearisierung:



4) Stützmengenrestriktion: es sei bekannt, dass es eine Teilmenge T von F (in Klauselform) gibt, so dass F-T erfüllbar. Es dürfen nie zwei Klauseln aus F-T miteinander resolviert werden. T heißt Stützmenge.

Umso besser, je kleiner T ist. Anwendungsbeispiel: F ist Wissensbasis, deren Erfüllbarkeit gesichert ist, T negiertes Ziel der Anfrage.

5) Einheitsrestriktion (Unit resolution): mindestens eine Elternklausel ist einelementig. Kann nicht vollständig sein, siehe obiges Beispiel (nur zweielementige Klauseln in F)

6) Input-Restriktion: eine der Elternklauseln muss Element der Ausgangsklauselmenge F sein nicht vollständig. (liefert Beweise gdw es Unit-Resolution Beweise gibt).

7) SLD Resolution (linear res. with selection funct. for definite clauses): nur definiert für Hornklauseln:

Input-Resolution, wobei grundsätzlich mit einer negativen Klausel (Zielklausel) begonnen wird, die mit einer nicht-negativen (= definiten) Klausel resolviert wird. Es entstehen dabei nur negative Klauseln.

Bsp.: Meise, $\neg$ Meise $\vee$ Vogel, $\neg$Vogel $\vee$ Singt,

neg. Ziel: $\neg$Singt

SLD Resolution:

$\neg$Vogel $\vee$ Singt	$\neg$Singt
$\neg$Meise $\vee$ Vogel	$\neg$Vogel
Meise	$\neg$Meise
	[]

Satz:

a) Resolution unter P-Restriktion ist vollständig.
b) Resolution unter N-Restriktion ist vollständig.
c) Resolution unter linearer Resolutionsrestriktion ist vollständig.
d) Resolution unter Stützmengen-Restriktion ist vollständig.
e) Input-Resolution ist vollständig für Hornklauseln.
f) Einheitsresolution ist vollständig für Hornklauseln.
g) SLD-Resolution ist vollständig für Hornklauseln.

Bemerkung 1: „vollständig“ heißt hier immer: widerlegungsvollständig (siehe Diskussion in Abschn. Aussagenlogik).

Bemerkung 2: die Kombination zweier vollständiger Restriktionen ist im Allgemeinen nicht mehr vollständig: Beispiel N- und P Restriktion, F wie eben: $A \vee B$, $A \vee \neg B$, $\neg A \vee B$, $\neg A \vee \neg B$. Es ist nur Resolution mit $A \vee B$ und $\neg A \vee \neg B$ möglich, das ergibt $A \vee \neg A$, $B \vee \neg B$. keine weiteren Resolventen.

dagegen P-Resolution: siehe oben