Aussagenlogik

Literale

Definition

Ein Literal ist eine Aussagenvariable oder die Negation einer Aussagenvariablen.

Literale

Prolog-Programm p03.pl (Anfang)

```
:- op(550,fx,p).        %Aussagenvariable z.B.  p 3
:- op(600,fy,nicht).
:- op(620,xfy,und).     %oder :- op(630,xfx,und).
:- op(630,xfy,oder).    %oder :- op(630,xfx,oder).
:- op(800,xf,ist_Aussagenvariable).
:- op(800,xf,ist_Literal).

p _ ist_Aussagenvariable.

P ist_Literal:-
   P ist_Aussagenvariable.
nicht P  ist_Literal:-
   P ist_Aussagenvariable.
```

Literale

Beispiele

```
?- [p03].
% p03 compiled 0.00 sec, 2,824 bytes

Yes
?- p 5 ist_Literal.
```

P_5

```
Yes
?- nicht p 4 ist_Literal.
```

$\neg\ P_4$

```
Yes
?- (p 2 und nicht p 4) oder p 1 ist_Literal.
```

$(P_2 \wedge \neg\ P_4) \vee\ P_1$

```
No
```

Negationsnormalform

Definition

Eine Formel ist in Negationsnormalform (NNF), wenn sie das Negationszeichen $\neg$ nur unmittelbar vor Aussagenvariablen enthält.

Definition (äquivalente induktive Definition für NNF)

- Jedes Literal ist in NNF
- Wenn F und G in NNF sind, dann auch $(F \wedge G)$ und $(F \vee G)$.

Negationsnormalform

Prolog-Programm `p03.pl` (Fortsetzung)

```
:- op(800,xf,ist_in_NNF).

L ist_in_NNF:-
   L ist_Literal.
F und G  ist_in_NNF:-
   F ist_in_NNF,
   G ist_in_NNF.
F oder G  ist_in_NNF:-
   F ist_in_NNF,
   G ist_in_NNF.
```

Negationsnormalform

Beispiele

- $P_1 \land \neg P_2$ ist in NNF.
- $\neg\neg P_1$ ist nicht in NNF.
- $\neg(P_1 \land P_2)$ ist nicht in NNF.

Hierzu `?- [p03].` aufrufen und die entsprechenden Anfragen eingeben!

Verwandlung in Negationsnormalform

Satz

Zu jeder Formel F gibt es eine semantisch äquivalente Formel in NNF.

Verwandlung in Negationsnormalform

Prolog-Programm p03.pl (Fortsetzung)

```
:- op(800,xfx,hat_die_NNF).

L hat_die_NNF L:-
   L ist_Literal.
nicht nicht F  hat_die_NNF G:-
   F hat_die_NNF G.
```

Verwandlung in Negationsnormalform

Prolog-Programm p03.pl (Fortsetzung)

```
nicht (F und G)  hat_die_NNF  H oder I:-
   nicht F  hat_die_NNF  H,
   nicht G  hat_die_NNF  I.
nicht (F oder G)  hat_die_NNF  H und I:-
   nicht F  hat_die_NNF  H,
   nicht G  hat_die_NNF  I.
F und G  hat_die_NNF  H und I:-
   F hat_die_NNF H,
   G hat_die_NNF I.
F oder G  hat_die_NNF  H oder I:-
   F hat_die_NNF H,
   G hat_die_NNF I.
```

Konjunktion und Disjunktion

Definition

Die Konjunktion $F_1 \wedge \cdots \wedge F_n$ von n Formeln $F_1, \ldots, F_n$ ist durch Induktion nach n folgendermaßen definiert:

- Die Konjunktion von F_1 ist F_1.
- $F_1 \wedge \cdots \wedge F_{n+1} := F_1 \wedge (F_2 \wedge \cdots \wedge F_{n+1})$.

Definition

Die Disjunktion $F_1 \vee \cdots \vee F_n$ von n Formeln $F_1, \ldots, F_n$ ist durch Induktion nach n folgendermaßen definiert:

- Die Disjunktion von F_1 ist F_1.
- $F_1 \vee \cdots \vee F_{n+1} := F_1 \vee (F_2 \vee \cdots \vee F_{n+1})$.

Disjunktive Normalform

Definition

Eine Formel ist in disjunktiver Normalform (DNF), wenn sie Disjunktion von Konjunktionen von Literalen ist.

Beispiele

- P ist in DNF.
- $\neg P$ ist in DNF.
- $(\neg P \wedge (Q \wedge P))$ ist in DNF. Abkürzung: $\neg P \wedge Q \wedge P$
- $((P \wedge Q) \wedge R)$ ist nicht in DNF.
- $(P \vee (\neg Q \vee \neg R))$ ist in DNF. Abkürzung: $P \vee \neg Q \vee \neg R$.
- $(P \vee ((\neg P \wedge (Q \wedge R)) \vee ((R \wedge Q) \vee \neg Q)))$ ist in DNF. Abkürzung: $P \vee (\neg P \wedge Q \wedge R) \vee (R \wedge Q) \vee \neg Q$.

Disjunktive Normalform

Prolog-Programm p03.pl (Fortsetzung)

```
:- op(800,xf,ist_Konjunktion_von_Literalen).
:- op(800,xf,ist_in_DNF).

L ist_Konjunktion_von_Literalen:-
   L ist_Literal.
L und F  ist_Konjunktion_von_Literalen:-
   L ist_Literal,
   F ist_Konjunktion_von_Literalen.

F ist_in_DNF:-
   F ist_Konjunktion_von_Literalen.
F oder G  ist_in_DNF:-
   F ist_Konjunktion_von_Literalen,
   G ist_in_DNF.
```

Konjunktive Normalform

Definition

Eine Formel ist in konjunktiver Normalform (KNF), wenn sie Konjunktion von Disjunktionen von Literalen ist.

Beispiele

- P ist in KNF.
- $\neg P$ ist in KNF.
- $(\neg P \vee (Q \vee P))$ ist in KNF. Abkürzung: $\neg P \vee Q \vee P$
- $((P \vee Q) \vee R)$ ist nicht in KNF.
- $(P \wedge (\neg Q \wedge \neg R))$ ist in KNF. Abkürzung: $P \wedge \neg Q \wedge \neg R$.
- $(P \wedge ((\neg P \vee (Q \vee R)) \wedge ((R \vee Q) \wedge \neg Q)))$ ist in KNF. Abkürzung: $P \wedge (\neg P \vee Q \vee R) \wedge (R \vee Q) \wedge \neg Q$.

Konjunktive Normalform

Prolog-Programm p03.pl (Fortsetzung)

```
:- op(800,xf,ist_Disjunktion_von_Literalen).
:- op(800,xf,ist_in_KNF).

L ist_Disjunktion_von_Literalen:-
   L ist_Literal.
L oder F  ist_Disjunktion_von_Literalen:-
   L ist_Literal,
   F ist_Disjunktion_von_Literalen.

F ist_in_KNF:-
   F ist_Disjunktion_von_Literalen.
F und G  ist_in_KNF:-
   F ist_Disjunktion_von_Literalen,
   G ist_in_KNF.
```

Verwandlung in disjunktive Normalform

Satz

Zu jeder Formel gibt es eine semantisch äquivalente Formel in DNF.

Verwandlung in disjunktive Normalform

Beweisidee

Beweise nacheinander:

1. Wenn F und G zwei Formeln in DNF sind, dann kann man zu $F \vee G$ eine semantisch äquivalente Formel in DNF konstruieren durch Umklammerung (Anwendung des Assoziativgesetzes).
2. Wenn F und G zwei Formeln in DNF sind, dann kann man zu $F \wedge G$ eine semantisch äquivalente Formel in DNF konstruieren durch Ausmultiplizieren und Umklammern (Anwendung des Distributivgesetzes und des Assoziativgesetzes).
3. Zeige, durch Induktion nach der Länge von F, dass man zu jeder Formel F in NNF eine semantisch äquivalente Formel in DNF konstruieren kann.
4. Kombiniere dies mit dem Satz über die Verwandlung in NNF.

Verwandlung in disjunktive Normalform

Aufgabe 6

Führen Sie den Beweis des letzten Satzes aus!

Aufgabe 7

1. Schreiben Sie, beruhend auf der Beweisidee des Satzes, ein Prolog-Programm zur Verwandlung einer Formel in DNF.
2. Analysieren Sie Ihr Programm im Hinblick auf seine Zeitkomplexität!

Verwandlung in disjunktive Normalform

Alternative Beweisidee für den Satz

1. Schränke die Logiksprache auf diejenigen Aussagenvariablen ein, die in der gegebenen Formel F vorkommen.
2. Dann hat F nur endlich viele Modelle.
3. Für jedes Modell I von F betrachte die Konjunktion derjenigen Literale, die bei I wahr sind.
4. Die Disjunktion aller so erhaltenen Konjunktionen stellt eine DNF von F dar (ausgenommen, wenn F unerfüllbar ist).

Aufgabe 8

1. Schreiben Sie, beruhend auf dieser alternativen Beweisidee, ein Prolog-Programm zur Verwandlung einer Formel in DNF.
2. Analysieren Sie Ihr Programm im Hinblick auf seine Zeitkomplexität!

Verwandlung in konjunktive Normalform

Satz

Zu jeder Formel gibt es eine semantisch äquivalente Formel in KNF.

Beweisideen

Die zu den vorigen Beweisen dualen Beweise sind Beweise dieses Satzes.

Aufgabe 9

Formulieren Sie den zum zweiten (alternativen) Beweis dualen Beweis!

Testen einer Formel in KNF auf Allgemeingültigkeit

Aufgabe 10

1. Entwerfen Sie einen effizienten Algorithmus, um von einer Formel in KNF festzustellen, ob sie allgemeingültig ist!
2. Implementieren Sie ihn durch ein Prolog-Programm, das zumindest nur polynomiale Zeitkomplexität hat!

Länge der KNF

Aufgabe 11

- Für eine Formel F ist die Länge $|F|$ von F definiert als die Länge des Wortes F, wobei jede Aussagenvariable, jeder der Junktoren $\neg$, $\wedge$ und $\vee$ und jede der Klammern (und) jeweils als ein Zeichen zählt.
- Unter der *KNF-Komplexität* $k_{\mathrm{KNF}}(F)$ einer Formel F wollen wir die Länge der kürzesten Formel in KNF, die semantisch äquivalent zu F ist, verstehen:

$$k_{\mathrm{KNF}}(F) := \min\{|G| \mid F \sim G \text{ und } G \text{ ist in KNF}\}.$$

- Die Funktion $f\colon \mathbb{N} \to \mathbb{N}$ sei folgendermaßen definiert.

$$f(n) := \max\{k_{\mathrm{KNF}}(F) \mid |F| \leq n\}.$$

- Wie schnell wächst f an?

Automatisches Beweisen und Normalform

Zu beweisende Formel hat oft die Form:
Aus Voraussetzungen folgt Behauptung:

$$A_1 \wedge \cdots \wedge A_n \to B$$

Dies ist semantisch äquivalent zu

$$\neg A_1 \vee \cdots \vee \neg A_n \vee B$$

und oft nahe an einer DNF.

Automatisches Beweisen und Normalform

Formel F in DNF

$$(L_{11} \wedge \cdots \wedge L_{1n_1}) \vee \cdots \vee (L_{m1} \wedge \cdots \wedge L_{mn_m})$$

allgemeingültig genau dann, wenn duale Formel

$$(L_{11} \vee \cdots \vee L_{1n_1}) \wedge \cdots \wedge (L_{m1} \vee \cdots \vee L_{mn_m})$$

(in KNF) unerfüllbar. Auch genau dann, wenn $\neg F$ unerfüllbar. Formel $\neg F$ ist ebenfalls mit deMorgan und dem Gesetz der doppelten Negation leicht in KNF verwandelbar.

Automatisches Beweisen und Normalform

Folgerung

Das Problem, von einer Formel in DNF die Algemeingültigkeit zu beweisen, ist äquivalent zum Problem, von einer Formel in KNF die Unerfüllbarkeit zu beweisen.

Zwei Verfahren des automatischen Beweisens

Konnektionsmethode

zum Nachweis der Allgemeingültikeit
(bei der einfachsten Formulierung: für Formeln in DNF)

Resolution

zum Nachweis der Unerfüllbarkeit
für Formeln in KNF