Der Klauselkalkül

Definitionen

Entsprechend der Definition eines Logikkalküls aus dem Skript S.67 müssen wir **Alphabet X**, **Sprache A, Satzmenge S** und **Deduktionsoperator F** definieren.

> **Klauseln sind Alternativen von Literalen und die immer falsche Klausel**

Ich bezeichne die in der Vorlesung definierte Menge von Variablen mit **AV** und nehme daher in mein Alphabet **X** die Zeichen **p** und * auf.

Die Sprache der Klauseln **A** definiere ich induktiv unter Verwendung des Attributes **vKommtVorInC,** welches wie in der Übung induktiv definiert ist, und für eine Zeichenkette **v** und eine Zeichenkette **C** wahr ist, wenn **v** in **C** als Teilzeichenkette vorkommt und sonst falsch ist. Das Zeichen ⊥ wird ebenfalls in **X** aufgenommen und wird als die immer falsche Klausel interpretiert. Als letzte Zeichen, nehmen wir noch ¬ und ∨ in **X** auf. Damit definieren wir:

> **X := { p, *, ⊥, ¬, ∨ }**

> **A ist die kleinste Menge von Zeichenketten über X,**
> **die folgendes erfüllt :**

> **IA. ⊥ ∈ A und für jedes v∈AV sind v ∈ A und ¬ v ∈ A**
> **IS. Ist C ∈ A \ { ⊥ } und v ∈ AV,**
> **dann sind auch C ∨ v und C ∨ ¬ v Klauseln aus A,**
> **falls in den neu eingeführten Klauseln v**
> **bzw. ¬ v NICHT doppelt vorkommt.**

Damit haben wir genau die Menge von Ausdrücken für Klauseln erfasst wie in der Vorlesung. Aufgrund der nachfolgenden Definition könnten wir auch auf den Ausschluß doppelter Literale verzichten, da sie sowieso in eine Klasse fallen. Ich führe folgende induktiv definierte Mengennotation für Klauseln ein:

> **IA. ⊥ ≅ {} (= ∅) und p ≅ {p} und ¬ p ≅ {¬ p}**
> **sind die Darstellungen der Klauseln aus dem Anfangsschritt.**

> **IS. Ist C eine Klausel in Mengennotation und v ∈ AV dann sind C ⋃ {v} und C ⋃ {¬ v}**
> **ebenfalls Klauseln in Mengennotation.**

Bemerkung: In Wirklichkeit ist dies nicht nur eine neue Notation, sondern auch noch eine Klassenbildung, da Klauseln, welche sich nur in der Reihenfolge der auftretenden Literale (**v**'s und **¬v**'s) unterscheiden, in dieser Notation identifiziert werden.

Nun definiere ich **cut** als ternäre Relation über Tripeln von Klauseln in Mengennotation.

Es gilt : cut[C_1, C_2, C_3] gdw. es gibt ein v∈AV mit v∈C_1 und ¬ v∈C_2 und C_3 = (C_1 \ {v}) ⋃ (C_2 \ {¬ v})

Zur Definition der Satzmenge **S** bemerken wir, daß genau die Klauseln, die für mindestens ein **v** auch **¬v** enthalten, allgemeingültig im Sinne des AK sind. Die Umwandlung einer Klausel in einen Ausdruck **H ∈ ausd** ist naheliegend (**,** wird ∨ und **{** fällt weg) und wird hier nicht weiter ausgeführt.

Der Deduktionsoperator **F**, den wir hier **CUT** nennen, ist nun induktiv wie folgt definiert:

Sei Y eine Menge von Klauseln,
dann ist CUT[Y] folgendermaßen definiert

IA. Falls C ∈ Y, dann ist auch C ∈ CUT[Y].

IS. Sind C_1 und C_2 aus CUT[Y] und gilt cut[C_1, C_2, C_3],
dann ist auch C_3 ∈ CUT[Y].

Es bleiben die Hülleneigenschaften, der Endlichkeitssatz und die Invarianz der Satzmenge zu zeigen.

Beweise

Satz der Einbettung: X ⊆ CUT(X)

Sei **H ∈ X**, dann folgt die Beh. sofort mit dem IA. der Def. von **CUT(X).**

Der Satz der Monotonie: Wenn X ⊆ Y, so CUT(X) ⊆ CUT(Y).

Sei **H ∈ CUT(X).** Wir zeigen **H ∈ CUT(Y).**

Was könnte der Grund für **H ∈ CUT(X)** sein?

1. Fall: H ∈ X (nach Induktionsanfang der Definition von **CUT(X)**). Dann also auch **H ∈ Y (**wegen **X ⊆ Y)** und somit **H ∈ CUT(Y)** (nach IS der Definition von **CUT(Y)).**

2. Fall: Es gibt Klauseln C_1 und C_2 aus **CUT(X)** und **cut(** C_1, C_2, **H**). Für C_1 und C_2 können wir nun die Induktionsvoraussetzung in Anwendung bringen, d.h. C_1 und C_2 sind auch aus **CUT(Y).** Dann ist aber mit **cut(** C_1, C_2, **H**) auch **H ∈ CUT(Y) (**wieder nach Def. von **CUT(Y).** w.z.b.w.

Der Satz der Abgeschlossenheit: CUT(CUT(Y)) ⊆ CUT(Y).

Wir zeigen induktiv über die Erzeugung von CUT, d.h. unter Verwendung der cut-Ableitbarkeitsdefinition:

Wenn **H** ∈ CUT(CUT(**Y**)), dann gilt auch schon **H** ∈ CUT(**Y).**

Sei **H** ∈ CUT(CUT(**Y**).)Dies ist induktiv definiert und bedeutet entweder **H**∈CUT(**Y**) (nach Induktionsanfang der Cut-Ableitbarkeitsdefinition), dann sind wir fertig mit dem Induktionsanfang des Beweises, oder nach dem Induktionsschritt der Cut-Ableitbarkeitsdefinition existiert ein Paar Klauseln C_1, C_2 ∈ CUT(CUT(**Y**)) und cut(C_1, C_2, **H**). Nach Induktionsvoraussetzung unseres Beweises können wir annehmen, daß schon gilt C_1, C_2 ∈ CUT(**Y**). Wenn wir daraus nun zeigen können, daß auch **H**∈

CUT(**Y**) ist, sind wir fertig. Aber mit $C_1, C_2 \in$ CUT(**Y**) und cut(C_1, C_2, **H**) gilt nach dem Induktionsschritt der Cut-Ableitbarkeitsdefinition **H** ∈ CUT(**Y**). **w.z.b.w.**

Wir zeigen den **Endlichkeitssatz** ebenfalls induktiv über die Erzeugung von **CUT**:

Sei **H** ∈ CUT(**Y**). Das heißt entsprechend der induktiven Definition entweder

Fall 1: H ∈ Y (Induktionsanfang),

oder

Fall 2: Es existieren zwei Klauseln C_1, C_2 mit $C_1, C_2 \in$ CUT(**Y**) und cut(C_1, C_2, **H**) (Induktionsschritt).

zu Fall 1: Hier folgt **H** ∈ CUT({**H**}), d.h. im Schnittregelabschluß der endlichen Teilmenge $Y^* = \{H\}$ von **Y.** Fertig für diesen Fall.

zu Fall 2: Hier können wir als Induktionsvoraussetzung annehmen, daß der Endlichkeitssatz für C_i schon gilt, d.h.es existieren Teilmengen Y_1^* und Y_2^* von **Y**, beide endlich, und $C_i \in \mathrm{CUT}\ (Y_i^*)$. Wir definieren $Y^* = Y_1^* \bigcup Y_2^*$. Wegen der Monotonie für CUT, (die wir schon gezeigt haben), gilt auch $C_i \in \mathrm{CUT}\ (Y^*)$. Es folgt mit cut($C_1, C_2$, **H**) dann auch $H \in \mathrm{CUT}\ (Y^*)$. Da Y^* als Vereinigung endlicher Mengen endlich ist und auch eine Teilmenge von **Y** ist, ist der Beweis von Fall 2 fertig. Fall 1 und Fall 2 zusammen ergeben den kompletten Beweis des Endlichkeitssatzes.

Invarianz der Satzmenge: CUT(S) ⊆ S

Es genügt nicht zu zeigen, daß aus zwei allgemeingültigen Klauseln nur wieder eine allgemeingültige Klausel cut-abgeleitet werden kann.

Wir führen wieder einen induktiven Beweis.

Sei also $H \in \mathrm{CUT}\ (S)$. Der Induktionsanfang ist trivial.

Seien nun $C_1, C_2 \in$ **CUT(S)** und gelte **cut(**C_1, C_2**, H).** Nach Induktionsvoraussetzung können wir für C_1, C_2 annehmen, daß sie auch aus S sind (d.h. die Behauptung gilt schon für C_1, C_2). Nach der Diskussion zur Satzmenge ist klar, daß in jeder der beiden Klauseln C_i, ein Paar $(v_i, \neg v_i)$ vorkommen muß. Wenn der **cut** nicht bei einer der Variablen v_i stattgefunden hat, bleibt also mindestens ein Paar $(v_i, \neg v_i)$ in **H**. Nun der Fall, daß bei einem v_i geschnitten wird. Wir unterscheiden die Fälle, daß in beiden Klauseln dasselbe Variablenpaar oder verschiedene v_1 bzw. v_2 vorkommen.

Im ersten Fall $v_1 = v_2$ ist trotz Streichung noch jeweils das entgegengesetzte Literal zu v_1 in den C_i, also auch in **H,** während im zweiten Fall ebenfalls mindestens ein "entgegengesetztes" **(v, ¬v)** Variablenpaar in **H** übrig beleibt. Also ist H auch allgemeingültig, d.h. in S. w.z.b.w.