Aussagenlogik – Syntax

- atomare Formel: A_i, $i = 1, 2, 3, \ldots$.
- Formeln:

1. Alle atomaren Formeln sind Formeln.
2. $(F \wedge G)$ und $(F \vee G)$ sind Formeln, falls F und G Formeln sind.
3. $\neg F$ ist eine Formel, falls F eine Formel ist.

Aussagenlogik – Semantik

- **D**: Teilmenge der atomaren Formeln.
- Belegung: $\mathcal{A} : \mathbf{D} \to \{0, 1\}$.
- **E**: Menge aller Formeln, die nur aus den atomaren Formeln in **D** aufgebaut sind.

- Formelauswertung: $\overline{\mathcal{A}} : \mathbf{E} \to \{0, 1\}$.

1. $\overline{\mathcal{A}}(A_i) = \mathcal{A}(A_i)$.

2. $\overline{\mathcal{A}}(F \wedge G) = \begin{cases} 1, & \overline{\mathcal{A}}(F) = 1 \text{ und } \overline{\mathcal{A}}(G) = 1 \\ 0, & \text{sonst} \end{cases}$

3. $\overline{\mathcal{A}}(F \vee G) = \begin{cases} 1, & \overline{\mathcal{A}}(F) = 1 \text{ oder } \overline{\mathcal{A}}(G) = 1 \\ 0, & \text{sonst} \end{cases}$

4. $\overline{\mathcal{A}}(\neg F) = \begin{cases} 1, & \overline{\mathcal{A}}(F) = 0 \\ 0, & \text{sonst} \end{cases}$

Aussagenlogik – Formeleigenschaften

- F heißt *erfüllbar*, wenn es eine Belegung $\mathcal{A}$ gibt mit $\overline{\mathcal{A}}(F) = 1$.

- Formeln F und G heißen *äquivalent*, falls für alle Belegungen $\mathcal{A}$ gilt: $\overline{\mathcal{A}}(F) = \overline{\mathcal{A}}(G)$.

- F heißt Folgerung von G, wenn aus $\overline{\mathcal{A}}(G) = 1$ auch $\overline{\mathcal{A}}(F) = 1$ folgt.

<u>Alternativ:</u> $G \wedge \neg F$ ist nicht erfüllbar.

Aussagenlogik – Normalformen

- Literal: atomare Formel oder Negation einer atomaren Formel.

- F ist in *konjunktiver Normalform* (**KNF**), falls

$$F = \left(\bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} L_{i,j} \right) \right),$$

$L_{i,j} \in \{A_1, A_2, \ldots\} \cup \{\neg A_1, \neg A_2, \ldots\}$.

- Theorem: Zu jeder Formel F gibt es eine äquivalente in **KNF**.

Aussagenlogik – Klauseldarstellung

- Literal: atomare Formel oder Negation einer atomaren Formel.
- Klausel: Menge von Literalen.
- F in **KNF**, d.h.

$$F = (L_{1,1} \vee \ldots \vee L_{1,m_1}) \wedge \ldots \wedge (L_{n,1} \vee \ldots \vee L_{n,m_n}),$$

$L_{i,j} \in \{A_1, A_2, \ldots\} \cup \{\neg A_1, \neg A_2, \ldots\}$,
kann als Menge von Klauseln dargestellt werden:

$$F = \{\{L_{1,1}, \ldots, L_{1,m_1}\}, \ldots, \{L_{n,1}, \ldots, L_{n,m_n}\}\}$$

Aussagenlogik – Resolution I

R, K_1, K_2 : Klauseln.

R heißt *Resolvente* von K_1 und K_2, falls

- es gibt Literal L mit $L \in K_1, \overline{L} \in K_2$,
- $R = (K_1 - L) \cup (K_2 - \overline{L})$,

wobei

$$\overline{L} = \begin{cases} \neg A_i, & L = A_i \\ A_i, & L = \neg A_i. \end{cases}$$

Aussagenlogik – Resolution II

Resolutionslemma:

- F: Formel in **KNF** als Klauselmenge
- R: Resolvent zweier Klauseln K_1, K_2 von F.

Dann sind F und $F \cup \{R\}$ äquivalent.

- $Res(F) := \{R \mid R$ ist Resolvent zweier Klauseln von $F\}$
- $Res^0(F) := F$.
- $Res^{n+1}(F) := Res(Res^n(F))$.
- $Res^*(F) := \bigcup_{n \geq 0} Res^n(F)$.

Lemma: Zu jeder endl. Klauselmenge F gibt es ein k mit $Res^k(F) = Res^{k+1}(F) = Res^*(F)$.

Aussagenlogik – Resolution III

Resolutionssatz: F ist unerfüllbar gdw. $\emptyset \in Res^*(F)$.

Resolutionsalgorithmus: Eingabe: Formel F als Klauselmenge

```
1) n:=0; R[n]:=F;

2) repeat
     n:=n+1;
     R[n]:=Res(R[n-1]);
   until (empty in R[n]) or (R[n]=R[n-1])

3) if empty in R[n] then ‘‘F ist unerfuellbar’’
   else ‘‘F ist erfuellbar’’.
```

Prädikatenlogik – Syntax

- Variablen x_i, $i = 1, 2, 3, \ldots$
- Funktionssymbole f_i^k, $i = 1, 2, 3, \ldots$; k : Stelligkeit
- Prädikatssymbole P_I^k, $i = 1, 2, 3, \ldots$; k : Stelligkeit

- Terme:

1. Jede Variable ist ein Term.

2. $f(t_1, \ldots, t_k)$ ist ein Term, falls f $k-$stelliges Funktionssymbol und $t_1, \ldots, t_k$ Terme sind.

- Formeln:

1. $P(t_1, \ldots, t_k)$ ist eine Formel, falls P $k-$stelliges Prädikatssymbol und $t_1, \ldots, t_k$ Terme sind.

2. $(F \wedge G)$ und $(F \vee G)$ sind Formeln, falls F und G Formeln sind.

3. $\neg F$ ist eine Formel, falls F eine Formel ist.

4. $\exists x F$ und $\forall x F$ sind Formeln, falls F eine Formel und x eine Variable sind.

Prädikatenlogik – Normalformen

Lemma: Zu F gibt es äquivalentes G in bereinigter Form, d.h.

- keine Variable kommt sowohl gebunden als auch frei vor, und
- hinter allen Quantoren stehen verschiedene Variablen.

Lemma: (Existenzabschluß) F Formel mit freien Variablen $x_1, \ldots, x_n$.
F ist erfüllbar gdw. $\exists x_1 \exists x_2 \ldots \exists x_n F$ erfüllbar.

Lemma: Zu F gibt es äquivalentes G in Pränex-Form, d.h.

$$G = Q_1 y_1 Q_2 y_2 \ldots Q_n y_n F',$$

wobei $Q_i \in \{\forall, \exists\}$, in F' kommt kein Quantor vor.

Prädikatenlogik – Skolemform

Skolemform: Eingabe: Formeln F in **BPF**
(bereinigt und in Pränexnormalform)

while (F enthaelt Existenzquantor) **do**
begin
  sei $F = \forall y_1 \forall y_2 \ldots \forall y_n \exists z G,\ n \geq 0$
  sei f ein neues, n-stelliges Funktionssymbol
  $F := \forall y_1 \forall y_2 \ldots \forall y_n G[z/f(y_1, \ldots, y_n)]$
end

Lemma: F in **BPF** erfüllbar gdw. Skolemform von F erfüllbar.

Prädikatenlogik – Zusammenfassung Normalisierung

Eingabe: prädikatenlogische Formel F

Ausgabe: erfüllbarkeitsäquivalente Klauselmenge

1.) Bereinigen liefert F_1 äquivalent zu F.
2.) Existenzabschluß liefert F_2, erfüllbarkeitsäquivalent zu F_1.
3.) F_3 ist Pränexform (äquivalent) zu F_2.
4.) Skolemisierung liefert F_4, erfüllbarkeitsäquivalent zu F_3.
5.) Schreibe den quantorenfreien Teil von F_4 als Klauselmenge.

Herbrand-Expansion

$F = \forall y_1 \forall y_2 \ldots \forall y_n F^*$ in Skolemform.

Herbrand-Universum $D(F)$:

1.) Alle in F vorkommenden Konstanten sind in $D(F)$. Falls F keine Konstanten hat, ist $a \in D(F)$.
2.) $f(t_1, \ldots, t_n)$ ist in $D(F)$, falls $t_1, \ldots, t_n$ in $D(F)$ und f n-stelliges Funktionssymbol aus F.

Herbrand-Expansion:

$$E(F) := \{F^*[y_1/t_1] \ldots [y_n/t_n] \,|\, t_1, \ldots t_n \in D(F)\}.$$

Semi-Entscheidungsverfahren für Unerfüllbarkeit

<u>Satz von Herbrand:</u> F in Skolemform unerfüllbar gdw. endl. Teilmenge von $E(F)$ unerfüllbar.

<u>Algorithmus von Gilmore</u> Eingabe: F in Skolemform.
Sei $E(F) = \{F_1, F_2, F_3, \ldots\}$.

n:=0;
repeat
    n:=n+1
until $(F_1 \wedge F_2 \wedge \ldots \wedge F_n)$ unerfüllbar
gib „unerfüllbar“ aus und stoppe.

Grundresolutionsalgorithmus PL

Eingabe: F als Klauselmenge

$i := 0;$

$M := \emptyset$

repeat

$\quad i := i + 1;$

$\quad M := M \cup \{F_i\};$

$\quad M := Res^*(M);$

until $(\emptyset \in M)$

gib „unerfüllbar“ aus und stoppe.

Theorem: Alg. stoppt nach endl. vielen Schritten gdw. F unerfüllbar ist.

Grundresolutionsatz

$F = \forall y_1 \forall y_2 \ldots \forall y_n F^*$ unerfüllbar gdw.
es gibt Folge von Klauseln $K_1, \ldots, K_n$ mit:

- $K_n = \emptyset$ und
- $K_i = K[y_1/t_1] \ldots [y_n/t_n]$ mit $K \in F^*$, $t_1, \ldots t_n \in D(F)$ oder
- K_i ist Resolvent von K_a und K_b mit $a, b < i$.

Unifikation

Unifikator für eine Menge von Literalen $\mathbf{L} = \{L_1, \ldots, L_k\}$ ist eine *Substitution* sub mit

$$L_1 \ sub = L_2 \ sub = \ldots = L_k \ sub.$$

allgemeinster Unifikator: Unifikator sub eine Menge $\mathbf{L}$, so daß für alle Unifkatoren sub' von $\mathbf{L}$ gilt: $\exists s : sub\, s = sub'$.

Prädikatenlogik – Resolution I

R, K_1, K_2 : prädikatenlogische Klauseln.

R heißt *prädikatenlogische Resolvente* von K_1 und K_2, falls

- $K_1\ s_1$ und $K_2\ s_2$ enthalten keine gemeinsamen Variablen (s_1, s_2 Substitutionen)

- es gibt $\{L_1, \ldots, L_m\} \in K_1$, und $\{L'_1, \ldots, L'_n\} \in K_2$ so daß: sub ist allgem. Unif. von $\{\overline{L_1}, \ldots, \overline{L_m}, L'_1, \ldots, L'_n\}$.

- $R = ((K_1\ s_1 - \{L_1, \ldots, L_m\}) \cup (K_2\ s_2 - \{L'_1, \ldots, L'_n\}))sub.$

Prädikatenlogik – Resolution II

F: Klauselmenge

- $Res(F) := \{R \mid R$ ist prädikatenlog. Resolvent zweier Klauseln von $F\}$
- $Res^0(F) := F$.
- $Res^{n+1}(F) := Res(Res^n(F))$.
- $Res^*(F) := \bigcup_{n \geq 0} Res^n(F)$.

<u>Theorem:</u> F ist unerfüllbar gdw. $\emptyset \in Res^*(F)$.

Hornklauseln – SLD-Resolution

Hornformel: F ist Hornformel, falls F in **KNF** ist und jedes Disjunktionsglied in F höchstens ein positives Literal enthält.

negative Klausel/Zielklausel: Klausel ohne positives Literal.

definite Klausel/Programmklausel: Klausel mit einem positiven Literal.

SLD-Resolution:

- Resolution startet mit einer Zielklausel.
- in jedem Resolutionsschritt ist eine Elternklausel eine Programmklausel.

Warum Hornklauseln?

- Die „meisten“ mathematischen Theorien scheinen per Hornklauseln axiomatisierbar zu sein.

- Antwortgenerierung ist bei Hornklauseln klar.

- SLD-Resolution ist „vollständig“ für Hornklauseln, ist „effizientes“ Verfahren.

Logikprogramm

Tatsachenklauseln: positive Atomformel.

Prozedurklauseln: $\{P, \neg Q_1, \ldots, \neg Q_n\}$, in Prolog:

$$P : -Q_1, \ldots, Q_n$$

Logikprogramm: endl. Menge von Tatsachen- und Prozedurklauseln.

Anfrage/Zielklausel: $\{\neg Q_1, \ldots, \neg Q_n\}$, in Prolog:

$$: -Q_1, \ldots, Q_n$$