Theorembeweisen in PL 1

Vorlesung 2 Stunden
WS 2003/2004

J. Avenhaus
Universität Kaiserslautern

Inhaltsverzeichnis

1 Motivation 1
2 Prädikatenlogik 1. Stufe 5
  2.1 Syntax und Semantik 5
  2.2 Transformation in Klausel-Normalform 17
  2.3 Der Satz von Herbrand 36
3 Reine Gleichheitslogik 47
  3.1 Der generelle Ansatz 47
  3.2 All-quantifizierte Ziele 52
  3.3 Allgemeinere Ziele 78
4 Theorembeweisen in PL1 94
  4.1 Resolution und Paramodulation 99
  4.2 Superposition für Horn-Klauseln 133
  4.3 Saturierung bis auf Redundanz 160
  4.4 Superposition für beliebige Klauseln 186
  4.5 Ausnutzung der Redundanz 216
Literatur 229

1 Motivation

Problemstellung

Gegeben: $\mathcal{A}x$, Menge von Axiomen.
φ, Formel.

Frage: Gilt φ in allen Modellen von $\mathcal{A}x$
(oder $\mathcal{A}x \models \varphi$)?

Beispiel: „Nilpotente Monoide sind kommutativ"

$$\begin{array}{ll} \mathcal{A}x: & \forall x.\ \ x \cdot 1 = x \\ & \forall x.\forall y.\forall z.\ \ (x \cdot y) \cdot z = x \cdot (y \cdot z) \\ & \forall x.\ \ x \cdot x = 1 \end{array}$$

$$\varphi: \quad \forall x.\forall y.\ \ x \cdot y = y \cdot x$$

Dann: Es gilt $\mathcal{A}x \models \varphi$.

Motivation (Forts.)

Beispiel Doktor-Quacksalber-Problem

a) Problemstellung verbal:
Einige Patienten mögen alle Doktoren.
Kein Patient mag einen Quacksalber.
Also ist kein Doktor ein Quacksalber.

b) Problemstellung formal:

$$\begin{array}{ll} \mathcal{A}x: & \exists x.(\mathsf{pat}(x) \wedge \forall y.(\mathsf{doc}(y) \Rightarrow \mathsf{mag}(x,y))) \\ & \forall x.(\mathsf{pat}(x) \Rightarrow \forall y.(\mathsf{qua}(y) \Rightarrow \neg\mathsf{mag}(x,y))) \end{array}$$

$$\varphi: \quad \forall x.(\mathsf{doc}(x) \Rightarrow \neg\mathsf{qua}(x))$$

Es gilt $\mathcal{A}x \models \varphi$.

Frage

Wie zeigt man dies?

Vorsicht

Das Problem ist unentscheidbar, aber rekursiv aufzählbar.

$\Rightarrow$ Der Suchraum ist i. Allg. unendlich.

$\Rightarrow$ Gute Suchheuristiken nötig.

Ansatz für Beweissysteme

1. Syntax + Semantik festlegen

Was ist eine Spezifikation?

Was heißt genau $\mathcal{A}x \models \varphi$?

2. Widerlegungsverfahren

Es gilt $\mathcal{A}x \models \varphi \text{ gdw } \mathcal{A}x \cup \{\neg\varphi\}$ ist unerfüllbar.

Zeige: Eine Menge Φ von Formeln ist unerfüllbar.

3. Transformation in Klauselform

Transformiere Φ in eine Menge S von Klauseln.

Sei $\Box$ die leere Klausel.

4. Inferenzsystem auf Klauseln

Angabe von Inferenzsystem $\mathcal{I}$ mit

$S \overset{*}{\vdash}_{\mathcal{I}} \Box \text{ gdw } \Phi$ ist unerfüllbar.

5. Transformation in einen Beweiser

Gute Datenstrukturen + Algorithmen.

Geschickte Architektur des Beweisers.

Suchraumeinschränkung.

Gute Suchheuristiken.

Voraussetzungen für die Vorlesung

Vorausgesetzt werden Grundkenntnisse aus der Logik

Aussagenlogik

- logische Operatoren
- Syntax + Semantik
- Folgerungsbegriff

Prädikatenlogik

a) Syntax
   Quantoren $\exists$ und $\forall$
   Aufbau der Terme und Formeln
   Spezifikationen

b) Semantik
   sig-Algebren $\mathcal{A}$
   Gültigkeit von Formeln in $\mathcal{A}$
   Gültigkeit in allen Modellen einer Spezifikation

Wiederholungen

Alle wesentliche Begriffe werden definiert und kurz motiviert.

Ziel

Effiziente Beweiser für die Prädikatenlogik 1. Stufe.

2 Prädikatenlogik 1. Stufe

2.1 Syntax und Semantik

Bemerkung

Beschränkung auf nur eine Sorte ANY (Die Verallgemeinerung auf mehrere Sorten ist einfach.)

Hilfssorte BOOL: Prädikatssymbole werden als BOOL-wertige Funktionssymbole behandelt.

Sei

F=Menge der Funktionssymbole.

P=Menge der Prädikatssymbole.

Stelligkeit $\alpha : F \cup P \to \mathbb{N}$ mit

$\alpha(f) = n$ heißt f : ANY$^n \to$ ANY für $f \in F$, $n \geq 0$
$\alpha(p) = n$ heißt p : ANY$^n \to$ BOOL für $p \in P$, $n \geq 0$

Definition 2.1.

Eine **Signatur** ist sig $= (F, P, \alpha)$ mit $F \cap P = \varnothing$,
$\alpha : F \cup P \to \mathbb{N}$ wie oben.

Bemerkung

Die $c \in F$ mit $\alpha(c) = 0$ heißen Konstanten.

Die $p \in P$ mit $\alpha(p) = 0$ heißen aussagenlogische Symbole.

Terme und Formeln (1)

Sei sig $= (F, P, \alpha)$ eine Signatur,
V= Menge von Variablen (der Sorte ANY).

Bezeichnungen

Die Menge Term(F, V) der **Terme** ist wie üblich definiert.

Ein **Atom** ist $p(t_1, \ldots, t_n)$ mit $p \in P$, $\alpha(p) = n$, t_i Terme.

Eine **Formel** φ ist

- ein Atom oder
- $\neg\varphi_1$, $\varphi_1 \wedge \varphi_2$, $\varphi_1 \vee \varphi_2$, $\varphi_1 \Rightarrow \varphi_2$, $\varphi_1 \Leftrightarrow \varphi_2$, $\exists x.\varphi_1$ oder $\forall x.\varphi_1$, falls φ_1, φ_2 Formeln sind.

Sei Form (F, P, V) die Menge aller Formeln.

Die Menge free(φ) der **freien Variablen** in φ ist

free(φ) =var(φ)	falls	φ Atom
free(φ_1)		$\varphi \equiv \neg\varphi_1$
free$(\varphi_1) \cup$ free(φ_2)		$\varphi \equiv \varphi_1 * \varphi_2$
free$(\varphi_1) \setminus \{x\}$		$\varphi \equiv \exists x.\varphi_1$ oder $\varphi \equiv \forall x.\varphi_1$

Hier ist $* \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$.

Eine Formel φ heißt **abgeschlossen**, falls free$(\varphi) = \varnothing$ gilt.

Terme und Formeln (2)

Bezeichnungen

Die Menge Pos(φ) der **Positionen** in φ ist

- wie üblich, falls φ Atom.
- $\{1.\pi \mid \pi \in \text{Pos}(\varphi_1)\} \cup \{\lambda\}$, falls $\varphi \equiv \neg\varphi_1, \forall x.\varphi_1, \exists x.\varphi_1$
- $\{1.\pi \mid \pi \in \text{Pos}(\varphi_1)\} \cup \{2.\pi \mid \pi \in \text{Pos}(\varphi_2)\} \cup \{\lambda\}$, falls $\varphi \equiv \varphi_1 * \varphi_2$

Ein **Literal** l ist A oder $\neg A$, wobei A ein Atom ist.

Eine **Klausel** $C \equiv l_1 \vee \cdots \vee l_n$ ist eine Disjunktion von Literalen.

Schreibweisen

$\forall x_1, \ldots, x_n.\varphi$ statt $\forall x_1. \cdots \forall x_n.\varphi$

$C = \{l_1, \ldots, l_n\}$ statt $C \equiv l_1 \vee \cdots \vee l_n$

$\bar{x}_n$ statt $x_1, \ldots, x_n$ und $\bar{t}_n$ statt $t_1, \ldots, t_n$

$\varphi_1 \vee \varphi_2 \vee \varphi_3$ ohne Klammern, da $\vee$ assoziativ.

Also

Klauseln sind Multimengen von Literalen.

Semantik (1)

Sei sig $= (F, P, \alpha)$ eine Signatur,
$V =$ Menge von Variablen.

Definition 2.2.

Eine **sig-Algebra** $\mathcal{A} = (A;\ F^{\mathcal{A}}, P^{\mathcal{A}})$ besteht aus
$A \neq \emptyset$, eine Menge (das Universum),
$F^{\mathcal{A}} = \{f^{\mathcal{A}} \mid f \in F\}$ mit $f^{\mathcal{A}} : A^n \to A$, falls $\alpha(f) = n$,
$P^{\mathcal{A}} = \{p^{\mathcal{A}} \mid p \in P\}$ mit $p^{\mathcal{A}} \subseteq A^n$, falls $\alpha(p) = n$.

Intention

Ein Funktionssymbol $f \in F$ wird durch eine Funktion $f^{\mathcal{A}}$ interpretiert.

Ein Prädikatssymbol $p \in P$ wird durch ein Prädikat $p^{\mathcal{A}}$ interpretiert. Wir identifizieren ein n-stelliges Prädikat $p^{\mathcal{A}}$ auf A mit einer Teilmenge von A^n und einer n-stelligen Relation auf A.

Die Variablen $x \in V$ werden durch eine **Evaluierungsfunktion** eval : $V \to A$ interpretiert.

Semantik (2)

Sei sig $= (F, P, \alpha)$ eine Signatur,
$(\mathcal{A}, \text{eval})$ eine sig-Algebra mit Evaluierungsfunktion.

Definition 2.3.

Der **Wahrheitswert** $\text{eval}(\varphi) = \text{eval}_{\mathcal{A}}(\varphi) \in \{\text{true}, \text{false}\}$ einer Formel φ in $\mathcal{A}$ ist rekursiv erklärt durch

- $\text{eval}(c) = c^{\mathcal{A}}$
- $\text{eval}(f(t_1, \ldots, t_n)) = f^{\mathcal{A}}(\text{eval}(t_1), \ldots, \text{eval}(t_n))$
- $\text{eval}(p(t_1, \ldots, t_n)) = \text{true gdw } (\text{eval}(t_1), \ldots, \text{eval}(t_n)) \in p^{\mathcal{A}}$
- $\text{eval}(\neg\varphi) = \neg\text{eval}(\varphi)$
- $\text{eval}(\varphi_1 * \varphi_2) = \text{eval}(\varphi_1) * \text{eval}(\varphi_2)$ für $* \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$
- $\text{eval}(\forall x.\varphi) = \text{true gdw}$ für alle $a \in A$ gilt $\text{eval}^{x,a}(\varphi) = \text{true}$
- $\text{eval}(\exists x.\varphi) = \text{true gdw}$ für ein $a \in A$ gilt $\text{eval}^{x,a}(\varphi) = \text{true}$

Dabei ist

$$\text{eval}^{x,a}(y) = \begin{cases} a & y = x \\ \text{eval}(y) & y \neq x \end{cases}$$

Semantik (3)

Lemma 2.1

Seien φ abgeschlossen, $\mathcal{A}$ eine sig-Algebra und eval, eval′ Evaluierungsfunktionen.

Dann gilt $\text{eval}(\varphi) = \text{eval}'(\varphi)$.

Bezeichnung

$\mathcal{A} \models \varphi$, falls für alle Evaluierungsfunktionen $\text{eval}_{\mathcal{A}}$ gilt $\text{eval}_{\mathcal{A}}(\varphi) = \text{true}$.

Definition 2.4.

Eine **Spezifikation** spec $= (\text{sig}, \mathcal{A}x)$ besteht aus einer Signatur sig und einer Menge $\mathcal{A}x$ von abgeschlossenen Formeln (Axiomen).

Eine sig-Algebra $\mathcal{A}$ ist ein **Modell** von spec, falls $\mathcal{A} \models \varphi$ für alle $\varphi \in \mathcal{A}x$ gilt.

Bezeichnung

Mod(spec) ist die Klasse aller Modelle von spec.

Definition 2.5.

a) φ heißt **erfüllbar**, falls es eine sig-Algebra $\mathcal{A}$ gibt mit $\mathcal{A} \models \varphi$.

b) $\mathcal{A}x \models \varphi$ gdw $\mathcal{A} \models \varphi$ für alle $\mathcal{A} \in \text{Mod}(\text{sig}, \mathcal{A}x)$.

Beispiele

Beispiel 1

Sei $F = \emptyset,\ \ P = \{p\},\ \ \varphi \equiv \forall x.\exists y.p(x,y),\ \ A = \mathbb{N}$.

a) $\mathcal{A} = (\mathbb{N};\ P^{\mathcal{A}})$ mit $p^{\mathcal{A}}(x,y)$ gdw $x < y$.
Es gilt $\mathcal{A} \models \varphi$, denn zu jedem $x \in \mathbb{N}$ gibt es ein $y \in \mathbb{N}$ mit $x < y$.

b) $\mathcal{A} = (\mathbb{N};\ P^{\mathcal{A}})$ mit $p^{\mathcal{A}}(x,y)$ gdw $x > y$.
Es gilt nicht $\mathcal{A} \models \varphi$, denn für $x = 0$ gibt es kein $y \in \mathbb{N}$ mit $x > y$.

Beispiel 2

Sei $F = \emptyset, P = \{p\}$, $\mathcal{A}x = \{\varphi_1, \varphi_2, \varphi_3\}$ mit

- $\varphi_1 \equiv \forall x.p(x,x)$
- $\varphi_2 \equiv \forall x,y.(p(x,y) \Rightarrow p(y,x))$
- $\varphi_3 \equiv \forall x,y,z.(p(x,y) \wedge p(y,z) \Rightarrow p(x,z))$

Sei spec $=$ (sig, $\mathcal{A}x$).

Eine sig-Algebra $\mathcal{A}$ ist genau dann ein Modell von spec, wenn $p^{\mathcal{A}}$ eine Äquivalenzrelation ist.

Prädikatenlogik mit Gleichheit: Motivation

Bisher

Beschrieben ist die Prädikatenlogik 1. Stufe.

Bezeichnung: PL1 oder FOL.

Beachte: Jedes $p \in P$ kann beliebig interpretiert werden.

Jetzt

Prädikatenlogik 1. Stufe mit Gleichheit.

Bezeichnung: PL1= oder FOLE.

P enthält das Prädikatssymbol „=“ mit $\alpha(=) = 2$, es wird stets als das Gleichheitsprädikat interpretiert.

Beachte

$=$ muss in $\mathcal{A}$ stets als eine Kongruenzrelation interpretiert werden. Es muss z. B. gelten

$$\mathcal{A} \models t_1 = t_1' \wedge \cdots \wedge t_n = t_n' \Rightarrow f(t_1,\ldots,t_n) = f(t_1',\ldots,t_n')$$

$$\mathcal{A} \models t_1 = t_1' \wedge \cdots \wedge t_n = t_n' \wedge p(t_1,\ldots,t_n) \Rightarrow p(t_1',\ldots,t_n')$$

Prädikatenlogik mit Gleichheit: Formal

1. Ansatz: PL1=

Es bleibt alles wie in PL1, aber

1. $= \in P$ mit $\alpha(=) = 2$
2. bei festem $(\mathcal{A}, \mathsf{eval})$ gilt
   $\mathsf{eval}(t_1 = t_2) = \mathsf{true}\ \text{ gdw }\ \mathsf{eval}(t_1) = \mathsf{eval}(t_2)$ in $\mathcal{A}$.

Beachte

Hier wird $=$ stets fest interpretiert (als Gleichheit).

Frage

Kann man PL1= in PL1 einbetten?

Einbettung von PL1= in PL1

2.Ansatz für PL1

Bezeichnung

$\mathcal{A}x_=$ bestehe aus (für alle $f \in F$, $p \in P$)

$$\forall x.\ \ x = x$$

$$\forall x, y.\ \ (x = y \Rightarrow y = x)$$

$$\forall x, y, z.\ \ (x = y \land y = z \Rightarrow x = z)$$

$$\forall x_1, y_1, \dots, x_n, y_n.(x_1 = y_1 \land \cdots \land x_n = y_n \Rightarrow f(x_1, \dots, x_n) = f(y_1, \dots, y_n))$$

$$\forall x_1, y_1, \dots, x_n, y_n.(x_1 = y_1 \land \cdots \land x_n = y_n \land p(x_1, \dots, x_n) \Rightarrow p(y_1, \dots, y_n))$$

Satz 2.1.

Für jede Spezifikation spec $= (\mathsf{sig}, \mathcal{A}x)$ gilt

$\mathcal{A}x$ ist bezüglich PL1= erfüllbar $\ \text{gdw}$
$\mathcal{A}x \cup \mathcal{A}x_=$ ist bezüglich PL1 erfüllbar.

Beispiel zu PL1=

Sei

spec = (sig, $\mathcal{A}x$) mit

$$F = \{1, {}^{-1}, *\}, \; P = \{=\}, \; \mathcal{A}x = \{\varphi_1, \varphi_2, \varphi_3\}$$
$$\varphi_1 \equiv \forall x. \; x * 1 = x$$
$$\varphi_2 \equiv \forall x. \; x * x^{-1} = 1$$
$$\varphi_3 \equiv \forall x, y, z. \; (x * y) * z = x * (y * z)$$

Dann

Eine sig-Algebra $\mathcal{A}$ ist genau dann ein Modell von spec, wenn $\mathcal{A}$ eine Gruppe ist.

Beispiel $\mathcal{A} = (A, F^{\mathcal{A}}, P^{\mathcal{A}})$

a) $A = \mathbb{N}, \quad 1^{\mathcal{A}} = 0, \quad {}^{-1\mathcal{A}}=$id, $\quad *^{\mathcal{A}} = +$
   $\mathcal{A}$ ist kein Modell von spec.

b) $A = \mathbb{Z}, \quad 1^{\mathcal{A}} = 0, \quad {}^{-1\mathcal{A}} = -, \quad *^{\mathcal{A}} = +$
   $\mathcal{A}$ ist ein Modell von spec.

c) $A = \{0, 1\}, \quad 1^{\mathcal{A}} = 0, \quad {}^{-1\mathcal{A}}=$id, $\quad *^{\mathcal{A}} = +$ modulo 2
   $\mathcal{A}$ ist ein Modell von spec.

Der Gültigkeitsbegriff

Sei spec = (sig, $\mathcal{A}x$) eine Spezifikation und φ eine abgeschlossene Formel über sig.

Bezeichnung

φ **gilt** in spec, falls φ in allen Modellen von spec gilt.
Also: φ gilt in spec $\;\mathrm{gdw}\;\; \mathcal{A}x \models \varphi$
$\mathrm{gdw}\;\; \mathcal{A} \models \varphi$ für alle $\mathcal{A} \in$ Mod(spec).

Bemerkung

Es gibt verschiedene sinnvolle Gültigkeitsbegriffe zu einer Spezifikation. Wir wählen für diese Vorlesung den obigen Begriff. Er heißt auch „Gültigkeit in PL1".

Beachte

Sei $\mathcal{A}x = \{A_1, \ldots, A_n\}$, $A_i, \varphi \in$ Form(sig).

Es gilt $\mathcal{A}x \models \varphi \;\; \mathrm{gdw} \;\; A \equiv A_1 \wedge \cdots \wedge A_n \wedge \neg\varphi$ ist unerfüllbar.

Nächstes Ziel

Transformation von A in Klausel-Normalform.

2.2 Transformation in Klausel-Normalform

Bezeichnung

Eine Formel φ ist in **Klausel-Normalform (CNF)**, falls sie die Form hat

$$\varphi \equiv \forall^*(C_1 \wedge \cdots \wedge C_n)$$

wobei die C_i Klauseln sind.

Ziel (*)

Transformation von φ in φ' mit

(1) φ' ist in CNF.

(2) φ ist erfüllbar gdw φ' ist erfüllbar.

Dann heißt φ' eine **CNF für** φ.

Beachte

Ist φ' in CNF, so enthält φ' nicht den Quantor $\exists$ und die C_i sind ganz Quantoren-frei.

Zur Erfüllbarkeit (1)

Sei sig $= (F, P, \alpha)$ eine Signatur,
spec $=$ (sig, $\mathcal{A}x$) eine Spezifikation,
$\mathcal{A}x = \{\varphi_1, \ldots, \varphi_n\}$,
φ eine abgeschlossene Formel über sig.

Ziel des Theorembeweisens

Gegeben: $\mathcal{A}x, \varphi$ über sig.

Zeige: $\mathcal{A}x \models \varphi$.

Beachte

$\mathcal{A}x \models \varphi \ \mathrm{gdw}\ \varphi_1 \wedge \cdots \wedge \varphi_n \wedge \neg\varphi$ ist unerfüllbar.

Also

Ziel (*) ist sinnvoll.

Zur Erfüllbarkeit (2)

Definition 2.6.

Sei $T : \varphi \mapsto \psi$ eine Transformation von Formeln.

T **erhält die Erfüllbarkeit**, falls gilt

$$\varphi \text{ ist erfüllbar } \mathrm{gdw}\ \psi \text{ ist erfüllbar.}$$

T **erhält die Äquivalenz**, falls für jede sig-Algebra $\mathcal{A}$ gilt:

$$\mathcal{A} \models \varphi \ \ \mathrm{gdw} \ \ \mathcal{A} \models \psi$$

Beachte

Erhält T die Äquivalenz, so transformiert T sig-Formeln in sig-Formeln.

Die „Skolemisierung“ führt neue Funktionssymbole ein, erweitert also die Signatur. Sie erhält nur die Erfüllbarkeit, nicht die Äquivalenz.

Ziel

Angabe von T mit Transformationsregeln, so dass $T(\varphi)$ eine CNF für φ ist.

Intention

Die meisten Beweissysteme arbeiten auf Formeln in Klausel-Normalform viel effizienter als auf Formeln in beliebiger Form.

Simplifikationsregeln (1)

Bezeichnung

Einführung neuer Symbole $\top$ und $\bot$ (top und bottom).
$\top$ steht für true, $\bot$ steht für false.

Simplifikationsregeln

$\varphi \wedge \varphi$	$\rightarrow \varphi$	$\varphi \vee \varphi$	$\rightarrow \varphi$
$\varphi \wedge \neg\varphi$	$\rightarrow \bot$	$\varphi \vee \neg\varphi$	$\rightarrow \top$
$\varphi \Leftrightarrow \varphi$	$\rightarrow \top$	$\varphi \Rightarrow \varphi$	$\rightarrow \top$
$\neg\top$	$\rightarrow \bot$	$\neg\bot$	$\rightarrow \top$
$\varphi \wedge \top$	$\rightarrow \varphi$	$\varphi \vee \top$	$\rightarrow \top$
$\varphi \wedge \bot$	$\rightarrow \bot$	$\varphi \vee \bot$	$\rightarrow \varphi$
$\varphi \Rightarrow \top$	$\rightarrow \top$	$\top \Rightarrow \varphi$	$\rightarrow \varphi$
$\varphi \Rightarrow \bot$	$\rightarrow \neg\varphi$	$\bot \Rightarrow \varphi$	$\rightarrow \top$
$\varphi \Leftrightarrow \top$	$\rightarrow \varphi$	$\varphi \Leftrightarrow \bot$	$\rightarrow \neg\varphi$
$\forall x.\varphi$	$\rightarrow \varphi$	falls $x \notin$ free(φ)	
$\exists x.\varphi$	$\rightarrow \varphi$	falls $x \notin$ free(φ)	

Beachte hier und in den folgenden Regeln:

$\wedge$ und $\vee$ sind kommutativ und assoziativ. Also gilt z. B. auch

$$\neg\varphi \wedge \varphi \rightarrow \bot \qquad \neg\varphi \vee \varphi \rightarrow \top$$

Simplifikationsregeln (2)

Lemma 2.2

Die Anwendung der Regeln terminiert.

Alle Regeln sind Äquivalenz-erhaltend.

Ist φ irreduzibel, so gilt $\varphi \in \{\bot, \top\}$ oder φ enthält $\bot, \top$ nicht.

Beachte

Sei $\varphi \xrightarrow{*} \psi$ mit den Simplifikationsregeln und ψ irreduzibel.

Ist $\psi \in \{\bot, \top\}$ so stoppt die CNF-Transformation.

Ab jetzt: Sei ψ $\{\bot, \top\}$-frei.

Negations-Normal-Form

Bezeichnung

φ ist in **Negations-Normal-Form (NNF)**, falls

(1) φ enthält $\Rightarrow, \Leftrightarrow$ nicht.

(2) φ enthält $\neg$ nur direkt vor Atomen.

NNF-Regeln

$$\begin{array}{llll}
\neg(\varphi \wedge \psi) \to \neg\varphi \vee \neg\psi & & \neg(\varphi \vee \psi) \to \neg\varphi \wedge \neg\psi & \\
\neg\forall x.\varphi & \to \exists x.\neg\varphi & \neg\exists x.\varphi & \to \forall x.\neg\varphi \\
\varphi \Rightarrow \psi & \to \neg\varphi \vee \psi & \neg\neg\varphi & \to \varphi \\
\varphi \Leftrightarrow \psi & \to (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi) & &
\end{array}$$

Lemma 2.3

Das bisherige Regelsystem ist terminierend. Alle Regeln sind Äquivalenz-erhaltend. Gilt $\varphi \xrightarrow{*} \psi$ und ist ψ irreduzibel, so ist ψ in NNF.

Miniscoping

Ziel

Die Quantoren möglichst weit nach innen ziehen.

Miniscoping-Regeln

$\exists x.(\varphi \wedge \psi) \rightarrow \exists x.\varphi \wedge \psi$	falls $x \notin$ free(ψ)
$\exists x.(\varphi \vee \psi) \rightarrow \exists x.\varphi \vee \psi$	falls $x \notin$ free(ψ)
$\forall x.(\varphi \wedge \psi) \rightarrow \forall x.\varphi \wedge \psi$	falls $x \notin$ free(ψ)
$\forall x.(\varphi \vee \psi) \rightarrow \forall x.\varphi \vee \psi$	falls $x \notin$ free(ψ)
$\forall x.(\varphi \wedge \psi) \rightarrow \forall x.\varphi \wedge \forall x.\psi$	falls $x \in$ free(φ) und $x \in$ free(ψ)
$\exists x.(\varphi \vee \psi) \rightarrow \exists x.\varphi \vee \exists x.\psi$	falls $x \in$ free(φ) und $x \in$ free(ψ)

Lemma 2.4

Das bisherige Regelsystem ist terminierend.

Alle Regeln sind Äquivalenz-erhaltend.

Beispiel zum Miniscoping

Beispiel (*)

$\forall x.\exists y.\forall z.\,(r(x,x) \wedge [p(y) \vee r(x,y) \vee q(z)])$

$\rightarrow \forall x.\exists y.\,(r(x,x) \wedge \forall z.[p(y) \vee r(x,y) \vee q(z)])$

$\xrightarrow{*} \forall x.\exists y.\,(r(x,x) \wedge [p(y) \vee r(x,y) \vee \forall z.q(z)])$

$\rightarrow \forall x.\,(r(x,x) \wedge \exists y.[p(y) \vee r(x,y) \vee \forall z.q(z)])$

$\xrightarrow{*} \forall x.\,(r(x,x) \wedge [\exists y.p(y) \vee \exists y.r(x,y) \vee \forall z.q(z)])$

$\rightarrow \forall x.r(x,x) \wedge \forall x.[\exists y.p(y) \vee \exists y.r(x,y) \vee \forall z.q(z)]$

$\rightarrow \forall x.r(x,x) \wedge [\exists y.p(y) \vee \forall x.(\exists y.r(x,y) \vee \forall z.q(z))]$

$\rightarrow \forall x.r(x,x) \wedge [\exists y.p(y) \vee \forall x.\exists y.r(x,y) \vee \forall z.q(z)]$

Beachte

Es gilt nicht

$\forall x.(p(x) \vee q(x)) \rightarrow \forall x.p(x) \vee \forall x.q(x)$

$\exists x.(p(x) \wedge q(x)) \rightarrow \exists x.p(x) \wedge \exists x.q(x)$

Diese Regeln sind weder Äquivalenz- noch Erfüllbarkeit-erhaltend.

Variablen-Umbenennung

Regel

Benenne in φ die gebundenen Variablen so um, dass keine Variable mehrfach gebunden ist und keine gebundene Variable frei in φ vorkommt.

Beispiele

$\forall x.p(x) \vee \forall x.q(x) \rightarrow \forall x.p(x) \vee \forall x'.q(x')$

$\forall x.(\exists y.r(x,y) \wedge \exists y.r(y,x)) \rightarrow \forall x.(\exists y.r(x,y) \wedge \exists y'.r(y',x))$

$\forall x.(\forall y.r(x,y) \wedge \forall y.r(y,x)) \rightarrow \forall x.(\forall y.r(x,y) \wedge \forall y'.r(y',x))$

Beispiel (*) liefert

$\forall x.r(x,x) \wedge [\exists y.p(y) \vee \forall x.\exists y.r(x,y) \vee \forall z.q(z)]$

$\xrightarrow{*} \forall y_1.r(y_1,y_1) \wedge [\exists y_2.p(y_2) \vee \forall y_3.\exists y_4.r(y_3,y_4) \vee \forall y_5.q(y_5)]$

Lemma 2.5

Die Regel ist Äquivalenz-erhaltend.

Skolemisierung (1)

Ziel

Elimination der Existenz-Quantoren $\exists$.

Beachte zur Motivation

$\mathcal{A}$ erfüllt $\forall x.\exists y.r(x,y)$ gdw

$\mathcal{A}' = \mathcal{A}$ erweitert um das „neue g“ erfüllt $\forall x.r(x,g(x))$.
(Man muss das g in $\mathcal{A}'$ geeignet interpretieren.)

Allgemeiner: Sei g neues Funktionssymbol.

$\varphi \wedge \forall x_1,\ldots,x_n.\exists y.r(x_1,\ldots,x_n,y)$ ist erfüllbar gdw

$\varphi \wedge \forall x_1,\ldots,x_n.r(x_1,\ldots,x_n,g(x_1,\ldots,x_n))$ ist erfüllbar.

Beachte zur $\exists$-Elimination.

1. $\exists$-Elimination „von außen nach innen“, d. h. $\exists y.$ steht nicht im Wirkungsbereich eines anderen $\exists$-Quantors.
2. Jeder Eliminationsschritt führt neues Funktionssymbol ein.
   Also: Signatur-Erweiterung.

Skolemisierung (2)

Regel

$$\varphi[\exists x.\psi]_\pi \to \varphi[\pi \leftarrow \psi[x \mapsto g(y_1, \ldots, y_n)]]$$

falls free($\exists x.\psi) = \{y_1, \ldots, y_n\}$, g neu,
π minimale Position in φ mit $\varphi/\pi \equiv \exists x.\psi$.

Lemma 2.6

Das bisherige Regelsystem ist terminierend.

Diese Regel ist Erfüllbarkeit-erhaltend.

Also sind alle Regeln Erfüllbarkeit-erhaltend.

Beispiel (*) liefert

$\forall y_1.r(y_1, y_1) \land [\exists y_2.p(y_2) \lor \forall y_3.\exists y_4.r(y_3, y_4) \lor \forall y_5.q(y_5)]$

$\xrightarrow{*} \forall y_1.r(y_1, y_1) \land [p(a) \lor \forall y_3.r(y_3, g(y_3)) \lor \forall y_5.q(y_5)]$

mit a, g neu.

Erstellung der CNF (1)

Beachte

Sei $\varphi \xrightarrow{*} \psi$ und ψ irreduzibel. Dann gilt:

ψ enthält nur noch $\neg$, $\lor$, $\land$ und den Quantor $\forall$.

Es steht $\neg$ nur direkt vor Atomen.

Jetzt

a) Alle $\forall$-Quantoren ganz nach vorn ziehen.
Also: $\psi \xrightarrow{*} \forall^*\psi_0$, die „Matrix“ ψ_0 ist $\forall$-frei.

b) Transformation von ψ_0 in Konjunktion von Disjunktionen.

Beachte

Jede Variable in ψ ist nur einmal gebunden.

Also ist die Regel $\forall x.\psi' \lor \psi'' \to \forall x.(\psi' \lor \psi'')$,
angewandt auf ψ, Äquivalenz-erhaltend, da $x \notin$ free(ψ'').

Erstellung der CNF (2)

Finale Regeln

$\forall x.\varphi \wedge \psi \rightarrow \forall x.(\varphi \wedge \psi)$ falls $x \notin$ free(ψ)
$\forall x.\varphi \vee \psi \rightarrow \forall x.(\varphi \vee \psi)$ falls $x \notin$ free(ψ)
$\varphi \vee (\psi_1 \wedge \psi_2) \rightarrow (\varphi \vee \psi_1) \wedge (\varphi \vee \psi_2)$

Beispiel (*) liefert

$\forall y_1.r(y_1, y_1) \wedge [p(a) \vee \forall y_3.r(y_3, g(y_3)) \vee \forall y_5.q(y_5)]$

$\xrightarrow{*} \forall y_1, y_3, y_5.(r(y_1, y_1) \wedge [p(a) \vee r(y_3, g(y_3)) \vee q(y_5)])$

Dies ist in Klausel-Normalform. Die Klauselmenge ist

$$S = \{r(y_1, y_1), p(a) \vee r(y_3, g(y_3)) \vee q(y_5)\}$$

Beispiele

1. Beispiel $\forall$-$\exists$-Vertauschung

a) $\varphi \equiv \forall x.\exists y.\ x \cdot y = 1$
Skolemisierung liefert $\forall x.\ x \cdot g(x) = 1.$

b) $\varphi \equiv \exists y.\forall x.\ x^{-1} \cdot x = y$
Skolemisierung liefert $\forall x.\ x^{-1} \cdot x = a.$

c) $\varphi \equiv \neg\exists y.\forall x.\ x^{-1} \cdot x = y$
Es ergibt sich φ
$\varphi \xrightarrow{*} \forall y.\exists x.\ \neg x^{-1} \cdot x = y \rightarrow \forall y.\ \neg g(y)^{-1} \cdot g(y) = y.$

2. Beispiel Quacksalber

$\varphi \equiv \exists x.(\mathrm{pat}(x) \wedge \forall y.(\mathrm{doc}(y) \Rightarrow \mathrm{mag}(x, y)))$

$\rightarrow \exists x.(\mathrm{pat}(x) \wedge \forall y.(\neg\mathrm{doc}(y) \vee \mathrm{mag}(x, y)))$

$\rightarrow (\mathrm{pat}(a) \wedge \forall y.(\neg\mathrm{doc}(y) \vee \mathrm{mag}(a, y)))$

$\rightarrow \forall y.(\mathrm{pat}(a) \wedge (\neg\mathrm{doc}(y) \vee \mathrm{mag}(a, y)))$

Die Klauselmenge zu φ ist also
$S = \{\mathrm{pat}(a), \neg\mathrm{doc}(y) \vee \mathrm{mag}(a, y)\}.$

Erstellung der CNF (3)

Sei T die durch die Regeln beschriebene Transformation:

Erst die Simplifikationsregeln anwenden, dann die Regeln bis zur Skolemisierung anwenden, dann die finalen Regeln anwenden (jeweils so lange wie möglich).

Satz 2.2.

Die Transformation $T : \varphi \mapsto \psi$ terminiert. Sie ist Erfüllbarkeit-erhaltend. Es ist $\psi = T(\varphi)$ eine CNF für φ.

Bezeichnung

Die Klauselmenge $S = \{C_1, \ldots, C_n\}$ ist eine **Klauselmenge zu** φ, falls gilt:

φ ist erfüllbar gdw $\forall^*(C_1 \wedge \cdots \wedge C_n)$ ist erfüllbar.

Ergebnis zur CNF

Satz 2.3.

Sei $\mathcal{A}x = \{\varphi_1, \ldots, \varphi_n\}$ und ψ eine CNF zu $\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \neg\varphi$.

Es gilt

$\mathcal{A}x \models \varphi$ gdw ψ ist unerfüllbar.

Beweis:

Zusammenfassung der bisherigen Überlegungen.

Frage also

Wie zeigt man, dass eine Formel ψ in CNF unerfüllbar ist?

Beachte

Die CNF zu $\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \neg\varphi$ ist nicht eindeutig. Es ist also vermutlich ratsam, eine kleine CNF (also eine kleine Klauselmenge S) zu berechnen.

Zur Berechnung einer Klauselmenge

Sei

$\mathcal{A}x = \{\varphi_1, \ldots, \varphi_n\}$ und φ gegeben.

Berechne Klauselmengen $S_1, \ldots, S_n, S_0$ zu $\varphi_1, \ldots, \varphi_n, \neg\varphi$. Benutze dabei keine Skolemfunktion mehrfach.

Dann ist $S = S_1 \cup \cdots \cup S_n \cup S_0$ eine Klauselmenge zu $\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \neg\varphi$.

Also

$\varphi_1 \wedge \cdots \wedge \varphi_n \models \varphi \text{ gdw } S$ ist unerfüllbar.

Beachte

Sei $\{x_1, \ldots, x_n\} = \mathsf{Var}(D)$ und $\sigma = \{x_i \leftarrow y_i \mid i = 1, \ldots, n\}$.

Dann sind $\forall x_1, \ldots, x_n.(C \wedge D)$ und $\forall x_1, \ldots, x_n.C \wedge \forall y_1, \ldots, y_n.\sigma(D)$ logisch äquivalent.

Also

Man kann durch Variablenumbenennung erreichen, dass in $S = \{C_1, \ldots, C_n\}$ die Klauseln C_i paarweise Variablen-disjunkt sind.

Beispiel Quacksalber

Sei

$\varphi_1 \equiv \exists x.(\mathsf{pat}(x) \wedge \forall y.(\mathsf{doc}(y) \Rightarrow \mathsf{mag}(x, y)))$

$\varphi_2 \equiv \forall x.(\mathsf{pat}(x) \Rightarrow \forall y.(\mathsf{qua}(y) \Rightarrow \neg\mathsf{mag}(x, y)))$

$\varphi \equiv \forall x.(\mathsf{doc}(x) \Rightarrow \neg\mathsf{qua}(x))$

Transformation in CNF liefert

$\varphi_1 \xrightarrow{*} \forall y.(\mathsf{pat}(a) \wedge (\neg\mathsf{doc}(y) \vee \mathsf{mag}(a, y)))$

$\varphi_2 \xrightarrow{*} \forall x, y.(\neg\mathsf{pat}(x) \vee \neg\mathsf{qua}(y) \vee \neg\mathsf{mag}(x, y))$

$\neg\varphi \xrightarrow{*} \mathsf{doc}(b) \wedge \mathsf{qua}(b)$

Also

$\varphi_1 \wedge \varphi_2 \models \varphi \text{ gdw } S$ ist unerfüllbar.

S:	$\mathsf{pat}(a)$, $\neg\mathsf{doc}(y) \vee \mathsf{mag}(a, y)$	aus φ_1
	$\neg\mathsf{pat}(x) \vee \neg\mathsf{qua}(y') \vee \neg\mathsf{mag}(x, y')$	aus φ_2
	$\mathsf{doc}(b)$, $\mathsf{qua}(b)$	aus $\neg\varphi$

Bemerkungen

1. Bemerkung

Es gibt viele Möglichkeiten, Klauselmengen zu vereinfachen.

Beispiel

Klausel C ist eine **Tautologie**, falls sie komplementäre Literale A und $\neg A$ enthält (A ein Atom).

Ist $S \cup \{C\}$ eine Klauselmenge zu φ und C eine Tautologie, so ist auch S eine Klauselmenge zu φ.

2. Bemerkung

Beschrieben ist eine Standardtransformation in CNF. Sie liefert i. Allg. eine zu große Klauselmenge.

Für Verbesserungen siehe
A. Nonnengart + C. Weidenbach.
Computing small clause normal forms
in: Robinson + Voronkov (Eds.): Handbook of Automated Reasoning, Elsevier, 2001.

2.3 Der Satz von Herbrand

Frage

Wie zeigt man die Unerfüllbarkeit einer Formel φ in CNF?
(Es gibt viele sig-Algebren $\mathcal{A}$, die φ erfüllen könnten!)

Hier

Es reicht, solche $\mathcal{A}$ zu betrachten, bei denen die Grundterme über sig das Universum bilden.

$\Rightarrow$ Herbrand-Interpretationen.

Bezeichnung

Ein Objekt (Term, Gleichung, Atom, ...) heißt ein **Grundobjekt**, falls es keine Variablen enthält.

Sei Term(F) die Menge der **Grundterme** über sig $= (F, P, \alpha)$.

Ab jetzt

sig $= (F, P, \alpha)$ ist eine Signatur. Es gibt eine Konstante $c \in F$.

φ ist eine Formel über sig in CNF.

S ist eine Klauselmenge über sig, die Klauseln sind paarweise Variablen-disjunkt.

Bezeichnungen

Klauseln

Schreibweise einer Klausel $C \equiv \Gamma \Rightarrow \Delta$ mit Γ, Δ Multimengen von Atomen:

- Ist $\Gamma = \{A_1, \ldots, A_n\}$ und $\Delta = \{B_1, \ldots, B_m\}$, so ist $C \equiv \neg A_1 \vee \cdots \vee \neg A_n \vee B_1 \vee \cdots \vee B_m$.
- Γ ist der **Antezedent**, Δ ist der **Sukzedent** von C.
- Die A_i (B_i) sind die negativen (positiven) Literale in C.
- Es ist $\Box$ die **leere Klausel**, d. h. $n = m = 0$.
- C ist eine **Horn-Klausel**, falls $m \leq 1$.
- $A, \Gamma, \Gamma' \Rightarrow \Delta, B$ steht für $\{A\} \cup \Gamma \cup \Gamma' \Rightarrow \Delta \cup \{B\}$.

Substitutionen

Eine Substitution σ ist gegeben durch $\sigma : V \to \text{Term}(F, V)$.

Dann
$$\begin{aligned}
&\sigma(f(t_1, \ldots, t_n)) \equiv f(\sigma(t_1), \ldots, \sigma(t_n))\\
&\sigma(p(t_1, \ldots, t_n)) \equiv p(\sigma(t_1), \ldots, \sigma(t_n))\\
&\sigma(\Gamma) \equiv \{\sigma(A) \mid A \in \Gamma\}\\
&\sigma(\Gamma \Rightarrow \Delta) \equiv \sigma(\Gamma) \Rightarrow \sigma(\Delta)
\end{aligned}$$

Herbrand-Strukturen

Definition 2.7.

Eine sig-Algebra $\mathcal{A} = (A;\ F^{\mathcal{A}}, P^{\mathcal{A}})$ über sig $= (F, P, \alpha)$ ist eine **Herbrand-Algebra**, falls

$$\begin{array}{ll}
A = \text{Term}(F) & \\
f^{\mathcal{A}}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n) & \text{für } f \in F,\ t_1, \ldots, t_n \in A\\
p^{\mathcal{A}} \subseteq A^n & \text{für } p \in P
\end{array}$$

Also

Eine Herbrand-Algebra $\mathcal{A}$
- hat die Grundterme als Elemente des Universums,
- interpretiert die $f \in F$ in kanonischer Weise,
- interpretiert die $p \in P$ als Prädikate auf $A = \text{Term}(F)$.

Sie ist festgelegt durch die **Herbrand-Interpretation**
$$I = \{p(t_1, \ldots, t_n) \mid (t_1, \ldots, t_n) \in p^{\mathcal{A}}\}$$

Dies steht für
$$I(p(t_1, \ldots, t_n)) = \begin{cases} \text{true} & \text{falls } (t_1, \ldots, t_n) \in p^{\mathcal{A}}\\ \text{false} & \text{falls } (t_1, \ldots, t_n) \notin p^{\mathcal{A}} \end{cases}$$

Beachte

Eine Herbrand-Algebra $\mathcal{A}$ (**ein H-Modell**) ist eindeutig durch die Interpretation I der Grundatome $p(t_1, \ldots, t_n)$, d. h., deren Belegung mit Wahrheitswerten, festgelegt.

Herbrand-Erfüllbarkeit

Sei $\varphi \equiv \forall^*(C_1 \wedge \cdots \wedge C_n)$ eine Formel in CNF.
$S_\varphi = \{C_1, \ldots, C_n\}$.
$\mathcal{A}$ eine H-Algebra, festgelegt durch I.

Bezeichnung

- I erfüllt Grundatom $p(t_1, \ldots, t_n)$ gdw $p(t_1, \ldots, t_n) \in I$.
- I erfüllt Grundklausel $\Gamma \Rightarrow \Delta$ gdw I erfüllt ein Atom aus Γ nicht oder I erfüllt ein Atom aus Δ.
- I erfüllt die Klausel C, falls für jede Grundsubstitution σ gilt: I erfüllt $\sigma(C)$.
- I erfüllt eine Klauselmenge S, falls I alle $C \in S$ erfüllt.
- gnd$(S) = \{\sigma(C) \mid C \in S, \sigma(C)$ ist eine Grundklausel $\}$

Lemma 2.7

$\mathcal{A}$ erfüllt φ gdw I erfüllt gnd(S_φ).

Definition 2.8.

φ ist **Herbrand-erfüllbar**, falls es eine H-Algebra $\mathcal{A}$ gibt, die φ erfüllt (also: falls es eine H-Interpretation I gibt, die gnd(S_φ) erfüllt).

Hauptresultat für PL1

Satz 2.4. (Gödel-Herbrand-Skolem)

Eine Formel φ in CNF ist erfüllbar genau dann, wenn sie Herbrand-erfüllbar ist.

Beweis:

- Sei $\varphi \equiv \forall x_1, \ldots, x_m.M$ und $M \equiv C_1 \wedge \cdots \wedge C_n$.
- Ist φ H-erfüllbar, so ist φ auch erfüllbar.
- Sei φ erfüllbar, etwa durch sig-Algebra $\mathcal{A} = (A, F^{\mathcal{A}}, P^{\mathcal{A}})$, d. h., $\mathcal{A} \models \varphi$.
- Wir konstruieren eine H-Interpretation I, die alle $\sigma(C_i)$ mit $i = 1, \ldots, n$, σ Grundsubstitution erfüllt: Setze

  $$I = \{p(t_1, \ldots, t_k) \mid \mathcal{A} \models p(t_1, \ldots, t_k), t_j \in \mathrm{Term}(F)\}$$

- Wegen $\mathcal{A} \models \varphi$ gilt $\mathcal{A} \models \forall x_1, \ldots, x_m.C_i$,
  also auch für alle Grundsubstitutionen σ und alle $i = 1, \ldots, n$:
  $\mathcal{A} \models \sigma(C_i)$
  also $\mathcal{A} \models l$ für ein Literal l in $\sigma(C_i)$
  also I erfüllt $\sigma(C_i)$.
- Also erfüllt I gnd(S_φ).

Bemerkung zu Satz 2.4

Sei $\varphi \equiv \forall^*(C_1 \wedge \cdots \wedge C_n)$ eine Formel in CNF.
$S = \{C_1, \ldots, C_n\}$.

Es gilt

S ist erfüllbar gdw gnd(S) ist erfüllbar.

Beachte

gnd(S) ist eine (i. Allg. unendliche) Menge von Grundklauseln. Die Erfüllbarkeit von gnd(S) bezieht sich also auf die Aussagenlogik.

Also

Sei $\mathcal{A}x = \{\psi_1, \ldots, \psi_n\}$ und S eine Klauselmenge zu $\psi_1 \wedge \cdots \wedge \psi_n \wedge \neg\psi$. Dann gilt

$\mathcal{A}x \models \psi$ gdw gnd(S) ist aussagenlogisch unerfüllbar.

Beispiel

Sei $\mathcal{A}x = \{\forall x.p(x), \forall y.q(y)\} \quad \psi \equiv \forall x.(p(f(x)) \wedge q(x))$
Dann $S = \{p(x), q(y), \neg p(f(a)) \vee \neg q(a)\}$.
Also gnd$(S) = \{p(t_1), q(t_2) \mid t_1, t_2 \in \text{Term}(F)\}$
$\cup\{\neg p(f(a)) \vee \neg q(a)\}$,
also $p(f(a)), q(a), \neg p(f(a)) \vee \neg q(a) \in$ gnd(S).
Also gnd(S) ist unerfüllbar.

Hauptresultat für PL1 mit Gleichheit

Definition 2.9.

Eine **Herbrand-Algebra** $\mathcal{A}$ in PL1 mit Gleichheit ist eine Herbrand-Algebra in der $=^{\mathcal{A}}$ eine Kongruenzrelation ist (d. h., es gilt $\mathcal{A} \models \varphi$ für alle $\varphi \in \mathcal{A}x_=$). Die durch sie festgelegte Interpretation ist eine **Herbrand-Interpretation** (**H-Interpretation**).

Dann

Es bleiben alle Begriffe sinnvoll und die Überlegungen richtig.

Satz 2.5.

Eine Formel φ in CNF ist genau dann erfüllbar, wenn sie Herbrand-erfüllbar ist.

Also

Sei $\mathcal{A}x = \{\varphi_1, \ldots, \varphi_m\}$ und S eine Klauselmenge zu
$\varphi_1 \wedge \cdots \wedge \varphi_m \wedge \neg\varphi$. Es gilt
$\mathcal{A}x \models \varphi$ gdw keine Herbrand-Interpretation I erfüllt S.

Beachte

Eine H-Algebra $\mathcal{A}$ zu sig ist eine sig-Algebra im Sinn von PL1.
Es gilt dann $\mathcal{A} \models \varphi$ in PL1 $\quad\mathrm{gdw}\quad \mathcal{A}/=^{\mathcal{A}} \models \varphi$ in PL1=

Behandlung von Prädikatssymbolen (1)

Ziel

Der Superpositionskalkül, wie er später eingeführt wird, geht davon aus, dass $=$ das einzige Prädikatssymbol ist. Wir zeigen, dass dies keine wesentliche Beschränkung ist.

Transformation

Gegeben: Signatur sig $= (S, F, P \cup \{=\}, \alpha)$ und Klauselmenge S über sig.

Gesucht: Signatur sig$_0 = (S_0, F_0, \{=\}, \alpha_0)$ und Klauselmenge S_0 über sig$_0$ mit S ist H-erfüllbar gdw S_0 ist H-erfüllbar.

Setze $S_0 = S \cup \{\mathsf{BOOL}_0\}$, BOOL_0 eine neue Sorte
$F_0 = F \cup P \cup \{\top\}$
$\alpha_0(f) = \alpha(f)$ für $f \in F$
$\alpha_0(p) = s_1 \dots s_n \to \mathsf{BOOL}_0$, falls $\alpha(p) = s_1 \dots s_n$
$\alpha_0(\top) = \to \mathsf{BOOL}_0$.
Die Klauselmenge S_0 entsteht aus S dadurch, dass jedes Atom A durch $A = \top$ ersetzt wird.

Beachte

Ein H-Modell zu sig hat als Universum die Menge Term$(F, \emptyset)$ der Grundterme. Es ist eindeutig festgelegt durch eine Menge I von Grundatomen (den Atomen, die im Modell wahr sind). Das Modell muss die Gleichheitsaxiome erfüllen.

Behandlung von Prädikatssymbolen (2)

Satz 2.6.

Sei (sig, S) und (sig$_0$, S_0) wie eben. Es gilt S ist H-erfüllbar über sig gdw S_0 ist H-erfüllbar über sig$_0$.

Beweis:

a) Sei I ein H-Modell zu sig, das S erfüllt. Wir konstruieren ein H-Modell I_0 zu sig$_0$, das S_0 erfüllt: Setze
$I_0 = =_E$ mit $E = E' \cup E''$
$E' = \{t_1 = t_2 \mid I \models t_1 = t_2,\ t_1 = t_2$ Grundatom zu sig$\}$
$E'' = \{A = \top \mid I \models A,\ A$ Grundatom zu sig$\}$
Es gilt
(1) $A =_E \top$ gdw $I \models A$ (Übung)
(2) I_0 ist ein H-Modell zu sig$_0$
(3) I_0 erfüllt S_0, wegen (1) und I erfüllt S.

b) Sei I_0 ein H-Modell zu sig$_0$, das S_0 erfüllt. Wir konstruieren ein H-Modell zu sig, das S erfüllt: Setze
$I = M \cup =_E$ mit
$M = \{A \mid I_0 \models A = \top,\ A$ Grundatom zu sig$\}$
$E = \{t_1 = t_2 \mid I_0 \models t_1 = t_2,\ t_1 = t_2$ Grundatom zu sig$\}$
Es gilt
(1) I ist ein H-Modell zu sig, da I_0 ein H-Modell zu sig$_0$ ist.
(2) I erfüllt S, da I_0 S_0 erfüllt.

Bemerkung zur Herbrand-Erfüllbarkeit

Betrachte PL1:

Sei $S = \{C_1, \ldots, C_n\}$
gnd$(S) = \{\sigma(C) \mid C \in S, \sigma$ Grundsubstitution$\}$

Es gilt

S ist H-erfüllbar gdw gnd(S) ist aussagenlogisch erfüllbar.

Satz 2.7. (Kompaktheitssatz der Aussagenlogik)

Eine unendliche Menge M von aussagenlogischen Formeln ist genau dann unerfüllbar, wenn eine endliche Teilmenge $M_0 \subseteq M$ unerfüllbar ist.

ohne Beweis.

Folgerung

Das folgende Problem ist rekursiv aufzählbar.

Eingabe: S, eine endliche Klauselmenge über endlicher Signatur.
Frage: Ist S unerfüllbar?

Beweis: Man zähle die endlichen Teilmengen S_0 von gnd(S) auf und teste jeweils auf Erfüllbarkeit.
Antworte JA, falls eine Menge $S_0 \subseteq$ gnd(S) unerfüllbar ist.

Das Entscheidungsproblem der Prädikatenlogik

Satz 2.8.

Das folgende Problem ist rekursiv aufzählbar, aber nicht rekursiv.

Eingabe: $\mathcal{A}x, \varphi$ über endlicher Signatur.
Frage: Gilt $\mathcal{A}x \models \varphi$?

Beweis-Idee:

Aus der letzten Folgerung ergibt sich, dass das Problem rekursiv aufzählbar ist.
Man kann das Post'sche Korrespondenzproblem PCP auf dieses Problem reduzieren. Also ist es nicht rekursiv.

Ziel der Vorlesung

Ein effizientes Verfahren, das JA meldet, wenn eine Klauselmenge S unerfüllbar ist.

Beachte

Das eben skizzierte Verfahren ist hochgradig ineffizient, wenn das Herbrand-Universum Term(F) unendlich ist. Sonst ist das Problem entscheidbar, da reduzierbar auf ein endliches Problem der Aussagenlogik.

3 Reine Gleichheitslogik

3.1 Der generelle Ansatz

Problemstellung in Kapitel 3

Gegeben: $\mathcal{A}x = \{\varphi_1, \ldots, \varphi_n\}$ und φ mit
$\varphi_i \equiv \forall^* s_i = t_i$ und $\varphi \equiv Q^* u = v$

Frage: Gilt $\mathcal{A}x \models \varphi$?

Dabei ist Q^* eine Folge von $\forall$, $\exists$-Quantoren.

Transformation in Klauselform von $\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \neg\varphi$ liefert
$S = Ax \cup \{\neg \bar{u} = \bar{v}\}$ mit $Ax = \{s_i = t_i \mid i = 1, \ldots, n\}$

Es gilt

$\mathcal{A}x \models \varphi$

gdw es gibt keine H-Interpretation für S,

gdw jede H-Interpretation für Ax erfüllt nicht $\forall^* \bar{u} \neq \bar{v}$

gdw jede H-Interpretation für Ax erfüllt $\exists^* \bar{u} = \bar{v}$ (*)

Frage

Wie zeigt man (*)?

Die Beweistechnik

Beachte

Jede H-Interpretation für Ax ist eine Kongruenzrelation $\sim_{Ax}$ auf den Grundtermen, die Ax erfüllt.

Die gröbste dieser H-Interpretationen ist $=_{Ax}$. Dabei gilt $u =_{Ax} v$ genau dann, wenn sich u durch Anwenden von Gleichungen aus Ax in v überführen lässt.

Erfüllt $=_{Ax}$ die Formel $\exists^* \bar{u} = \bar{v}$, so gilt dies auch für alle $\sim_{Ax}$.

Definition 3.1.

Eine Ax-**Lösung** von $\bar{u} = \bar{v}$ ist eine Grundsubstitution σ mit $\sigma(\bar{u}) =_{Ax} \sigma(\bar{v})$.

Folgerung

Es gilt $\mathcal{A}x \models \varphi$ gdw $\bar{u} = \bar{v}$ ist Ax-lösbar.

Ziel

a) Aus Ax ein grundkonvergentes Regelsystem R erstellen.
b) Mit R die Ax-Lösbarkeit von $\bar{u} = \bar{v}$ testen.

Ein Beispiel

Beachte

Sei $\varphi \equiv \forall x_1, \ldots, x_k.u = v$ ein all-quantifiziertes Ziel.

Skolemisierung liefert $\bar{u} = \bar{v}$ mit
$\bar{u} \equiv \sigma(u), \bar{v} \equiv \sigma(v), \sigma = \{x_i \leftarrow a_i \mid i = 1, \ldots, k\}, a_i$ neu.

Also ist $\bar{u} = \bar{v}$ eine Grundgleichung (variablenfrei).

Es gilt: $\bar{u} = \bar{v}$ ist Ax-lösbar gdw $\bar{u} =_{Ax} \bar{v}$.

Beispiel

$\mathcal{A}x$: $\forall x.\ x + 0 = x$, $\forall x, y.\ x + s(y) = s(x + y)$, $s^3(0) = 0$.
φ: $\forall x.\ s^3(x) = x$.

Dies liefert
Ax: $x + 0 = x \quad x + s(y) = s(x + y) \quad s^3(0) = 0$.
$\bar{u} = \bar{v}$: $s^3(a) = a$.

Vervollständigung von Ax liefert

$R : x + 0 \to x \quad x + s(y) \to s(x + y) \quad s^3(x) \to x$

Also $s^3(a) \downarrow a$ und $\bar{u} =_{Ax} \bar{v}$.

Also gilt $\mathcal{A}x \models \varphi$.

Bemerkungen zu $\mathcal{A}x \models \varphi$

1. Bemerkung

Die Beweistechnik funktioniert immer, falls

1. $=$ das einzige Prädikatensymbol ist, also jedes Atom die Form $t_1 = t_2$ hat.
2. S nur Unit-Klauseln enthält, also jedes C in S nur aus einem Literal besteht.

2. Bemerkung

Sei sig $= (F, P, \alpha)$ und $\mathcal{A}x \cup \{\varphi\}$ enthalte nur sig-Formeln.

Dann sind die Terme in S F^e-Terme mit $F \subseteq F^e$, denn die Skolemisierung von $\neg\varphi$ führt neue Funktionssymbole ein.

Dann ist das Herbrand-Universum Term(F^e), nicht Term(F).

3. Bemerkung

Man benötigt häufig kein grund-konvergentes R zu $\mathcal{A}x$. (Es kann unendlich sein!) Es reicht, R hinreichend genau zu approximieren, so dass die Ax-Lösbarkeit von $\bar{u} = \bar{v}$ nachweisbar ist.

Reduzierte Problemstellung

Problemstellung ursprünglich

Gegeben: $\mathcal{A}x, \varphi$ über sig $= (F, P, \alpha)$ mit $P = \{=\}$ wie oben.
Frage: Gilt $\mathcal{A}x \models \varphi$?

Problemstellung reduziert

Gegeben: $Ax = \{s_i = t_i \mid i = 1, \ldots, n\}$ und $\bar{u} = \bar{v}$ mit $s_i, t_i, \bar{u}, \bar{v} \in \mathrm{Term}(F^e, V)$, $F \subseteq F^e$.
Frage: Ist $\bar{u} = \bar{v}$ Ax-lösbar?

Ab jetzt:

Nur noch reduzierte Problemstellung betrachtet.

Wir schreiben F statt F^e und $u = v$ statt $\bar{u} = \bar{v}$.

Beachte

Ist φ all-quantifiziert, d. h. $\varphi \equiv \forall^* u = v$, so ist $\bar{u} = \bar{v}$ eine Grundgleichung. Enthält φ auch den $\exists$-Quantor, so enthält $\bar{u} = \bar{v}$ auch Variablen.

3.2 All-quantifizierte Ziele

Problem

Gegeben: $Ax = \{s_i = t_i \mid i = 1, \ldots, n\}$ $\quad s_i, t_i \in \mathrm{Term}(F, V)$
$u = v$ $\quad u, v \in \mathrm{Term}(F)$
Frage: Wie zeigt man $u =_{Ax} v$?

Beachte

a) Sei $\longrightarrow$ grundkonvergent auf $\mathrm{Term}(F)$ und $=_{Ax} = \overset{*}{\longleftrightarrow}$ auf $\mathrm{Term}(F)$. Dann gilt $u =_{Ax} v$ gdw $u \downarrow v$.
b) Die Vervollständigung von Ax liefert ein konvergentes $\longrightarrow$, es reicht aber die Grundkonvergenz.
c) Die Vervollständigung kann divergieren oder abbrechen.
Wie vermeidet man dies?

Beispiel

Sei $F = \{0, s, +\}$ und

$$\begin{aligned} R: x + 0 &\to x \\ x + s(y) &\to s(x + y) \\ (x + y) + z &\to x + (y + z) \end{aligned}$$

Dann ist R grundkonfluent, aber nicht konfluent.

Bezeichnungen

$s = t$ ist eine **Gleichung**.

$s \rightarrow t$ ist eine **Regel**.

$s \doteq t$ steht für $s = t$ oder $t = s$.

$t[u]_\pi$ steht für $t[\pi \leftarrow u]$ mit $\pi \in \mathrm{Pos}(t)$.

$\longrightarrow$ ist eine **Reduktionsrelation** auf Term(F, V), falls $u \longrightarrow v$ impliziert $\sigma(u) \longrightarrow \sigma(v)$ und $t[u]_\pi \longrightarrow t[v]_\pi$.

Eine Ordnung $\succ$ auf Term(F, V) ist eine **Reduktionsordnung**, falls sie eine Noethersche Reduktionsrelation ist.

Sie ist eine **Grund-Reduktionsordnung**, falls sie total auf Term(F) ist.

$>_{TT}$ ist die **Teiltermordnung**.

$s \trianglerighteq t$ gdw es gibt σ mit $s >_{TT} \sigma(t)$ oder $s \equiv \sigma(t)$, aber s und t sind nicht gleich bis auf Variablenumbenennung.

Beachte

Jede LPO und jede KBO mit totaler Präzedenz ist eine Grund-Reduktionsordnung.

Behandlung von unrichtbaren Gleichungen

Sei $\succ$ eine Reduktionsordnung.
R ein Regelsystem mit $R \subseteq \succ$.
E ein Gleichungssystem.

Intention

Richtbare Instanzen von Gleichungen $u = v$ in E kann man zum Simplifizieren benutzen.

Gleichungen können auf Term(F) zu lokalen Divergenzen führen. Diese müssen durch kritische Paare überdeckt werden.

Bezeichnung

$E^\succ = \{\sigma(u) \rightarrow \sigma(v) \mid u \doteq v \text{ in } E, \sigma(u) \succ \sigma(v)\}$

$R(E) = R \cup E^\succ$

Definition 3.2.

Seien $u = v$ und $l = r$ zwei Gleichungen. Sei $\pi \in \mathrm{Pos}(l)$, $l/\pi \notin V$ und $\sigma = \mathrm{mgu}(u, l/\pi)$. Dann ist

$$\langle \sigma(r), \sigma(l)[\sigma(v)]_\pi \rangle$$

ein **kritisches Paar**, falls $\sigma(r) \not\succ \sigma(l) \not\prec \sigma(l)[\sigma(v)]_\pi$.

Sei CP(R, E) die Menge aller kritischen Paare zu Gleichungen aus $R \cup E$.

Bemerkungen zu kritischen Paaren

Sei $\longrightarrow\ =\ \longrightarrow_{R(E)}$.

1. Bemerkung

Sei $t_1 \longleftarrow t \longrightarrow t_2$ mit $t, t_1, t_2 \in$ Term(F) eine echte Überlappung

und $t \longrightarrow t_1$ mit $l = r$, τ bei $\pi' = \lambda$, $t/\lambda \equiv \tau(l) \succ \tau(r)$
$t \longrightarrow t_2$ mit $u = v$, τ bei $\pi' = \pi$, $t/\pi \equiv \tau(u) \succ \tau(v)$

Dann ist $\langle t_1, t_2 \rangle$ eine Instanz des kritischen Paares $\langle \sigma(r), \sigma(l)[\sigma(v)]_\pi \rangle$ zu $u = v$ und $l = r$. Die kritischen Paare überdecken also die echten Überlappungen auf Term(F).

2. Bemerkung

Es gilt $s \not\succ t$ gdw ($t \succ s$ oder $s \equiv t$ oder
s, t sind $\succ$-unvergleichbar).

3. Bemerkung (zu den kritischen Paaren)

Ist $l \succ r$, so ist $\sigma(l) \succ \sigma(r)$, also $\sigma(r) \not\succ \sigma(l)$.

Ist $u \succ v$, so ist $\sigma(l) \equiv \sigma(l)[\sigma(u)]_\pi \succ \sigma(l)[\sigma(v)]_\pi$.

Also: Ist $E = \emptyset$, so stimmt CP(R, E) mit der klassischen Menge CP(R) überein.

Das Inferenzsystem $\mathcal{U}$: Motivation

Idee

Das Inferenzsystem für die KB-Vervollständigung so modifizieren, dass sie nicht abbricht.

Dies liefert UKB (unfailing KB-completion).

Dazu bei Eingabe Ax:

Die richtbaren Gleichungen in R speichern.

Die unrichtbaren Gleichungen in E speichern.

Simplifizieren mit $R(E) = R \cup E^{\succ}$.

Kritische Paare mit R und E bilden: CP(R, E).

Beachte

Dies vermeidet den Abbruch der Vervollständigung, nicht aber die Divergenz.

Beweistechnik für die Vollständigkeit und Korrektheit: Ansatz der Beweisverkleinerung (auf Grundbeweisen).

Das Inferenzsystem $\mathcal{U}$: Definition

Definition 3.3.

Sei $\succ$ eine Grund-Reduktionsordnung.

$\mathcal{U} = \mathcal{U}_\succ$ besteht aus folgenden Regeln:

1. Richten einer Gleichung
   $(R, E \cup \{s \doteq t\}) \vdash (R \cup \{s \to t\}, E)$
   falls $s \succ t$

2. Erzeugen einer neuen Gleichung
   $(R, E) \vdash (R, E \cup \{s = t\})$
   falls $s \longleftrightarrow_{R \cup E} u \longleftrightarrow_{R \cup E} t$ und $s \not\succ u$ und $t \not\succ u$

3. Simplifizieren einer Gleichung
   $(R, E \cup \{s = t\}) \vdash (R, E \cup \{u = t\})$
   falls
   - $s \longrightarrow_R u$,
   - $s \longrightarrow_{E^\succ} u$ mit $l = r \in E$ und $s \rhd l$ oder
   - $s \longrightarrow_{R(E)} u$ und $t \succ s$

Das Inferenzsystem $\mathcal{U}$: Definition (Forts.)

4. Löschen einer trivialen Gleichung
   $(R, E \cup \{s = s\}) \vdash (R, E)$

5. Subsumieren einer Gleichung
   $(R, E \cup \{s = t\}) \vdash (R, E)$
   falls $s \vdash\!\dashv_E t$

6. Simplifizieren einer Regel rechts
   $(R \cup \{s \to t\}, E) \vdash (R \cup \{s \to u\}, E)$
   falls $t \longrightarrow_{R(E)} u$

7. Simplifizieren einer Regel links
   $(R \cup \{s \to t\}, E) \vdash (R, E \cup \{u = t\})$
   falls $s \longrightarrow_{R(E)} u$ mit $l \to r$ und $s \rhd l$

Bemerkungen zu $\mathcal{U}$

Bemerkung

Regel (2) wird nur angewendet mit $\langle s, t\rangle \in \mathrm{CP}(R, E)$.

Eine $\mathcal{U}$-**Ableitung** (für E) ist eine Folge $(R_i, E_i)_{i\in\mathbb{N}}$ mit $(R_i, E_i) \vdash (R_{i+1}, E_{i+1})$ und $(R_0, E_0) = (\varnothing, E)$.

Lemma 3.1

Sei $(R_i, E_i)_{i\in\mathbb{N}}$ eine $\mathcal{U}$-Ableitung für E. Dann gilt

a) $R_i \subseteq \succ$.

b) $=_E \, = \, =_{R_i \cup E_i}$ auf $\mathrm{Term}(F, V)$.

Ziel (+)

Sei $\longrightarrow_i = \longrightarrow_{R_i(E_i)}$.

Für eine „faire“ $\mathcal{U}$-Ableitung $(R_i, E_i)_{i\in\mathbb{N}}$ gilt

$u =_E v \text{ gdw } \exists i \in \mathbb{N} : u \downarrow_i v$

Bemerkung

Dies gilt. Beweis hierzu später.

Zunächst: Verfeinerung des Ansatzes.

Beispiel

Problem

Ax: $x + y = y + x \qquad (x + y) + z = x + (y + z)$

Ziel: Zeige $(a + b) + (a + c) =_{Ax} a + (a + (b + c))$

Sei

$t_1 + t_2 + \cdots + t_n \equiv t_1 + (t_2 + \cdots + (t_{n-1} + t_n) \ldots)$

$\succ$ die LPO zur Präzedenz $+ >_F a >_F b >_F c$.

Dann

$\mathcal{U}$ divergiert mit Eingabe Ax, und es wird für fast jede Permutation $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ die folgende Gleichung erzeugt

$$x_1 + x_2 + \cdots + x_n = x_{\pi(1)} + x_{\pi(2)} + \cdots + x_{\pi(n)}$$

Das sind sehr viele Gleichungen. Wir zeigen, dass sie fast alle redundant sind.

Beispiel (Forts.)

Anfang der Vervollständigung mit $\mathcal{U}$ liefert

$R : (x + y) + z \to x + (y + z)$
$E : x + y = y + x$
$\quad x + (y + z) = y + (x + z)$

Dies System ist bereits grundkonvergent auf Term(F).

Es gilt z. B. mit $\longrightarrow\; = \;\longrightarrow_{R(E)}$

$$
\begin{array}{lll}
(a + b) + (a + c) & \longrightarrow a+(b+(a+c)) \longrightarrow b+(a+(a+c)) \\
& \longrightarrow b+(a+(c+a)) \longrightarrow b+(c+(a+a)) \\
& \longrightarrow c + (b + (a + a)) \\
a + (a + (b + c)) & \stackrel{*}{\longrightarrow} c + (b + (a + a))
\end{array}
$$

Also gilt $(a + b) + (a + c) =_{Ax} a + (a + (b + c))$.

Redundanz-Elimination

Bemerkung

Das Inferenzsystem $\mathcal{U}$ erzeugt viele „unnötige“ Gleichungen. Sie verursachen einerseits Arbeit, helfen aber andererseits beim Simplifizieren (Säubern der Datenbasis).

Fragen

1. Was heißt „unnötig“?
2. Wie erkennt man unnötige Gleichungen?
3. Kann man sie zum Simplifizieren verwenden, ohne mit ihnen kritische Paare zu bilden?

Ansatz

Beweistechnik für das Ziel $(+)$: Verkleinerung von Grundbeweisen.

Also: $s = t$ heißt **unnötig** in (R, E), falls es für jede Grundinstanz $\sigma(s) = \sigma(t)$ in (R, E) einen kleineren Beweis gibt als den 1-Schritt-Beweis mit $s = t$.

Ziel

Antwort JA auf Frage 3.

Eine Beweisordnung (1)

Sei $\succ$ eine Grundreduktionsordnung.

Bemerkung

Wir arbeiten mit Tripeln (R, E, G)

R enthält die gerichteten Gleichungen, also $R \subseteq \succ$.
E enthält die ungerichteten Gleichungen.
G enthält die redundanten Gleichungen.

Definition 3.4.

Ein **Grundbeweis** zu $s = t$ in (R, E, G) hat die Form

$$P = (t_0, \varrho_1, t_1, \ldots, \varrho_n, t_n)$$

mit $t_i \in$ Term(F), $t_0 \equiv s$, $t_n \equiv t$ und $t_{i-1} \varrho_i t_i$
mit
$\varrho_i \in \{\longrightarrow_R, \longleftarrow_R, \longrightarrow_{E^\succ}, \longleftarrow_{E^\succ}, \vdash\!\dashv_E, \longrightarrow_{G^\succ}, \longleftarrow_{G^\succ}, \vdash\!\dashv_G\}$.

Beachte

$\succ$ ist total auf Term(F).
Also gilt für alle $t_i, t_{i+1} \in$ Term(F):
$t_i \succ t_{i+1}$, $t_i \equiv t_{i+1}$ oder $t_{i+1} \succ t_i$.

Eine Beweisordnung (2)

Definition 3.5.

Die Komplexität eines Beweisschritts $s \varrho t$ ist

$$c(s\varrho t) = \begin{cases} (\{s\}, l, \{t\}) & \text{falls } s \longrightarrow_R t \text{ mit } l \rightarrow r \\ (\{t\}, l, \{s\}) & \text{falls } s \longleftarrow_R t \text{ mit } l \rightarrow r \\ (\{s\}, l, \{t, \max\}) & \text{falls } s \longrightarrow_{(E\cup G)^\succ} t \text{ mit } l = r \\ (\{t\}, l, \{s, \max\}) & \text{falls } s \longleftarrow_{(E\cup G)^\succ} t \text{ mit } l = r \\ (\{s, t\}, -, -) & \text{falls } s \vdash\!\dashv_{E\cup G} t \end{cases}$$

Dabei ist $\max$ ein neues Symbol mit $\max \succ t$ für alle $t \in$ Term(F).

Die Ordnung $\succ_t$ auf den Tripeln $c(s\varrho t)$ ist die lexikographische Kombination $\succ_t = (\succ\!\!\succ, \rhd, \succ\!\!\succ)$. Hier ist $\succ\!\!\succ$ die Multimengenerweiterung der festen Grundreduktionsordnung $\succ$.

Die Komplexität eines Beweises $P = (t_0, \varrho_1, t_1, \ldots, \varrho_n, t_n)$ ist

$$c(P) = \{c(t_{i-1}\varrho_i t_i) \mid i = 1, \ldots, n\}$$

Die Beweisordnung $\succ_{\mathcal{P}}$ ist

$$P \succ_{\mathcal{P}} P' \text{ gdw } c(P) \succ\!\!\succ_t c(P')$$

Präzisierung der Redundanz

Definition 3.6.

$s = t$ heißt in (R, E) **redundant**, falls für jede Grundsubstitution σ gilt:

i) $\sigma(s) \equiv \sigma(t)$ oder
ii) es gibt Beweis P zu $\sigma(s) = \sigma(t)$ in (R, E) mit
$\sigma(s) \longrightarrow_{\{s=t\}^{\succ}} \sigma(t) \succ_{\mathcal{P}} P$ oder
$\sigma(t) \longrightarrow_{\{s=t\}^{\succ}} \sigma(s) \succ_{\mathcal{P}} P$

Schreibweise $s = t \succ_{\mathcal{P}} R(E)$.

Lemma 3.2

Sei $s = t \succ_{\mathcal{P}} R(E \cup \{s = t\})$.
Dann gibt es zu jedem Grundbeweis $\sigma(s)\varrho\sigma(t)$ einen echt kleineren Grundbeweis ohne Schritte mit $s = t$.
Insbesondere folgt $s = t \succ_{\mathcal{P}} R(E)$.

Also

Sei $E' = E \cup \{s = t\}$. Aus $s = t \succ_{\mathcal{P}} R(E')$ folgt $=_{R\cup E'} \; = \; =_{R\cup E}$ auf den Grundtermen $\mathrm{Term}(F)$.

Die Elimination einer redundanten Gleichung ist semantisch korrekt (ändert nicht die Gleichheit auf Grundtermen).

Beweis zu Lemma 3.2

Sei $E' = E \cup \{s = t\}$, $\sigma \in \mathrm{GSub}(s = t)$.

Nach Voraussetzung gibt es einen Beweis P in (R, E') mit $\sigma(s)\varrho\sigma(t) \succ_{\mathcal{P}} P$.

Wir zeigen durch Induktion nach $\succ_{\mathcal{P}}$, dass es solch einen Beweis in (R, E) gibt:

Sei $\sigma'(s)\varrho'\sigma'(t)$ ein Beweisschritt in P.

Also $\sigma(s)\varrho\sigma(t) \succ_{\mathcal{P}} P \succeq_{\mathcal{P}} \sigma'(s)\varrho'\sigma'(t)$.

Also gibt es (nach IV) P' in (R, E) zu $\sigma'(s)\varrho'\sigma'(t)$ mit $\sigma'(s)\varrho'\sigma'(t) \succ_{\mathcal{P}} P'$.

Ersetze alle $\sigma'(s)\varrho'\sigma'(t)$ in P durch die P'.

Es entsteht P_0 zu $\sigma(s)\varrho\sigma(t)$ in (R, E) mit

$$\sigma(s)\varrho\sigma(t) \succ_{\mathcal{P}} P \succeq_{\mathcal{P}} P_0 \,.$$

Nach Definition 3.6 folgt $s = t \succ_{\mathcal{P}} R(E)$.

Das Inferenzsystem $\mathcal{G}$

Definition 3.7.

Das Inferenzsystem $\mathcal{G}$ besteht aus folgenden Regeln.

1. Richten einer Gleichung
   $(R, E \cup \{s \doteq t\}, G) \vdash (R \cup \{s \to t\}, E, G)$
   falls $s \succ t$

2. Erzeugen einer neuen Gleichung
   $(R, E, G) \vdash (R, E \cup \{s = t\}, G)$
   falls $s \longleftrightarrow_{R \cup E} u \longleftrightarrow_{R \cup E} t$ und $s \not\succ u$ und $t \not\succ u$

3. Simplifizieren einer Gleichung
   $(R, E \cup \{s \doteq t\}, G) \vdash (R, E \cup \{u = t\}, G)$
   falls
   - $s \longrightarrow_R u$ oder
   - $s \longrightarrow_{(E \cup G)^\succ} u$ mit $l = r \in E \cup G$ und $s \rhd l$ oder
   - $s \longrightarrow_{R(E \cup G)} u$ und $t \succ s$

4. Löschen einer trivialen Gleichung
   $(R, E \cup \{s = s\}, G) \vdash (R, E, G)$

5. Subsumieren einer Gleichung
   $(R, E \cup \{s = t\}, G) \vdash (R, E, G)$
   falls $s \vdash\!\dashv_{E \cup G} t$

Das Inferenzsystem $\mathcal{G}$ (Forts.)

6. Verschieben einer redundanten Gleichung von E nach G
   $(R, E \cup \{s = t\}, G) \vdash (R, E, G \cup \{s = t\})$
   falls $s = t \succ_{\mathcal{P}} R(E \cup G)$

7. Simplifizieren einer Regel rechts
   $(R \cup \{s \to t\}, E, G) \vdash (R \cup \{s \to u\}, E, G)$
   falls $t \longrightarrow_{R(E \cup G)} u$

8. Simplifizieren einer Regel links
   $(R \cup \{s \to t\}, E, G) \vdash (R, E \cup \{u = t\}, G)$
   falls $s \longrightarrow_{R(E \cup G)} u$ mit $l = r$ und $s \rhd l$

Bezeichnung

Wir schreiben $(R, E, G) \vdash (R', E', G')$, falls (R', E', G') mit $\mathcal{G}$ aus (R, E, G) in einem Schritt hergeleitet werden kann.

Eine $\mathcal{G}$-**Ableitung** ist eine Folge $(R_i, E_i, G_i)_{i \in \mathbb{N}}$ mit $(R_i, E_i, G_i) \vdash (R_{i+1}, E_{i+1}, G_{i+1})$ für alle $i \geq 0$. Es ist eine $\mathcal{G}$-**Ableitung für** E, falls $(R_0, E_0, G_0) = (\emptyset, E, \emptyset)$.

Ergebnisse zu $\mathcal{G}$ (1)

Sei $(R_i, E_i, G_i)_{i\in\mathbb{N}}$ eine $\mathcal{G}$-Ableitung zur Eingabe E.

Der Korrektheits- und Vollständigkeitsbeweis zu $\mathcal{G}$ läuft parallel zu dem für die normale KB-Vervollständigung.

$\mathcal{G}$ ist korrekt:

Lemma 3.3

a) $R_i \subseteq \succ$

b) $=_E \; = \; =_{R_i \cup E_i \cup G_i}$ auf Term(F, V).

$\mathcal{G}$ verkleinert Grundbeweise schwach:

Lemma 3.4

Zu jedem Grundbeweis P zu $u = v$ in (R_i, E_i, G_i) gibt es einen Grundbeweis P' zu $u = v$ in $(R_{i+1}, E_{i+1}, G_{i+1})$ mit $P \succeq_{\mathcal{P}} P'$.

Die Gleichungen in G_i sind redundant:

Lemma 3.5

Für alle $s = t$ in G_i gilt $s = t \succ_{\mathcal{P}} R_i(E_i)$.

Beweis zu Lemma 3.5

Sei $\mathbf{P}$ die Menge aller Grundbeweise in (R_i, E_i, G_i). Mit Induktion nach $\succ_{\mathcal{P}}$ zeigen wir:

> Für jeden Grundbeweis $P \in \mathbf{P}$ gibt es einen Grundbeweis P' in $(R_i, E_i, \emptyset)$ mit $P \succeq_{\mathcal{P}} P'$. Wird in P eine Gleichung aus G_i benutzt, gilt sogar $P \succ_{\mathcal{P}} P'$.

Wird in P keine Gleichung aus G_i benutzt, kann man $P' = P$ wählen.

Sei $\sigma(s)\varrho\sigma(t)$ ein Beweisschritt in P mit $s = t$ in G_i.

Es sei j der Zeitpunkt, zu dem $s = t$ (mit Regel 6) von E nach G verschoben wurde. Es gibt dann in (R_j, E_j, G_j) einen Grundbeweis P_0 für $\sigma(s) = \sigma(t)$, der kleiner als $\sigma(s)\varrho\sigma(t)$ ist.

Also $P \succeq_{\mathcal{P}} \sigma(s)\varrho\sigma(t) \succ_{\mathcal{P}} P_0$.

Also gibt es (nach Lemma 3.4) P_1 in (R_i, E_i, G_i) zu $\sigma(s)\varrho\sigma(t)$ mit $P_0 \succeq_{\mathcal{P}} P_1$.

Es ist $P \succ_{\mathcal{P}} P_1$, also gibt es zu P_1 nach IV einen Grundbeweis P_1', der keine G_i-Schritte enthält, mit $\sigma(s)\varrho\sigma(t) \succ_{\mathcal{P}} P_1 \succeq_{\mathcal{P}} P_1'$.

Ersetze so alle Beweisschritte der Art $\sigma(s)\varrho\sigma(t)$ in P durch solche P_1'.

Es entsteht P' zu P in $(R_i, E_i, \emptyset)$ mit $P \succeq_{\mathcal{P}} P'$. Wird mindestens ein G_i-Schritt ersetzt, gilt sogar $P \succ_{\mathcal{P}} P'$.

Daraus folgt die Behauptung.

Ergebnisse zu $\mathcal{G}$ (2)

Beachte

a) Gleichungen in G werden nie simplifiziert. Sie dienen aber zum Simplifizieren.
b) Lemma 3.5 ist das zentrale Resultat über die Redundanz. Es bricht den Zykel: $s = t$ in G simplifiziert die Regel $l \to r$, die benutzt wurde, um $s = t$ von E nach G zu verschieben.

Folgerung

Sei $u =_E v$ eine Grundgleichung. Es gibt Beweise $(P_i)_{i\in\mathbb{N}}$ zu $u = v$ in $(R_i, E_i, \varnothing)$ mit $P_i \succeq_{\mathcal{P}} P_{i+1}$.

Also gilt auch $=_E \; = \; =_{R_i \cup E_i}$ auf den Grundtermen.

Ziel

Unter einer Fairnessbedingung der $\mathcal{G}$-Ableitung verkleinert $\mathcal{G}$ die Grundbeweise echt.

Ergebnisse zu $\mathcal{G}$ (3)

Definition 3.8.

Sei $(R_i, E_i, G_i)_{i\in\mathbb{N}}$ eine $\mathcal{G}$-Ableitung. Dann ist

$$R^\infty = \bigcup_{i\geq 0} \bigcap_{j\geq i} R_j \qquad E^\infty = \bigcup_{i\geq 0} \bigcap_{j\geq i} E_j$$

die Menge der persistenten Regeln bzw. Gleichungen.

Die $\mathcal{G}$-Ableitung heißt **fair**, falls

$$\mathsf{CP}(R^\infty, E^\infty) \subseteq \bigcup_{i\geq 0} E_i$$

Also

Fairness bedeutet, dass im Grenzsystem (R^∞, E^∞) jedes bildbare kritische Paar auch gebildet worden ist.

Beachte

Die Forderung $E^\infty = \varnothing$ aus der KB-Vervollständigung ist hier i. Allg. unerfüllbar.

Satz 3.1.

Für jede faire $\mathcal{G}$-Ableitung gilt

a) $R^\infty(E^\infty)$ ist grundkonfluent und terminierend.
b) Für jede Grundgleichung $u = v$ gilt: $u =_E v \text{ gdw } u \downarrow v$ in $R_i(E_i)$ für ein i.

Ergebnisse zu $\mathcal{G}$

Zusammenfassung

Redundante Gleichungen

- kann man zum Simplifizieren benutzen,
- ohne sie für die Bildung kritischer Paare zu berücksichtigen.

Dies

- erhält die Simplifikationskraft und
- schränkt gleichzeitig den Suchraum ein.

Frage

Wie erkennt man redundante Gleichungen?

Ein Test auf Redundanz

Ansatz

Gilt $\sigma(s) \downarrow \sigma(t)$ in $R(E)$ für jede Grundsubstitution σ, so ist $s = t$ in (R, E) redundant.

Dies reicht nicht ganz! Es reicht nur für das Löschen von $s = t$, nicht aber für das Simplifizieren mit $s = t$: Der erste Beweisschritt in $\sigma(s) \downarrow \sigma(t)$ darf nicht zu groß sein.

Bezeichnung

$u \xrightarrow{s\rhd}_{R(E)} v \quad \mathrm{gdw}$

- $u \longrightarrow_R v$ mit $l \rightarrow r \in R$ und $s \rhd l$ oder
- $u \longrightarrow_{E\succ} v$ mit $l = r \in E$ und $s \rhd l$.

Definition 3.9.

$s \Downarrow_{\rhd} t$ in $R(E)$ gdw für jede Grundsubstitution σ gilt:

$\sigma(s) \equiv \sigma(t)$ oder

$\sigma(s) \succ \sigma(t)$ und $\sigma(s) \xrightarrow{s\rhd}_{R(E)} s_1 \downarrow \sigma(t)$ oder

$\sigma(t) \succ \sigma(s)$ und $\sigma(t) \xrightarrow{t\rhd}_{R(E)} t_1 \downarrow \sigma(s)$.

Lemma 3.6

Gilt $s \Downarrow_{\rhd} t$ in $R(E)$, so gilt auch $s = t \succ_{\mathcal{P}} R(E)$.

Redundanz bei AC-Gleichungen

Bemerkung

Der Redundanz-Test ist besonders dann hilfreich, wenn Ax AC-Axiome enthält.

Satz 3.2.

Sei $\succ$ eine LPO oder eine KBO mit totaler Präzedenz. Sei

R_0: $f(f(x,y),z) \to f(x, f(y,z))$

E_0: $f(x,y) = f(y,x)$
$f(x, f(y,z)) = f(y, f(x,z))$
$f(x, f(y,z)) = f(z, f(y,x))$
$f(x, f(y,z)) = f(y, f(z,x))$

Wenn $s = t$ nicht mit (R_0, E_0) simplifiziert oder subsumiert werden kann, so gilt: $s =_{R_0 \cup E_0} t$ impliziert $s \Downarrow_{\triangleright} t$ in $R_0(E_0)$.

Beispiel

Es gilt $f(f(t_3,t_0), f(t_2,t_1)) =_{R_0 \cup E_0} f(t_0, f(t_1, f(t_2,t_3)))$, also ist diese Gleichung in (R, E) redundant, falls $R_0 \subseteq R$ und $E_0 \subseteq E$.

Beachte

$R_0(E_0)$ ist auf den Grundtermen in Term(F) konfluent, falls $f \in F$ gilt.

Test auf AC-Gleichheit

Lemma 3.7

Seien $s, t \in$ Term(F, V), $\{x_1, \ldots, x_n\} =$ Var$(s = t)$,

$\sigma = \{x_i \leftarrow a_i \mid i = 1, \ldots, n\}$ mit $a_i \notin F$.

Sei $\succ_0$ eine LPO oder KBO mit totaler Präzedenz $>_F$ auf $\{a_1, \ldots, a_n\} \cup F$.

Es gilt

$s =_{R_0 \cup E_0} t \text{ gdw } \sigma(s) \downarrow \sigma(t)$ in $R_0(E_0)$ bezüglich $\succ_0$.

Also

Der Test auf AC-Gleichheit ist einfach.

Bemerkung

Durch Verschärfung der Beweisordnung $\succ_{\mathcal{P}}$ kann man Satz 3.2 dadurch verschärfen, dass man aud E_0 die letzten beiden Gleichungen streicht.

Beispiel Abelsche Gruppen

Sei

$$
\begin{array}{lll}
Ax\colon & (x+y)+z = x+(y+z) & x+0 = x \\
 & x+y = y+x & x+-x = 0
\end{array}
$$

Ziel: $a + m \cdot b + -a =_{Ax} m \cdot b \qquad m > 0$

Hier ist $m \cdot x \equiv x + \cdots + x$ (m Summanden).

Das Redundanzkriterium aus dem letzten Satz

a) stoppt die Bildung kritischer Paare mit den AC-Gleichungen allein, nachdem (R_0, E_0) erzeugt ist.

b) erkennt viele sonstige Gleichungen als redundant.

c) beschleunigt die Zeit für die Herleitung des Ziels erheblich: Sei $\succ$ die LPO zur Präzedenz $- \;>_F\; + \;>_F\; 0 \;>_F\; a \;>_F\; b$: Beschleunigung der Beweissuche von fast 2 Stunden auf weniger als 2 Sekunden (für $m = 7$).

3.3 Allgemeinere Ziele

Beachte

Wir betrachten in Kapitel 3 nur Formeln über sig $= (F, P, \alpha)$ mit $P = \{=\}$. Atome haben also die Form $s = t$.

In Abschnitt 3.2 wurden Beweisaufgaben betrachtet, die zu Klauselmengen S führen mit
(1) Jede Klausel C aus S besteht nur aus einem Literal.
(2) Es gibt nur ein negatives Literal.
(3) Das negative Literal ist ein Grundliteral.

Jetzt

Wir fordern nur noch (1) von S.

Typische Beweisaufgabe

Gegeben: $\mathcal{A}x = \{\varphi_1, \ldots, \varphi_n\}$ und φ mit
$\varphi_i \equiv \forall^* s_i = t_i$ und $\varphi \equiv Q^* C$ mit C eine Klausel.
Frage: Gilt $\mathcal{A}x \models \varphi$?

Beachte

Dies führt zu einer Klauselmenge S, die (1) erfüllt.

Beispiel/Bemerkung

Beispiel Sei

$\varphi \equiv \forall x.\exists y.(\neg A_1(x,y) \vee \neg A_2(x,y) \vee B_1(x,y) \vee B_2(x,y))$

Die Skolemisierung von $\neg\varphi$ liefert

$$\neg\forall x.\exists y.(\neg A_1(x,y) \vee \neg A_2(x,y) \vee B_1(x,y) \vee B_2(x,y))$$

$$\exists x.\forall y.\neg(\neg A_1(x,y) \vee \neg A_2(x,y) \vee B_1(x,y) \vee B_2(x,y))$$

$$\exists x.\forall y.(A_1(x,y) \wedge A_2(x,y) \wedge \neg B_1(x,y) \wedge \neg B_2(x,y))$$

$$\forall y.(A_1(a,y) \wedge A_2(a,y) \wedge \neg B_1(a,y) \wedge \neg B_2(a,y))$$

Dies liefert 4 Klauseln der Länge 1:

$A_1(a,y) \qquad A_2(a,y) \qquad \neg B_1(a,y) \qquad \neg B_2(a,y)$

Davon sind 2 Klauseln negativ.

Bemerkung

Enthält C kein positives Literal, so enthält S nur positive Unit-Klauseln. Dann ist S immer H-erfüllbar. Also gilt dann nicht $\mathcal{A}x \models Q^*C$.

H-Erfüllbarkeit von Unit-Klausel-Mengen

Bezeichnung

C heißt eine **Unit-Klausel**, falls C nur ein Literal enthält.

Problemstellung jetzt

Gegeben: S, eine Menge von Unit-Klauseln.

Frage: Ist S H-erfüllbar?

Lemma 3.8

Sei $E = \{s_1 = t_1, \ldots, s_n = t_n\}$ und $U = \{u_1 \neq v_1, \ldots, u_m \neq v_m\}$ und $S = E \cup U$. Es ist S genau dann H-unerfüllbar, wenn es ein i gibt, so dass $u_i = v_i$ E-lösbar ist.

Beweis: Übung. Siehe dazu die Vorüberlegung in Abschnitt 3.2.

Voraussetzung ab jetzt: $U \neq \emptyset$.

Berechnung von E-Lösungen: Narrowing

Problemstellung

Eingabe: R ein grund-konvergentes Regelsystem,
$s = t$ eine Gleichung.
Frage: Gibt es eine R-Lösung σ zu $s = t$?

Grundidee für Narrowing

1. Sei σ R-Lösung zu $s = t$, also $\sigma(s) =_R \sigma(t)$.
   Setze $\tau(x) :\equiv \sigma(x) \downarrow$. Dann gilt auch $\tau(s) =_R \tau(t)$.
   Also: Es reicht „irreduzible Lösungen“ τ zu suchen.
2. Sei $\succ$ Reduktionsordnung, total auf Term(F), $\longrightarrow_R \subseteq \succ$.
   Sei $\tau(s) =_R \tau(t)$, also $\tau(s) \downarrow \tau(t)$.
   Gilt $\tau(s) \equiv \tau(t)$, so sind s und t unifizierbar.
   Sonst sei o.B.d.A. $\tau(s) \succ \tau(t)$. Wegen $\tau(s) \downarrow_R \tau(t)$ ist $\tau(s)$ reduzierbar, etwa mit $l \to r$ aus R. Dann gibt es eine Überlappung von $l \to r$ auf $s \to t$ und das kritische Paar $\langle u \equiv \mu(s[r]_\pi), v \equiv \mu(t) \rangle$ mit $\mu = \text{mgu}(s/\pi, l)$ und $\tau = \varrho \circ \mu$. Dann ist ϱ eine R-Lösung zu $u = v$ mit $\{\tau(s), \tau(t)\} \succ\!\!\succ \{\varrho(u), \varrho(v)\}$.

Also

Narrowing verkleinert Grundlösungen.

Narrowing (1)

Problemstellung

Gegeben: R grund-konvergentes Regelsystem.
$Z = \{u_1 = v_1, \ldots, u_m = v_m\}$ Menge von Zielen.
Frage: Ist ein $u_i = v_i$ aus Z R-lösbar?

Ansatz

1. Ist ein Paar (u_i, v_i) unifizierbar, so ist Z lösbar.
2. Sonst wird jedes $u_i = v_i$ durch Narrowing-Schritte gemäss der Grundidee überdeckt. Dies erzeugt neue Ziele.
3. Man darf jedes Ziel mit R simplifizieren ohne die Lösbarkeit von Z zu verlieren.

Definition 3.10.

Ein **kritisches Ziel** zu R und Z ist ein kritisches Paar zur Überlappung von $l \to r$ aus R auf $u \to v$ mit $u \doteq v$ aus Z, $\mu = \text{mgu}(u/\pi, l)$ und $\mu(u) \not\succ \mu(v)$. Sei $\boldsymbol{CP_{\mathcal{N}}(R, u = v)}$ die Menge der kritischen Ziele zu R auf $u = v$.

Narrowing (2)

Das Inferenzsystem $\mathcal{N}$

Sei R fest. Sei $\mathcal{N}$ das folgende Inferenzsystem.

$$\frac{Z \cup \{u \doteq v\}}{Z \cup \{s = v\}} \quad \text{falls } u \longrightarrow_R s$$

$$\frac{Z \cup \{u = v\}}{\text{ERFOLG}} \quad \begin{array}{l}\text{falls } u \text{ und } v \text{ unifizierbar und}\\ u, v \text{ irreduzibel sind}\end{array}$$

$$\frac{Z \cup \{u = v\}}{Z \cup CP_{\mathcal{N}}(R, u = v)} \quad \begin{array}{l}\text{falls } u \text{ und } v \text{ nicht unifizierbar und}\\ u, v \text{ irreduzibel sind}\end{array}$$

Satz 3.3.

Es gilt $Z \overset{*}{\underset{\mathcal{N}}{\vdash}} \text{ERFOLG gdw}$ ein $u = v \in Z$ ist R-lösbar.

ohne Beweis.

Zur H-Erfüllbarkeit von S

Seien

E, Z Gleichungssysteme.

$S = E \cup \{s \neq t \mid s = t \text{ in } Z\}$.

Strategie für den H-Erfüllbarkeitstest

Parallel E zu R vervollständigen und den Narrowing-Algorithmus ablaufen lassen.

Also

Inferenzsystem $\mathcal{GN}$ arbeitet auf 4-Tupeln (E, R, G, Z) mit

E enthält die ungerichteten Gleichungen.

R enthält die gerichteten Gleichungen.

G enthält die redundanten Gleichungen.

Z enthält die Ziele.

$\mathcal{GN} = \mathcal{GN}_{\succ}$ arbeitet mit fester Grund-Reduktionsordnung.

Definition 3.11.

$CP_{\mathcal{N}}(R, E, s = t) = CP_{\mathcal{N}}(R, s = t) \cup CP_{\mathcal{N}}(E, s = t)$.

$CP_{\mathcal{N}}(E, s = t)$ ist die Menge der kritischen Paare $u = v$ zu $l \to r$ auf $s = t$ mit $l \doteq r$ in E, $\mu = \text{mgu}(s/\pi, l)$ und $u \not\succ \mu(s) \not\prec v$.

Das Inferenzsystem $\mathcal{GN}$

Definition 3.12.

Das Inferenzsystem $\mathcal{GN}_\succ$ ist gegeben durch

- die Inferenzregeln von $\mathcal{G}$ und
- die folgenden Narrowing-Regeln:

Simplifizieren eines Ziels

$$\frac{(E, R, G, Z \cup \{s \doteq t\})}{(E, R, G, Z \cup \{u = t\})} \quad \text{falls } s \longrightarrow_{R(E \cup G)} u$$

Subsumieren eines Ziels

$$\frac{(E, R, G, Z \cup \{s = t, u \doteq v\})}{(E, R, G, Z \cup \{s = t\})} \quad \begin{array}{l}\text{falls es } \sigma \text{ gibt mit}\\ u \equiv \sigma(s) \land v \equiv \sigma(t)\end{array}$$

Überdecken eines Ziels

$$\frac{(E, R, G, Z \cup \{s = t\})}{(E, R, G, Z \cup \{s = t, u = v\})} \quad \begin{array}{l}\text{falls } s, t \text{ nicht unifizierbar und}\\ u = v \in CP_{\mathcal{N}}(R, E, s = t)\end{array}$$

Erfolgsmeldung

$$\frac{(E, R, G, Z \cup \{s = t\})}{\text{ERFOLG}} \quad \text{falls } s, t \text{ unifizierbar}$$

Korrektheit von $\mathcal{GN}$

Bezeichnung

Z ist E-lösbar, falls ein $u = v$ aus Z E-lösbar ist.

Z ist (E, R)-lösbar, falls ein $u = v$ aus Z $(E \cup R)$-lösbar ist.

Sei

$(E_i, R_i, G_i, Z_i)_{i \geq 0}$ eine $\mathcal{GN}$-Ableitung.

Lemma 3.9

a) Z_i ist (E_i, R_i)-lösbar gdw Z_{i+1} ist (E_{i+1}, R_{i+1})-lösbar.

b) Gilt $(E, \emptyset, Z) \overset{*}{\underset{\mathcal{GN}}{\vdash}}$ ERFOLG, so ist Z E-lösbar.

Beweis:

a) Übung.

b) Dies folgt aus a).

Beachte

Sei $=_i \; = \; =_{E_i \cup R_i}$. Es gilt $=_i \; = \; =_{i+1}$ auf Term(F).

Also gilt für jede Gleichung $u = v$: Sie ist (E_i, R_i)-lösbar gdw sie ist (E_{i+1}, R_{i+1})-lösbar.

Also: „lösbar“ heißt „(E_0, R_0)-lösbar“.

Vollständigkeit von $\mathcal{GN}$ (1)

Definition 3.13. (Komplexität von Zielen)

Sei $\succ_Z = (\ggg, \rhd\!\!\rhd)$ die lexikographische Kombination von $\ggg$ und $\rhd\!\!\rhd$. Dabei ist $\rhd$ die Umschließungsordnung. Sei

$$K(s=t,\sigma) = \begin{cases} (\{\sigma(s),\sigma(t)\},\{s,t\}) & \sigma(s) =_{E_0 \cup R_0} \sigma(t) \\ (\{\max\},\{s,t\}) & \text{sonst} \end{cases}$$

$K(s=t) = \min_{\succ_Z}\{K(s=t,\sigma) \mid \sigma \in \mathsf{GSub}(s=t)\}$

$K(Z) = \min_{\succ_Z}\{K(s=t) \mid s=t \text{ in } Z\}$

Beachte

Alle Ordnungen sind Noethersch. Also existieren die Minima. $K(Z)$ ist die Komplexität der kleinsten Lösung einer Gleichung aus Z.

Definition 3.14. (Fairness)

Eine $\mathcal{GN}$-Ableitung heißt **fair**, falls gilt:

$CP(R^\infty, E^\infty) \subseteq \bigcup_{i\in\mathbb{N}} E_i$ (alle kritischen Paare sind gebildet).

$CP_{\mathcal{N}}(R^\infty, E^\infty, Z^\infty) \subseteq \bigcup_{i\in\mathbb{N}} Z_i$ (alle Ziele sind überdeckt).

Auf kein (E_i, R_i, G_i, Z_i) ist die ERFOLG-Regel anwendbar oder es gibt ein (E_j, R_j, G_j, Z_j), auf das die ERFOLG-Regel angewendet wird.

Vollständigkeit von $\mathcal{GN}$ (2)

Lemma 3.10

Sei $(E_i, R_i, G_i, Z_i)_{i\geq 0}$ eine $\mathcal{GN}$-Ableitung.

a) Es gilt $K(Z_{i-1}) \succeq_Z K(Z_i)$ für alle i der Folge.
b) Ist die Ableitung fair und ist Z_0 (E_0, R_0)-lösbar, so gilt für alle i der Folge entweder $(E_i, R_i, G_i, Z_i) \vdash$ ERFOLG oder es gibt ein $j > i$ mit $K(Z_i) \succ K(Z_j)$. Also bricht die Folge dann mit ERFOLG ab.

Beweis:

a) Man rechne dies nach.
b) Da Z_0 (E_0, R_0)-lösbar ist, ist auch Z_i (E_i, R_i)-lösbar. Sei $\sigma(s) = \sigma(t)$ mit $s = t$ aus Z_i die kleinste Lösung, die Z_i enthält. Nach Satz 3.1 gibt es ein k mit $\sigma(s) \downarrow \sigma(t)$ in (E_j, R_j) für alle $j \geq k$.
   Sind s, t unifizierbar, so gilt $(E_i, R_i, G_i, Z_i) \vdash$ ERFOLG. Sonst wird wegen der Fairness zu einem Zeitpunkt $j > i, k$ die Gleichung $s = t$ überdeckt und dabei ein kritisches Ziel $u = v$ erzeugt, das eine echt kleinere Lösung $\varrho(u) = \varrho(v)$ enthält als $\sigma(s) = \sigma(t)$. Dann gilt $K(Z_i) \succ_Z K(Z_j)$. Da $\succ_Z$ Noethersch ist, muss nach endlicher Zeit der Erfolgsfall eintreten. Dann bricht die Kette ab.

Ergebnis zu $\mathcal{GN}$

Satz 3.4.

$\mathcal{GN}_{\succ}$ ist korrekt und vollständig: Es gilt $(E, \emptyset, \emptyset, Z) \overset{*}{\vdash}$ ERFOLG gdw Z ist E-lösbar.
Jede faire $\mathcal{GN}$-Ableitung $(E_i, R_i, G_i, Z_i)_{i \geq 0}$ mit $(E_0, R_0, G_0, Z_0) = (E, \emptyset, \emptyset, Z)$ bricht genau dann mit ERFOLG ab, wenn Z E-lösbar ist.

Beweis:

Dies folgt aus Lemma 3.9 und Lemma 3.10.

Satz 3.5.

Sei $S = E \cup U$ die Klauselmenge zum Problem „$\mathcal{A}x \models Q^*C$?“ und sei $Z = \{u = v \mid u \neq v \text{ in } U\}$.

Dann gilt

$\mathcal{A}x \models Q^*C \;\mathrm{gdw}\; (E, \emptyset, \emptyset, Z) \overset{*}{\vdash}_{\mathcal{GN}}$ ERFOLG

Beweis:

Dies folgt aus Satz 3.4 und der Vorüberlegung.

Beispiel 1

$\mathcal{A}x : \forall x, y, z.[x \cdot 1 = x \wedge x \cdot x^{-1} = 1 \wedge (x \cdot y) \cdot z = x \cdot (y \cdot z)]$.
$\varphi \equiv \exists x \forall y.\ x \cdot y = y$.

Dies liefert $S = E \cup \{\neg x \cdot g(x) = g(x)\}$ mit
$E : x \cdot 1 = x \qquad x \cdot x^{-1} = 1 \qquad (x \cdot y) \cdot z = x \cdot (y \cdot z)$.

Es ist $Z = \{x \cdot g(x) = g(x)\}$.

Die Vervollständigung von E mit einer LPO liefert u.a. die Regeln

(1) $x \cdot 1 \to x$ (3) $(x \cdot y) \cdot z \to x \cdot (y \cdot z)$
(2) $x \cdot x^{-1} \to 1$ (4) $1 \cdot x \to x$

Überlappung der Regel (4) in das Ziel liefert

$$\begin{tikzcd} & 1 \cdot g(1) \arrow[dl] \arrow[dr] & \\ g(1) & & g(1) \end{tikzcd}$$

also das neue Ziel $g(1) = g(1)$. Unifikation liefert jetzt ERFOLG.

Also ist Z E-lösbar und es gilt $\mathcal{A}x \models \varphi$.

Beachte: Die berechnete E-Lösung zu Z ist $\sigma = \{x \leftarrow 1\}$.

Beispiel 2

$\mathcal{A}x : \forall x, y.[x + 0 = x \wedge x + s(y) = s(x + y) \wedge$
$\quad d(0) = 0 \wedge d(s(x)) = s(s(d(x)))].$

$\varphi : \forall x \exists y.\ d(x) = y + y.$

Dies liefert

$$\begin{aligned} E : x + 0 &= x & x + s(y) &= s(x + y) \\ d(0) &= 0 & d(s(x)) &= s^2(d(x)). \end{aligned}$$

$Z : d(a) = y + y.$

Sei σ eine beliebige Grundsubstitution. Dann enthält $\sigma(y + y)$ n Vorkommen von a mit $n = 0$ oder $n > 1$. Wegen der speziellen Gestalt der Gleichungen in E enthält auch jeder Term t mit $t =_E \sigma(y + y)$ genau n a's.

Also gilt nicht $\sigma(d(a)) =_E \sigma(y + y)$.

Also gilt nicht $\mathcal{A}x \models \varphi$.

Übung: Man teste, ob gilt

a) $\mathcal{A}x \models \forall x.[x = 0 \vee \exists y.x = s(y)]$.

b) $\mathcal{A}x \models \forall x, y.[d(x) = y \Rightarrow d(d(x)) = d(y))$.

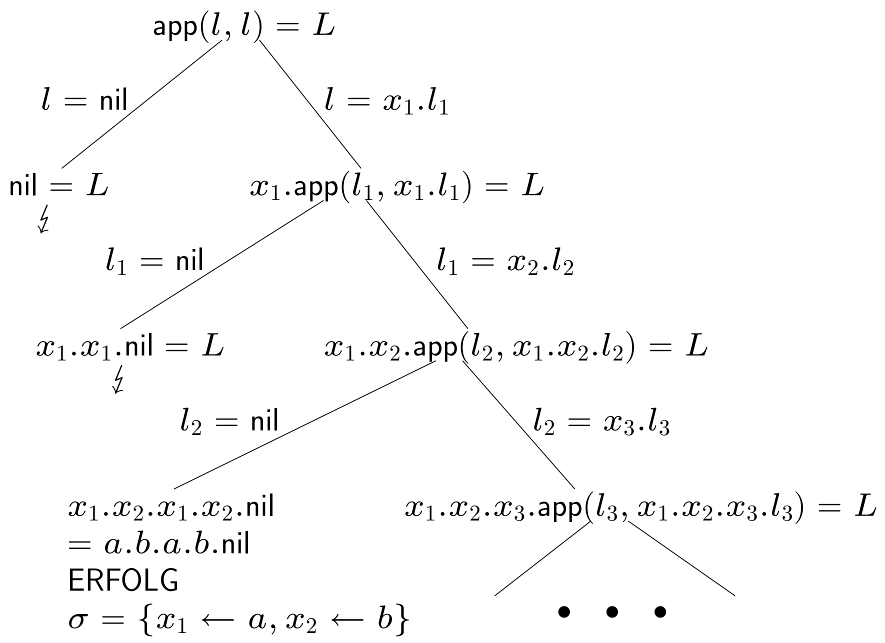
c) $\mathcal{A}x \models \forall x, y.[d(x) = y \Rightarrow d(y) = d(x) + d(x)]$.

Beispiel Narrowing

$$\begin{aligned} R :\ & \mathsf{app}(\mathsf{nil}, l_2) \rightarrow l_2 \\ & \mathsf{app}(x.l_1, l_2) \rightarrow x.\mathsf{app}(l_1, l_2) \end{aligned}$$

Gesucht: R-Lösungen zu $\mathsf{app}(l, l) = a.b.a.b.\mathsf{nil} \equiv L$

Narrowing: Suchbaum



$\rightsquigarrow$ $\sigma(l) = a.b.\mathsf{nil}$ ist eine R-Lösung zu $\mathsf{app}(l, l) = a.b.a.b.\mathsf{nil}$

Beispiel

$$\mathcal{A}x : \quad \begin{array}{l} \forall x.x \cdot 1 = x \\ \forall x.x \cdot x = x \\ \forall x.\forall y.\forall z.(x \cdot y) \cdot z = x \cdot (y \cdot z) \end{array} \qquad \varphi \equiv \exists x.\exists y.x \cdot y = y \cdot x$$

Frage: Gilt $\mathcal{A}x \models \varphi$?

Dies liefert

- $E : x \cdot 1 = x \quad x \cdot x = x \quad (x \cdot y) \cdot z = x \cdot (y \cdot z)$
- Ist $x' \cdot y' = y' \cdot x'$ E-lösbar?

Vervollständigung liefert

(1) $1 \cdot x \to x$ Also

$$\begin{tikzcd}
 & x' \cdot y' = y' \cdot x' \arrow[dl, "y' = 1"'] \arrow[dr, "x' = 1"] & \\
x' \cdot 1 = x' \arrow[d] & & y' = y' \cdot 1 \arrow[d] \\
x' = x' & & y' = y' \\
\rightsquigarrow \sigma = \{y' \leftarrow 1\} & & \rightsquigarrow \sigma = \{x' \leftarrow 1\}
\end{tikzcd}$$

(2) $x \cdot y = y \cdot x$ Also

$$\begin{tikzcd}
x' \cdot y' = y' \cdot x' \arrow[d] \\
x' \cdot y' = x' \cdot y' \\
\rightsquigarrow \sigma = \text{id}
\end{tikzcd}$$

Also gilt sogar $x' \cdot y' =_E y' \cdot x'$

also $\mathcal{A}x \models \forall x.\forall y.x \cdot y = y \cdot x$

4 Theorembeweisen in PL1

Ziel von Kapitel 4

Darstellung des Superpositionskalküls.

Beachte

Die heute schnellsten Beweiser beruhen auf diesem Kalkül.

Problemstellung

Gegeben: S, eine Menge von Klauseln.

Aufgabe: Zeige, dass S Herbrand-unerfüllbar ist.

Bemerkung

Der Superpositionskalkül baut auf folgenden Kalkülen auf

- Resolution,

- Paramodulation.

Wir führen diese Kalküle ein.

Beschreibung von Kalkülen

Wir schreiben Inferenzregeln jetzt etwas anders als in Kapitel 3.

Bezeichnung

Ein **Kalkül** $\mathcal{K}$ wird beschrieben durch ein System von Inferenzregeln, die auf Formelmengen S arbeiten.

Eine **Inferenzregel** hat die Form

$$\frac{C_1, \ldots, C_n}{D} \qquad \text{falls Anwendungsbedingung}$$

mit $n \geq 0$, wobei die C_i, D Formeln sind.

Wir schreiben $S \underset{\mathcal{K}}{\vdash} S'$, falls es solch eine Inferenzregel gibt mit $C_1, \ldots, C_n \in S$ und $S' = S \cup \{D\}$.

Eine $\mathcal{K}$-**Ableitung** ist eine Folge $(S_i)_{i \in \mathbb{N}}$ mit $S_i \underset{\mathcal{K}}{\vdash} S_{i+1}$ für alle $i \geq 0$. Sie kann endlich oder unendlich sein.

Wir schreiben $S \overset{*}{\underset{\mathcal{K}}{\vdash}} C$, falls es eine $\mathcal{K}$-Ableitung $(S_i)_{i \in \mathbb{N}}$ gibt mit $S = S_0$ und $C \in S_k$ für ein $k \geq 0$.

Hier

Die Formeln sind Klauseln, also ist S eine Klauselmenge.

Bemerkungen

Bemerkung

Wir schreiben $\mathcal{K}$-Ableitungen auch als Folge von Formeln. Eine $\mathcal{K}$-**Herleitung** aus S ist eine Folge $C'_1, \ldots, C'_m$, so dass für alle $1 \leq i \leq m$ gilt

a) $C'_i \in S$ oder

b) es gibt eine Inferenzregel $\dfrac{C_1, \ldots, C_n}{D}$, so dass die C_j unter den $C'_1, \ldots, C'_{i-1}$ vorkommen und $D \equiv C'_i$ ist.

Ist $C \equiv C'_m$, so ist dies eine $\mathcal{K}$-Herleitung von C aus S.

Schreibweise $S \overset{*}{\underset{\mathcal{K}}{\vdash}} C$.

Beachte

Uns interessieren Kalküle zum Nachweis der H-Unerfüllbarkeit von S. Die leere Klausel $\Box$ ist unerfüllbar. Ziel ist es also, $S \overset{*}{\underset{\mathcal{K}}{\vdash}} \Box$ zu zeigen.

Definition 4.1.

$\mathcal{K}$ heißt **korrekt**, falls aus $S \overset{*}{\underset{\mathcal{K}}{\vdash}} \Box$ folgt, dass S H-unerfüllbar ist.

$\mathcal{K}$ heißt **vollständig**, falls gilt: Ist S H-unerfüllbar, so gilt $S \overset{*}{\underset{\mathcal{K}}{\vdash}} \Box$.

Beispiel: Vervollständigungskalkül

Die Inferenzregel „Bildung kritisches Paar“ lautet

$$\frac{l = r \quad u = v}{\sigma(r) = \sigma(l[v]_\pi)} \quad \text{falls } \begin{array}{l} \pi \in \mathrm{Pos}(l), l/\pi \notin V, \\ \sigma = mgu(u, l/\pi), \\ \sigma(l) \not> \sigma(r) \not< \sigma(l[v]_\pi) \end{array}$$

Übung

1. Man beschreibe die KB-Vervollständigung als Kalkül $\mathcal{K}_{KB}$.
2. Sei $S = \{a = b, a = c, b = d\}$ und $>$ die RPO mit $a >_F b >_F c >_F d$. Man gebe eine $\mathcal{K}_{KB}$-Ableitung von $S \vdash c = d$ an.

Zur Korrektheit von $\mathcal{K}$

Definition 4.2.

Eine Inferenzregel $\frac{C_1, \ldots, C_n}{D}$ heißt **korrekt**, falls gilt: Ist $\{C_1, \ldots, C_n\}$ H-erfüllbar, so ist es auch $\{C_1, \ldots, C_n, D\}$.

Lemma 4.1

Ist jede Inferenzregel von $\mathcal{K}$ korrekt, so ist auch der Kalkül $\mathcal{K}$ korrekt.

Beweis:

Sei $S \overset{*}{\underset{\mathcal{K}}{\vdash}} \Box$, also $S_0, \ldots, S_n$ eine $\mathcal{K}$-Ableitung mit $S_0 = S$ und $\Box \in S_n$. Ist S_0 H-erfüllbar, so ist jedes S_i H-erfüllbar (Induktion nach i). Wäre S H-erfüllbar, so wäre auch S_n H-erfüllbar, wegen $\Box \in S_n$ also auch $\Box$. Das ist unmöglich.

Also ist S unerfüllbar.

Also ist $\mathcal{K}$ korrekt.

Beachte

Die Korrektheit von $\mathcal{K}$ ist häufig leicht zu zeigen. Schwierig ist der Nachweis der Vollständigkeit.

4.1 Resolution und Paramodulation

Resolution für PL1

Sei $=$ nicht in P, also die Gleichheit ausgeschlossen.
S eine Menge von Klauseln über sig $= (F, P, \alpha)$.
Seien A, B Atome, l_1, l_2 Literale, C, D Klauseln.

Definition 4.3.

Der **Resolutionskalkül** $\mathcal{R}$ besteht aus 2 Inferenzregeln:

Resolution

$$\frac{C \vee A \quad D \vee \neg B}{\sigma(C \vee D)} \qquad \text{falls } \sigma = \mathsf{mgu}(A, B)$$

Faktorisierung

$$\frac{C \vee l_1 \vee l_2}{\sigma(C \vee l_1)} \qquad \text{falls } \sigma = \mathsf{mgu}(l_1, l_2)$$

Ziel

Der Resolutionskalkül ist korrekt und vollständig:

$S \overset{*}{\underset{\mathcal{R}}{\vdash}} \Box \ \text{ gdw } \ S$ ist H-unerfüllbar.

Beispiel zur Resolution

Beispiel: Quacksalber

Sei $S = \{(1), \ldots, (5)\}$, siehe Einleitung.

(1) $\mathsf{pat}(a)$
(2) $\neg\mathsf{doc}(y) \vee \mathsf{mag}(a, y)$
(3) $\neg\mathsf{pat}(x') \vee \neg\mathsf{qua}(y') \vee \neg\mathsf{mag}(x', y')$
(4) $\mathsf{doc}(b)$
(5) $\mathsf{qua}(b)$

Anwendung der Resolutionsregel liefert

(6) $\neg\mathsf{qua}(y'') \vee \neg\mathsf{mag}(a, y'')$	mit (1), (3)
(7) $\neg\mathsf{mag}(a, b)$	mit (5), (6)
(8) $\mathsf{mag}(a, b)$	mit (2), (4)
(9) $\Box$	mit (7), (8)

Also ist S nicht H-erfüllbar.

Also gilt $\mathcal{A}x \models \forall x.(\mathsf{doc}(x) \Rightarrow \neg\mathsf{qua}(x))$ mit
$\mathcal{A}x = \{\exists x.(\mathsf{pat}(x) \wedge \forall y.(\mathsf{doc}(y) \Rightarrow \mathsf{mag}(x, y))),$
$\quad \forall x.(\mathsf{pat}(x) \Rightarrow \forall y.(\mathsf{qua}(y) \Rightarrow \neg\mathsf{mag}(x, y)))\}$

Literatur-Hinweis

J.A. Robinson: A machine-oriented logic based on the resolution principle, J. ACM 12 (1965) 23-41.

Korrektheit von $\mathcal{R}$

Erinnerung

Ein Herbrand-Modell über der Signatur sig $= (F, P, \alpha)$ ist festgelegt durch die Teilmenge $I \subseteq \{A \mid A \text{ ist Grundatom}\}$. Die Intention ist: $A \equiv p(t_1, \ldots, t_n)$ gilt in dem Modell genau dann, wenn $A \in I$ ist. Also gilt für jede Grundklausel C:

I ist Modell zu C gdw es gibt $A \in C$ mit $A \in I$ oder
$\neg A \in C$ mit $A \notin I$.

Lemma 4.2

Für die Regeln $\dfrac{C_1, \ldots, C_n}{D}$ $(n = 1, 2)$ des Resolutionskalküls gilt: Ist I ein Modell von $\{C_1, \ldots, C_n\}$, so ist I auch ein Modell von $\{C_1, \ldots, C_n, D\}$.

Beweis: Es hat D die Form $D \equiv \sigma(D_0)$. Es reicht für jede Grundsubstitution τ zu zeigen: Ist I ein Modell für alle $\tau(\sigma(C_i))$, so ist I auch ein Modell für $\tau(\sigma(D_0))$.

Übung: Man zeige dies für beide Inferenzregeln von $\mathcal{R}$.

Lemma 4.3

Der Resolutionskalkül $\mathcal{R}$ ist korrekt.

Beweis: Dies folgt aus Lemma 4.1 und 4.2.

Bemerkung zur Vollständigkeit von $\mathcal{R}$

Bemerkung

Der Vollständigkeitsbeweis für $\mathcal{R}$ ist viel schwieriger als der Korrektheitsbeweis.

Wir betrachten erst den Fall, dass S eine Menge von Grundklauseln ist und „liften" das Ergebnis auf den allgemeinen Fall.

Beachte

Für Grundklauseln reduziert sich $\mathcal{R}$ auf die Inferenzregeln

$$\frac{A \vee C \qquad \neg A \vee D}{C \vee D} \quad \text{Resolution}$$

$$\frac{A \vee A \vee C}{A \vee C} \quad \text{positive Faktorisierung}$$

$$\frac{\neg A \vee \neg A \vee C}{\neg A \vee C} \quad \text{negative Faktorisierung}$$

Weitere Beispiele

1. Beispiel

S bestehe aus den Klauseln (1) bis (4). Dies erzeugt (5) bis (7).

(1) $p \vee q$	(5) p
(2) $p \vee \neg q$	(6) $\neg p$
(3) $\neg p \vee r$	(7) $\square$
(4) $\neg p \vee \neg r$	

Also ist S H-unerfüllbar.

2. Beispiel

S bestehe aus

(1) $p \vee p$ (2) $\neg p \vee \neg p$

Abschluss unter der Resolutionsregel liefert nur

(3) $p \vee \neg p$

Mit der Faktorisierungsregel ergibt sich aus (1) und (2)

(4) p (5) $\neg p$ (6) $\square$

Also: Die Faktorisierungsregel wird benötigt für den Nachweis von $S \overset{*}{\underset{\mathcal{R}}{\vdash}} \square$.

Zur Vollständigkeit von $\mathcal{R}$ für Grundklauseln

Bezeichnung

S heißt **abgeschlossen unter** $\mathcal{R}$, falls für jede Inferenzregel gilt: Sind die Prämissen in S, so ist auch die Konklusion in S.

Beweistechnik zum Vollständigkeitsbeweis

Wir zeigen bei Eingabe S:

a) Jede „faire" $\mathcal{R}$-Ableitung liefert ein Grenzsystem S^*, das unter $\mathcal{R}$ abgeschlossen ist.
b) Es gilt $\square \in S^* \text{ gdw } S^*$ ist H-unerfüllbar.
c) Ist S unerfüllbar, so gilt $\square \in S^*$, also $S \overset{*}{\underset{\mathcal{R}}{\vdash}} \square$.

Also

Ist S unerfüllbar, so zeigt jede „faire" $\mathcal{R}$-Ableitung, dass $S \overset{*}{\underset{\mathcal{R}}{\vdash}} \square$ gilt.

Konstruktion eines Modells: Die Idee

Sei

S eine Menge von Grundklauseln über sig $= (F, P, \alpha)$.

GAtom die Menge der Grundatome A über sig.

C eine Klausel, also eine Multimenge von Literalen.

$I \subseteq$ GAtom ein H-Modell.

Dann: $I \models C$ gdw Es gibt A mit $A \in C \cap I$ oder $\neg A \in C, A \notin I$

$I \models S$ gdw $I \models C$ für alle $C \in S$.

Idee: Wir zeigen:

Sei S abgeschlossen unter $\mathcal{R}$ und $\Box \notin S$. Dann ist S H-erfüllbar.

Beweis-Idee

Sei $\succ$ totale Noethersche Ordnung auf GAtom.

Definiere Noethersche Ordnung $\succ_c$ auf Klauseln.

Definiere durch Induktion nach $\succ_c$ ein Modell $I \subseteq$ GAtom, so dass $I \models C$ für alle $C \in S$. Dann ist I ein Modell für S.

Eine Ordnung auf Literalen und Klauseln

Sei $\succ$ totale Noethersche Ordnung auf GAtom.

Wir definieren zunächst die Komplexitäten comp(l) und comp(C) für Literale l und Klauseln C.

Definition 4.4.

Sei comp$(A) = \{A\}$ und comp$(\neg A) = \{A, A\}$.
Dann ist comp(C) die Multimenge der comp(l) mit l in C.
Setze

$$l_1 \succ_l l_2 \text{ gdw } \mathrm{comp}(l_1) \succ\!\succ \mathrm{comp}(l_2)$$
$$C_1 \succ_c C_2 \text{ gdw } \mathrm{comp}(C_1) \succ\!\succ_l \mathrm{comp}(C_2)$$

Beispiel

Sei $C_1 = \{A_1, A_2, A_2, A_3, A_3\}$, $C_2 = \{A_1, \neg A_2, A_3, A_3\}$.

Es ist comp$(C_2) = \{\{A_1\}, \{A_2, A_2\}, \{A_3\}, \{A_3\}\}$ und comp$(C_1) = \{\{A_1\}, \{A_2\}, \{A_2\}, \{A_3\}, \{A_3\}\}$, also $C_2 \succ_c C_1$.

Allgemein gilt: Aus $C_1 \subseteq C_2$ folgt $C_2 \succeq_c C_1$.

Eine Ordnung auf Literalen und Klauseln (Forts.)

Beachte

a) $\succ_l$ und $\succ_c$ sind Noethersche Ordnungen auf den Grundliteralen bzw. den Grundklauseln.

b) Es gilt stets $\neg A \succ_l A$. Aus $A \succ B$ folgt $A \succ_l \neg B \succ_l B$.

c) Gilt $A \succeq_l l$ für alle $l \in C \vee D$, so gilt $\neg A \vee D \succ_c A \vee C$.

Bezeichnung

Sei $C \equiv l \vee C'$. Es heißt l (**strikt**) **maximal** in C, falls $l \succeq_l l'$ (bzw. $l \succ l'$) für alle $l' \in C'$ gilt.

Beachte

Da die Ordnung $\succ$ total ist, existiert ein maximales Literal l in C. Tritt l nur einmal in C auf, so ist l strikt maximal in C.

Kandidatenmodelle

Ansatz Verkleinerung von Gegenbeispielen

Wir ordnen jeder Klauselmenge S ein H-Modell I_S zu mit:

Gilt nicht $I_S \models S$ (also gibt es ein **Gegenbeispiel** $C \in S$ mit $I_S \not\models C$), so kann man aus S durch einen $\mathcal{R}$-Schritt ein Gegenbeispiel D erzeugen mit $C \succ_c D$.

Dazu: Wir definieren durch Induktion nach $\succ_c$ zu jeder Klausel C eine Menge ε_C von Atomen und die H-Interpretation $I_C = \bigcup_{C \succ_c D} \varepsilon_D$.

Definition 4.5.

Sei C eine Klausel und $I_C = \bigcup_{C \succ_c D} \varepsilon_D$ schon gegeben. Gilt

(1) $C \in S$,

(2) $C \equiv A \vee C'$ und A ist strikt maximal in C und

(3) $I_C \not\models C$,

so ist $\varepsilon_C = \{A\}$, sonst ist $\varepsilon_C = \emptyset$.

Wir sagen, C **produziert** A und C ist **produktiv**, falls $\varepsilon_C = \{A\}$.

Das **Kandidatenmodell** $I_S = I_S^{\succ}$ ist $I_S = \bigcup_{C \in S} \varepsilon_C$.

I_C ist die partielle Interpretation unterhalb C.

Sei $I^C = I_C \cup \varepsilon_C$.

Beispiele

Beispiel 1

Sei $A \succ B$ und $S = \{C_1, \ldots, C_4\}$. Es ergibt sich

$C \equiv C_i$	I_C	ε_C	Bemerkung
$B \vee \neg B$	$\varnothing$	$\varnothing$	Tautologie
$A \vee B$	$\varnothing$	$\{A\}$	A ist maximal
$\neg A \vee \neg B$	$\{A\}$	$\varnothing$	bereits erfüllt
$A \vee \neg A$	$\{A\}$	$\varnothing$	Tautologie

S ist abgeschlossen unter $\mathcal{R}$, es gilt $I_S = \{A\}$ und $I_S \models S$.

Beispiel 2

Sei $B_1 \succ A_1 \succ B_0 \succ A_0$ und $S = \{C_1, \ldots, C_5\}$.

$C \equiv C_i$	I_C	ε_C	Bemerkung
$A_0 \vee B_0$	$\varnothing$	$\{B_0\}$	B_0 ist maximal
$B_0 \vee A_1$	$\{B_0\}$	$\varnothing$	bereits erfüllt
$\neg B_0 \vee A_1$	$\{B_0\}$	$\{A_1\}$	A_1 ist maximal
$\neg B_0 \vee \neg A_1$	$\{B_0, A_1\}$	$\varnothing$	nicht erfüllt
$\neg B_0 \vee \neg B_1$	$\{B_0, A_1\}$	$\varnothing$	bereits erfüllt

S ist nicht abgeschlossen unter $\mathcal{R}$. Es ist $I_S = \{A_1, B_0\}$ und es gilt nicht $I_S \models S$. Aus S ist z. B. erst die Klausel $\neg B_0$ und dann A_0 und A_1 erzeugbar. Für $I = \{A_0, A_1\}$ gilt $I \models S$.

Eigenschaften der Modellkonstruktion (1)

Beachte

Sei $C \succ_c D$ und l das maximale Literal in C. Dann gilt $l \succeq_l l'$ für alle $l' \in D$. Ist l strikt maximal in C und $l \in D$, so ist l auch strikt maximal in D. Weiter gilt $I_D \subseteq I^D \subseteq I_C \subseteq I_S$.

Lemma 4.4

a) Gilt $D' \succ_c D \succeq_c C$ und $I_D \models C$, so gilt auch $I_{D'} \models C$ und $I_S \models C$.

b) Gilt $D \succ_c C$ und $I_S \models C$, so gilt $I_D \models C$.

Beweis:

a) Es gibt ein A mit (1) $A \in C$ und $A \in I_D$ oder (2) $\neg A \in C$ und $A \notin I_D$.
zu (1) Aus $A \in I_D$ folgt $A \in I_{D'} \subseteq I_S$, also $I_{D'} \models C$ und $I_S \models C$.
zu (2) Wir zeigen $A \notin I_S$. Dann gilt $A \notin I_{D'}$ und $I_{D'} \models C$ und $I_S \models C$.
Annahme: $A \in I_S$. Dann wird A von einem $C_0 \equiv A \vee C'$ produziert.
Gilt $C_0 \succeq_c C$, so gilt $A \succeq_l l$ für alle $l \in C$, also $A \succeq_l \neg A$. $\lightning$
Gilt $C \succ_c C_0$, so gilt $D \succeq C \succ C_0$, also $A \in I_D$. $\lightning$

Beweis (Forts.)

b) Gilt $I_C \models C$, so gilt $I_D \models C$ nach a). Sei also $I_C \not\models C$.
Es gibt ein A mit (1) $A \in C$ und $A \in I_S$ oder (2) $\neg A \in C$ und $A \notin I_S$.
Zu (1): Wegen $A \in I_S$ wird A von einem $D_0 \equiv A \vee D'$ produziert. Ist $C \succeq_c D_0$, so gilt $A \in I_D$ wegen $D \succ_c D_0$.
Ist $D_0 \succ_c C$, so ist A strikt maximal in C nach Beachte. Wegen $I_C \not\models C$ produziert C A. Dann gilt $A \in I_{D_0}$ und somit $I_{D_0} \models D_0$. Also kann D_0 nicht produktiv sein. $\lightning$
Zu (2): Aus $A \notin I_S$ folgt $A \notin I_D$, also gilt $I_D \models C$.

Eigenschaften der Modellkonstruktion (2)

Lemma 4.5 Sei $C \in S$.

a) Ist $C \equiv A \vee C'$ und A strikt maximal in C, so gilt $I_S \models C$.

b) Ist $I_S \not\models C$ und l maximal in C, so gilt (1) $l \equiv \neg A$ oder (2) $l \equiv A$ und $C \equiv A \vee A \vee C'$.

Beweis:

a) Produziert C A, so gilt $A \in I^C \subseteq I_S$, also $I_S \models C$. Sonst gilt $I_C \models C$, also auch $I_S \models C$ nach Lemma 4.4 a).

b) Ist l negativ, also $l \equiv \neg A$, so ist nichts zu zeigen. Sei $l \equiv A$. Dann ist A nicht strikt maximal in C nach a). Also gilt $C \equiv A \vee A \vee C'$.

Eigenschaften der Modellkonstruktion (3)

Satz 4.1. ($\mathcal{R}$ verkleinert Gegenbeispiele)

Sei $C \not\equiv \square$, $C \in S$ und $I_S \not\models C$. Dann gibt es eine $\mathcal{R}$-Inferenz mit C als Hypothese und Konklusion D, so dass $I_S \not\models D$ und $C \succ_c D$ gilt.

Beweis:

Nach Lemma 4.5, Teil b) hat C die Form $C \equiv \neg A \vee C'$ oder $C \equiv A \vee A \vee C'$, wobei $\neg A$ bzw. A maximal in C ist.

a) Ist $C \equiv A \vee A \vee C'$, so liefert eine Faktorisierungsinferenz $D \equiv A \vee C'$. Es gilt $C \succ_c D$ wegen $C \supset D$ und $I_S \not\models D$ wegen $I_S \not\models C$.

b) Sei $C \equiv \neg A \vee C'$. Wegen $I_S \not\models C$ gilt $A \in I_S$, also wird A von einem $C_0 \equiv A \vee C_0'$ aus S produziert. Eine Resolutionsinferenz mit Hypothesen C_0 und C liefert $D \equiv C_0' \vee C'$. Da C_0 A produziert, gilt $\neg A \succ_l A \succ_l l$ für alle $l \in C_0'$. Dies zeigt $C \equiv \neg A \vee C' \succ_c C_0' \vee C' \equiv D$. Da C_0 A produziert, gilt $I_{C_0} \not\models C_0$, also auch $I_{C_0} \not\models C_0'$. Lemma 4.4 b) liefert jetzt $I_S \not\models C_0'$. Wegen $I_S \not\models C$ gilt auch $I_S \not\models C'$. Dies zeigt $I_S \not\models D \equiv C_0' \vee C'$.

Ergebnis für $\mathcal{R}$ und Grundklauseln

Sei S eine Menge von Grundklauseln.

Satz 4.2.

Sei S abgeschlossen unter $\mathcal{R}$ und $\square \notin S$. Dann ist S H-erfüllbar.

Beweis:

Gilt $I_S \not\models S$, so gibt es ein minimales Gegenbeispiel $C \in S$. Nach Satz 4.1 gibt es dann auch ein Gegenbeispiel D mit $C \succ_c D$ und $D \in S$, da S unter $\mathcal{R}$ abgeschlossen ist. Dies ist nicht möglich.

Geordnete Grund-Resolution (1)

Beachte

Der Resolutionsschritt „$A \vee C$ und $\neg A \vee D$ liefert $C \vee D$“ ist in $\mathcal{R}$ für jedes A anwendbar.

Dies führt zu einem sehr großem Suchraum bei der Suche nach einer Herleitung von $\Box \in S$.

Der Beweis zu Satz 4.2 liefert ein Mittel zur Suchraumeinschränkung.

Definition 4.6.

Sei $\succ$ totale Noethersche Partialordnung auf GAtom und $\succ_c$ wie oben.

Der Kalkül **geordnete Resolution** $\mathcal{R}_\succ$ zu $\succ$ besteht aus den Inferenzregeln

$$\frac{A \vee C \quad \neg A \vee D}{C \vee D} \quad \text{falls } A \text{ strikt maximal in } A \vee C \text{ und } \neg A \text{ maximal in } \neg A \vee D \text{ ist}$$

$$\frac{A \vee A \vee C}{A \vee C} \quad \text{falls } A \text{ maximal in } A \vee A \vee C \text{ ist}$$

Geordnete Grund-Resolution (2)

Beachte

$\mathcal{R}_\succ$ beschränkt

a) Resolutionsschritte auf Atome, die maximal in beiden Prämissen sind und
b) Faktorisierungsschritte auf positive Literale A, so dass A maximal in der Prämisse ist.

Satz 4.3.

Satz 4.2 gilt auch für $\mathcal{R}_\succ$ statt $\mathcal{R}$.

Beweis: Übung. Der Beweis ist fast wörtlich übertragbar.

Bemerkung

Diese Suchraumbeschränkung gilt nicht nur für Grundklauseln. Sie ist auch auf die Resolution für beliebige Klauselmengen übertragbar. Siehe später.

Saturierung mit $\mathcal{R}_\succ$ für Grundklauseln

Definition 4.7.

Eine $\mathcal{R}_\succ$-Ableitung $(S_i)_{i\in\mathbb{N}}$ heißt **fair**, falls $S^* = \bigcup_{i\in\mathbb{N}} S_i$ abgeschlossen unter $\mathcal{R}_\succ$ ist.

Beachte

a) Fairness heißt, dass jede mögliche Resolutions- bzw. Faktorisierungsinferenz auch einmal gezogen werden muss.

b) Fairness ist leicht zu realisieren.

Lemma 4.6

a) Sei S_0 eine Menge von Grundklauseln und $(S_i)_{i\in\mathbb{N}}$ eine faire $\mathcal{R}_\succ$-Ableitung mit $S_0 = S$. Ist S_0 H-unerfüllbar, so gibt es ein $n \in \mathbb{N}$ mit $\Box \in S_n$.

b) $\mathcal{R}_\succ$ ist vollständig.

Beweis:

a) Ist S_0 H-unerfüllbar, so gilt dies auch für $S^* = \bigcup_{i\in\mathbb{N}} S_i$. Dann ist S^* abgeschlossen unter $\mathcal{R}_\succ$. Es gilt $\Box \in S^*$ nach Satz 4.3, also gibt es ein $n \in \mathbb{N}$ mit $\Box \in S_n$.

b) Für jedes H-unerfüllbare S gilt $S \overset{*}{\underset{\mathcal{R}_\succ}{\vdash}} \Box$ nach a).

Hauptergebnis für Grundklauseln

Satz 4.4.

Der Resolutionskalkül $\mathcal{R}_\succ$ ist korrekt und vollständig für Grundklauseln: Ist S eine Menge von Grundklauseln, so gilt $S \overset{*}{\underset{\mathcal{R}_\succ}{\vdash}} \Box \text{ gdw } S$ ist H-unerfüllbar.

Beweis:

Dies folgt aus Lemma 4.3 und Lemma 4.6.

Zusatz

Der obige Satz gilt erst recht für $\mathcal{R}$ statt $\mathcal{R}_\succ$, da jede $\mathcal{R}_\succ$-Inferenz auch eine $\mathcal{R}$-Inferenz ist.

Das Lifting-Lemma

Ziel

$\mathcal{R}_\succ$ ist korrekt und vollständig für beliebige Klauselmengen S.

Ansatz

Wir führen die Ideen wieder erst für $\mathcal{R}$ ein und übertragen sie dann auf $\mathcal{R}_\succ$.

S ist H-unerfüllbar gdw gnd(S) ist H-unerfüllbar.

Zeige: $\mathcal{R}_\succ$ führt auf S die Inferenzschritte der Resolution auf gnd(S) parametrisiert durch: Die Variablen sind zu Grundtermen instanziierbare Parameter.

Lemma 4.7 (Lifting-Lemma)

Seien C_1' und C_2' Grundinstanzen von C_1 bzw. C_2 und sei C' eine Resolvente von C_1', C_2'. Dann gibt es eine Resolvente C von C_1, C_2, so dass C' eine Grundinstanz von C ist.

Beweis: Seien o.B.d.A. C_1, C_2 variablendisjunkt. Es gilt

$C_1 \equiv A \vee D_1 \qquad C_2 \equiv \neg B \vee D_2$

$C_1' \equiv \tau(A) \vee \tau(D_1) \quad C_2' \equiv \neg\tau(B) \vee \tau(D_2)$

$C' \equiv \tau(D_1) \vee \tau(D_2) \quad \tau(A) \equiv \tau(B)$

Also existiert $\sigma =$ mgu(A, B) und τ' mit $\tau = \tau' \circ \sigma$ auf Var(C_i).

Dann ist $C \equiv \sigma(D_1) \vee \sigma(D_2)$ ist Resolvente zu C_1, C_2.

Es ist $C' \equiv \tau'(C)$ eine Instanz von C.

Der Resolutionskalkül für beliebige Klauseln (1)

Sei

gnd$(S) = \{\sigma(C) \mid C \in S, \sigma(C)$ Grundklausel$\}$.

Also S ist H-erfüllbar gdw gnd(S) ist H-erfüllbar.

Beachte

Lemma 4.7 „liftet“ Resolutionsschritte. Ein analoges Lemma gilt für Faktorisierungsschritte.

Lemma 4.8

Sei $C_1', \ldots, C_n'$ eine $\mathcal{R}$-Herleitung aus gnd(S).

Dann gibt es eine $\mathcal{R}$-Herleitung $C_1, \ldots, C_n$ aus S und Grundsubstitutionen τ_i, so dass $C_i' \equiv \tau_i(C_i)$ für $i = 1, \ldots, n$.

Beweis:

Induktion nach n mit dem Lifting-Lemma.

Der Resolutionskalkül für beliebige Klauseln (2)

Satz 4.5.

Der Resolutionskalkül $\mathcal{R}$ ist korrekt und vollständig für beliebige Klauselmengen S: Es gilt $S \overset{*}{\underset{\mathcal{R}}{\vdash}} \Box \ \mathrm{gdw}\ S$ ist H-unerfüllbar.

Beweis:

Zur Korrektheit: Dies folgt aus Lemma 4.2.

Zur Vollständigkeit: Sei S unerfüllbar.

Es gibt nach Satz 4.4 eine $\mathcal{R}$-Herleitung $C_1', \ldots, C_n' \equiv \Box$ aus gnd(S).

Nach dem letzten Lemma gibt es also eine $\mathcal{R}$-Herleitung $C_1, \ldots, C_n$ aus S mit $\Box \equiv \tau_n(C_n)$, also $C_n \equiv \Box$.

Also gilt $S \overset{*}{\underset{\mathcal{R}}{\vdash}} \Box$.

Geordnete Resolution auf beliebigen Klauseln (1)

Ziel

Angabe von $\mathcal{R}_\succ$ auf Klauseln, so dass $\mathcal{R}_\succ$ korrekt und vollständig ist.

Definition 4.8.

Eine Noethersche Partialordnung $\succ$ auf den Atomen heißt **zulässig**, falls gilt: (1) $\succ$ ist total auf den Grundatomen und (2) aus $A \succ B$ folgt $\sigma(A) \succ \sigma(B)$ für jede Grundsubstitution σ.

Beachte

Sei $F_P = F \cup P$ die Menge der Funktions- und Prädikatssymbole. Dann ist Term(F_P, V) definiert. Sei $\succ$ eine LPO oder KBO mit totaler Präzedenz auf F_P. Dann ist $\succ$ eine zulässige Partialordnung auf den Atomen.

Beachte

Eine zulässige Ordnung $\succ$ ist i. Allg. nicht total auf den Klauseln.
Mit $\succ$ soll die Resolutionsinferenz aus $\mathcal{R}$ so eingeschränkt werden, dass jede $\mathcal{R}_\succ$-Grundinferenz „$\sigma(A) \vee \sigma(C)$ und $\sigma(\neg B) \vee \sigma(D)$ liefert $\sigma(C \vee D)$“ noch „liftbar“ ist. Eine analoge Forderung gilt für die Faktorisierungsinferenz. Wir zeigen, dass dann $\mathcal{R}_\succ$ noch vollständig ist.

Geordnete Resolution auf beliebigen Klauseln (2)

Bezeichnung

Sei $\succ$ eine zulässige Partialordnung auf den Atomen. Dann sind die Ordnungen $\succ_l$ auf den Literalen und $\succ_c$ auf den Klauseln analog zu früher definiert:

$$l \succ_l l' \text{ gdw } \mathsf{comp}(l) \succ\!\!\succ \mathsf{comp}(l')$$
$$C \succ_c C' \text{ gdw } \mathsf{comp}(C) \succ\!\!\succ \mathsf{comp}(C')$$

Das Literal l heißt (**strikt**) **maximal** in $C \vee l$, falls $l' \not\succ_l l$ (bzw. $l' \succeq_l l$) für alle l' aus C gilt.

Definition 4.9.

Sei $\succ$ zulässig. Der Kalkül $\mathcal{R}_\succ$ auf den Klauseln ist gegeben durch

$$\frac{A \vee C \quad \neg B \vee D}{\sigma(C \vee D)} \quad \begin{array}{l} \text{falls } \sigma = \mathsf{mgu}(A, B), \\ \sigma(A) \text{ ist strikt maximal in } \sigma(A \vee C), \\ \sigma(\neg B) \text{ ist maximal in } \sigma(\neg B \vee D). \end{array}$$

$$\frac{A \vee B \vee C}{\sigma(B \vee C)} \quad \begin{array}{l} \text{falls } \sigma = \mathsf{mgu}(A, B), \\ \sigma(A) \text{ ist maximal in } \sigma(A \vee B \vee C) \end{array}$$

Beachte

Auf den Grundklauseln sind die $\mathcal{R}_\succ$-Inferenzen genauso wie die $\mathcal{R}_\succ$-Inferenzen von früher definiert.

Ist eine Grundinstanz $\tau(A)$ (strikt) maximal in $\tau(A \vee C)$, so ist A (strikt) maximal in $A \vee C$.

Eine Klausel C kann mehrere strikt maximale Literale besitzen.

Geordnete Resolution auf beliebigen Klauseln (3)

Frage

Ist jede $\mathcal{R}_\succ$-Inferenz auf Grundklauseln liftbar?

Definition 4.10.

Sei $\mathcal{K}$ ein Kalkül auf Klauseln. Sei C_i eine Klausel und $C_i' \equiv \sigma_i(C_i)$ eine Grundinstanz von C_i für $1 \leq i \leq n$.

Sei $\dfrac{C_1' \ldots C_n'}{D'}$ eine $\mathcal{K}$-Grundinferenz. Sie heißt **liftbar**, falls es eine $\mathcal{K}$-Inferenz $\dfrac{C_1 \ldots C_n}{D}$ gibt, so dass D' eine Instanz von D ist.

Lemma 4.9 (Lifting-Lemma)

Jede $\mathcal{R}_\succ$-Grundinferenz ist liftbar.

Beweis: Übung.

Folgerung

Sei $C_1', \ldots, C_n'$ eine $\mathcal{R}_\succ$-Herleitung von C_n' aus $\mathsf{gnd}(S)$. Dann gibt es eine $\mathcal{R}_\succ$-Herleitung $C_1, \ldots, C_n$ von C_n aus S und Grundsubstitutionen τ_i, so dass $C_i' \equiv \tau_i(C_i)$ für $i = 1, \ldots, n$.

Hauptergebnis für $\mathcal{R}_\succ$ auf Klauseln

Satz 4.6.

$\mathcal{R}_\succ$ ist korrekt und vollständig: Für jede Klauselmenge S gilt
$S \underset{\mathcal{R}_\succ}{\vdash} \Box \text{ gdw } S$ ist H-unerfüllbar.

Beweis:

a) (Korrektheit) Sei $S \underset{\mathcal{R}_\succ}{\vdash} \Box$.
Dann ist S H-unerfüllbar nach Lemma 4.3.

b) (Vollständigkeit) Sei S H-unerfüllbar.
Dann ist auch gnd(S) unerfüllbar. Es gibt nach Satz 4.4 eine $\mathcal{R}_\succ$-Herleitung $C_1', \ldots, C_n' \equiv \Box$ von $\Box$ aus gnd(S).
Also gibt es eine $\mathcal{R}_\succ$-Herleitung $C_1, \ldots, C_n$ von C_n aus S und ein τ_n mit $\tau_n(C_n) \equiv C_n' \equiv \Box$.
Also ist $C_n \equiv \Box$ und es gilt $S \underset{\mathcal{R}_\succ}{\vdash} \Box$.

Paramodulation für PL1=

Sei der Einfachheit halber

$P = \{=\}$.

$S =$ Menge von Klauseln über sig $= (F, P, \tau)$.

$C \vee s \doteq t$ steht für $C \vee s = t$ oder $C \vee t = s$.

Definition 4.11.

Der Paramodulationskalkül $\mathcal{P}$ besteht aus 3 Inferenzregeln:

Paramodulation

$$\frac{C \vee s \doteq t \qquad D[u]_\pi}{\sigma(C \vee D[t]_\pi)} \qquad \text{falls } \sigma = \text{mgu}(s, u)$$

Faktorisierung

$$\frac{C \vee s \doteq t \vee u = v}{\sigma(C \vee s = t)} \qquad \text{falls } \sigma = \text{mgu}\{(s, u), (t, v)\}$$

Resolution mit dem Reflexivitätsaxiom (Ref. Resolution)

$$\frac{C \vee \neg s = t}{\sigma(C)} \qquad \text{falls } \sigma = \text{mgu}(s, t)$$

Ergebnis zum Paramodulationskalkül

Bezeichnung

S heißt **funktional-reflexiv**, falls S für jedes $f \in F$ mit $\alpha(f) = n$ die Gleichung $f(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$ enthält.

Satz 4.7. (Robinson, Wos)

Sei S funktional-reflexiv und abgeschlossen unter den Inferenzregeln des Paramodulationskalküls. Dann gilt:
S ist nicht H-erfüllbar $\text{gdw} \quad \Box \in S$.

ohne Beweis. Dieses Ergebnis wird später wesentlich verschärft.

Also

Mit der Bezeichnung von oben gilt für jedes funktional-reflexive S

$S \overset{*}{\underset{\mathcal{P}}{\vdash}} \Box \quad \text{gdw} \quad S$ ist nicht H-erfüllbar.

Literatur-Hinweis

G. Robinson, L.T. Wos: Paramodulation and theorem proving in first order theories with equality. In: B. Meltzer, D. Michie (eds): Machine Intelligence 4, American Elsevier (1969) 133-150.

Beispiel 1 zur Paramodulation

Sei

$S = \{(1), (2)\}$

(1) $f(x) = g(a)$

(2) $\neg f(b) = g(y)$

Anwendung der Regeln liefert

(3) $\neg g(a) = g(y)$ Paramodulation mit (1), (2)

(4) $\Box$ Ref. Resolution mit (3)

Also ist S nicht H-erfüllbar.

Beachte

Hier werden die funktional-reflexiven Axiome $f(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$ nicht benötigt.

Beispiel 2 zur Paramodulation

Schreibweise hier: $p(x, y)$ statt $p(x, y) = \top$.

Zu zeigen (*) $\mathcal{A}x \models p(a, d)$ mit

$\mathcal{A}x : \forall x, y, z : (p(x, y) \wedge p(y, z) \Rightarrow p(x, z))$
$\quad p(a, b) \wedge p(b, c) \wedge p(c, d)$

Skolemisierung liefert $S = \{(1), \ldots, (5)\}$

(1) $\neg p(x, y) = \top \vee \neg p(y, z) = \top \vee p(x, z) = \top$
(2) $p(a, b) = \top$
(3) $p(b, c) = \top$
(4) $p(c, d) = \top$
(5) $\neg p(a, d) = \top$

Anwendung der Regel Paramodulation auf (1), (2) mit

C: $\square$
$s = t$: $p(a, b) = \top$
D: $\neg p(x, y) = \top \vee \neg p(y, z) = \top \vee p(x, z) = \top$

liefert $\sigma = \{x \leftarrow a, y \leftarrow b\}$, also die neue Klausel

(6) $\neg\top = \top \vee \neg p(b, z) = \top \vee p(a, z) = \top$

Beispiel 2 zur Paramodulation (Forts.)

Die Regeln liefern

(6)	$\neg\top = \top \vee \neg p(b, z) = \top \vee p(a, z) = \top$	Paramod. (2), (1)
(7)	$\neg p(b, z) = \top \vee p(a, z) = \top$	Ref. Resolution (6)
(8)	$\neg\top = \top \vee p(a, c) = \top$	Paramod. (3), (7)
(9)	$p(a, c) = \top$	Ref. Resolution (8)
(10)	$\neg\top = \top \vee \neg p(c, z) = \top \vee p(a, z) = \top$	Paramod. (9), (1)
(11)	$\neg p(c, z) = \top \vee p(a, z) = \top$	Ref. Resolution (10)
(12)	$\neg\top = \top \vee p(a, d) = \top$	Paramod. (4), (11)
(13)	$p(a, d) = \top$	Ref. Resolution (12)
(14)	$\neg\top = \top$	Paramod. (13), (5)
(15)	$\square$	Ref. Resolution (14)

Also

S ist H-unerfüllbar.
(*) gilt.

Bemerkungen

1. Bemerkung

Die Paramodulationsregel ist sehr oft anwendbar. Dies führt zu sehr großem Suchraum bei der Suche nach einer Ableitung $S \overset{*}{\underset{\mathcal{P}}{\vdash}} \square$.

Ziel: Suchraum einschränken.

2. Bemerkung

Es gibt schon lange Beschränkungen des Suchraums, die die Vollständigkeit erhalten.

Beispiele: Hyperresolution, Selektion negativer Literale, set-of-support-Strategie.

3. Bemerkung

Ziel: Beschränkung der Inferenzregeln durch Ordnungsbedingungen.

Beachte

Dies führt im Fall der positiven Unitklauseln (reines Gleichheitsbeweisen) zu Rewrite-Schritten und der Bildung kritischer Paare. Negative Unitklauseln werden wie verallgemeinerte Narrowingsregeln behandelt.

Saturierung

Bezeichnung

Ein Kalkül heißt **Saturierungskalkül**, wenn er die gegebene Formelmenge mit den Inferenzregeln immer weiter anreichert. Diese Saturierung kann, muss aber nicht halten. Sie wird abgebrochen, wenn die leere Klausel $\square$ hergeleitet ist oder keine neue Klausel mehr erzeugt werden kann.

Beachte

a) Es gibt auch **zerlegende Beweisverfahren**, die die Formeln zerlegen bis die Frage der Erfüllbarkeit geklärt ist. Beispiel: Das Tableau-Verfahren.
b) Wir betrachten hier nur saturierende Verfahren.

4.2 Superposition für Horn-Klauseln

Voraussetzung für Abschnitt 4.2

$=$ ist das einzige Prädikatssymbol.

S ist eine Menge von Horn-Klauseln. Also hat jede Klausel $\Gamma \Rightarrow \Delta$ die Form $\Gamma \Rightarrow l = r$ oder $\Gamma \Rightarrow$

$\succ$ ist eine totale Reduktionsordnung auf den Grundtermen Term(F).

$\succeq = (\succ \cup =)$ ist der reflexive Abschluß von $\succ$.

Beachte

Seien $s, t \in$ Term(F), $s >_{TT} t$. Dann gilt auch $s \succ t$. (Übung!)

Also enthält $\succ$ (auf Term(F)) die Teiltermrelation.

Häufige Wahl von $\succ$: Eine KBO oder LPO mit totaler Präzedenz.

Definition 4.12.

Sei $\Gamma \Rightarrow l = r$ eine Grund-Horn-Klausel.

a) l ist **strikt maximal** in $\Gamma \Rightarrow l = r$ (Schreibweise $l \succ r, \Gamma$), falls $l \succ t$ für alle $t \in \Gamma \cup \{r\}$.

b) l ist **maximal** in $\Gamma \Rightarrow l = r$ (Schreibweise $l \succeq r, \Gamma$), falls $l \succeq t$ für alle $t \in \Gamma \cup \{r\}$.

Das System $\mathcal{GH}$ für Grund-Horn-Klauseln

Definition 4.13.

Das Inferenzsystem $\mathcal{GH} = \mathcal{GH}_{\succ}$ besteht aus folgenden Regeln:

Superposition rechts

$$\frac{\Gamma' \Rightarrow l \doteq r \qquad \Gamma \Rightarrow s \doteq t}{\Gamma', \Gamma \Rightarrow s[r]_\pi = t} \qquad \text{falls } s/\pi \equiv l,\ \ l \succ r, \Gamma' \text{ und } s \succ t, \Gamma$$

Superposition links

$$\frac{\Gamma' \Rightarrow l \doteq r \qquad \Gamma, s \doteq t \Rightarrow \Delta}{\Gamma', \Gamma, s[r]_\pi = t \Rightarrow \Delta} \qquad \text{falls } s/\pi \equiv l,\ \ s \succ t,\ l \succ r, \Gamma' \text{ und } s \succeq \Gamma, \Delta$$

Reflexiv-Resolution

$$\frac{\Gamma, s = s \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \qquad \text{falls } s \succeq \Gamma, \Delta$$

Also: Einschränkung der Regeln des Paramodulationskalküls:

Die Paramodulation wird eingeschränkt auf die jeweils (strikt) maximalen Terme in der Klausel. Dies gilt auch für die Reflexiv-Resolution.

Beachte

Hier treten nur Grundterme auf, also ist der Test auf Unifizierbarkeit identisch mit dem Test auf Identität.

Beispiel 1 für $\mathcal{GH}$

Sei S gegeben durch

(1) $a = b \Rightarrow c = d$

(2) $a = c \Rightarrow b = d$

(3) $c = d \Rightarrow$

und $\succ$ die LPO zur Präzedenz $a \rhd b \rhd c \rhd d$.

a) Im Kalkül $\mathcal{GH}$ ist keine Inferenzregel anwendbar, die Saturierung stoppt sofort. S ist H-erfüllbar, z. B. durch $I_1 = \varnothing$ und $I_2 = \{b = d\}$.

b) Im Paramodulationskalkül $\mathcal{P}$ sind zwischen (1) und (2) z. B. folgende Paramodulationsschritte möglich.

(4) $a = b, a = \underline{d} \Rightarrow b = d$

(5) $a = b, a = c \Rightarrow b = \underline{c}$

Hier ist jeweils $C \vee s = t$ die Klausel (1) und D die Klausel (2). Der neu entstandene Term ist unterstrichen.

S erlaubt noch weitere Paramodulationsschritte.

Beispiel 2 für $\mathcal{GH}$

Sei $\succ$ die LPO zur Präzedenz $a \rhd b \rhd c \rhd d$ und $S = \{(1), \ldots, (4)\}$

(1) $\Rightarrow a = b$

(2) $a = b \Rightarrow a = c$

(3) $b = c \Rightarrow c = d$

(4) $b = d \Rightarrow$

Dann liefert die Saturierung mit $\mathcal{GH}$

(5)	$b = b \Rightarrow a = c$	Superposition links mit (1), (2)
(6)	$b = b \Rightarrow b = c$	Superposition rechts mit (1), (5)
(7)	$\Rightarrow b = c$	Reflexiv-Resolution mit (6)
(8)	$c = c \Rightarrow c = d$	Superposition links mit (7), (3)
(9)	$\Rightarrow c = d$	Reflexiv-Resolution mit (8)
(10)	$c = d \Rightarrow$	Superposition links mit (7), (4)
(11)	$d = d \Rightarrow$	Superposition links mit (9), (10)
(12)	$\Box$	Reflexiv-Resolution mit (11)

Übung (1), (2), (3) $\models b = d$.

Also ist S nicht H-erfüllbar.

Modellgenerierung: Die Idee

Ziel

Regelsystem R zu S konstruieren mit

a) R ist konvergent.

b) Ist S saturiert und $\Box \notin S$, so ist die R-Gleichheit $=_R$ ein Modell für S, d. h. $=_R \models C$ für alle C in S.
(Beachte, dass $I = =_R$ ein H-Modell zu R ist.)

Konstruktion von R aus S

Induktion nach der Ordnung $\succ_c$ auf Klauseln.

Definition 4.14. (Die Ordnung $\succ_c$ auf Klauseln)

Sei C die Klausel $\Gamma \Rightarrow \Delta$ und $s = t$ eine Gleichung in $\Gamma \cup \Delta$. Sei $\succ_0 = \succ\!\!\succ$ und $\succ_1 = \succ\!\!\succ_0$. Setze

$$\text{comp}(C, s = t) = \begin{cases} \{s, t\} & s = t \text{ in } \Delta \\ \{s, s, t, t\} & s = t \text{ in } \Gamma \end{cases}$$

$\text{comp}(C) = \{\text{comp}(C, s = t) \mid s = t \text{ in } \Gamma \cup \Delta\}$

Dann ist $\succ_c$ gegeben durch

$$C \succ_c D \ \text{ gdw } \ \text{comp}(C) \succ_1 \text{comp}(D)$$

Bemerkung zu $\succ_0$ und $\succ_c$

Bemerkung

Sei $C \equiv \Gamma \Rightarrow \Delta$, sei $s = t_1$ in Γ und $s = t_2$ in Δ und sei $s \succeq t_1, t_2$. Dann gilt $\text{comp}(C, s = t_1) \succ_0 \text{comp}(C, s = t_2)$.

Dies gilt auch für $t_1 \equiv t_2$. Eine Gleichung $s = t$ hat also eine größere Komplexität, wenn sie in Γ liegt, als wenn sie in Δ liegt.

Beachte

$C \succ_c D$ falls i) oder ii) oder iii):

i) Der größte Term in C ist echt größer als der größte Term in D.

ii) Der größte Term in C ist auch der größte Term in D und er kommt in C negativ und in D nur positiv vor.

iii) D entsteht aus C durch Ersetzen eines Terms durch einen kleineren Term oder durch Löschen eines Literals und gleichzeitiges Hinzufügen kleinerer Literale.

Also

$\mathcal{GH}$ ist verkleinernd: Die Konklusion ist in jeder Regel kleiner als die maximale (die rechte) Prämisse.

Beachte

$\succ_c$ ist eine Noethersche Partialordnung.

Das Regelsystem R_S

Sei S eine Menge von Grund-Horn-Klauseln.

Bezeichnung

Sei R ein Regelsystem über sig $= (F, \alpha)$. Dann ist R^* die Herbrand-Algebra mit Term(F) als Universum und $=_R$ als Interpretation der Gleichheit.

Definition 4.15.

Wir definieren R_S induktiv über die Ordnung $\succ_c$. Sei für $C \in S$ das Regelsystem R_C, der durch die $D \in S$ mit $C \succ_c D$ produzierten Regeln schon definiert. Die Klausel $C \equiv \Gamma \Rightarrow l \doteq r$ **produziert** die Regel $l \longrightarrow r$ falls

1. $R_C^* \not\models C$ und
2. $l \succ r, \Gamma$ und
3. l ist irreduzibel in R_C.

Sei $R = R_S$ die Menge der durch die $C \in S$ produzierten Regeln.

Beachte

zu 1. R_C^* erfüllt nicht C, also muss $=_R$ erweitert werden.

zu 2. Dies liefert $R \subseteq \succ$, also ist R terminierend.

zu 3. Dies garantiert die Konfluenz von R.

Nicht jede Klausel produziert eine Regel. R_S ist i. Allg. kein Modell zu S.

Ein Lemma zu R_S

Lemma 4.10

Zu jedem S gilt

a) $R = R_S$ und jedes R_C ist konfluent und terminierend.

b) Für alle $C \in S$ gilt: Gilt $R_C^* \models C$, so gilt auch $R^* \models C$.

Beweis-Skizze:

a) Es gilt $R \subseteq \succ$, also ist R terminierend.
Da R ein Grundregelsystem ist, reicht für die (lokale) Konfluenz zu zeigen: Sind $l \longrightarrow r$ und $l' \longrightarrow r'$ in R, so ist l' kein Teilterm von l und l kein Teilterm von l'. Dies gilt (Übung).

b) Sei $C \equiv \Gamma \Rightarrow \Delta$ in S und $R_C^* \models C$. Dann gilt (i) $R_C^* \models \Delta$ oder (ii) $R_C^* \not\models \Gamma$.
Zu (i): Wegen $R_C \subseteq R$ gilt auch $R^* \models \Delta$, also $R^* \models C$.
Zu (ii): Es gibt $s = t$ in Γ mit $R_C^* \not\models s = t$, also nicht $s \downarrow t$ in R_C. Seien s_0, t_0 die R_C-Normalformen zu s, t. Wir zeigen: nicht $s_0 \downarrow t_0$ in R. Dann gilt $R^* \not\models s = t$, also $R^* \not\models \Gamma$ und daher $R^* \models C$.
Sonst: Sei $s_0 \succ t_0$. Dann ist s_0 reduzierbar mit $l \longrightarrow r$ in $R - R_C$. Sei $l \longrightarrow r$ erzeugt durch $D \equiv \Gamma' \Rightarrow l \doteq r$. Dann ist $D \succ_c C$. Es ist $s_0 \succeq l$ (wegen $>_{TT} \subseteq \succ$) und $l \succ r, \Gamma'$. Dies zeigt comp$(C, s = t) = \{s, s, t, t\} \succeq_0 \{s_0, s_0, t_0, t_0\} \succ_0 \{l, r\} =$ comp$(D, l = r)$. Wegen $l \succ r$, Γ' folgt jetzt $C \succ_c D$. $\lightning$

Zur Vollständigkeit von $\mathcal{GH}$

Lemma 4.11

Sei S eine Menge von Grund-Horn-Klauseln, sei S abgeschlossen bezüglich $\mathcal{GH}$ und sei $\square \notin S$. Dann ist R_S^* ein Modell von S, d. h. $R_S^* \models C$ für alle $C \in S$.

Beweis: Sei $R = R_S$

Induktion nach $\succ_c$ zeigt $R^* \models C$ für alle C in S:
Sei $R^* \models D$ für alle $C \succ_c D$, $D \in S$.
Zu zeigen ist dann $R^* \models C$.
Sei s der größte Term in C. Er existiert, da $\succ$ total auf Term(F) ist.
Es gibt 4 Fälle: a) bis d).

Die Fälle b) bis d) benutzen wesentlich, dass S abgeschlossen ist bezüglich $\mathcal{GH}$.

Zur Vollständigkeit von $\mathcal{GH}$ (Forts.)

a) $C \equiv \Gamma \Rightarrow s = s$: Dann gilt $R^* \models C$.

b) $C \equiv \Gamma \Rightarrow s = t$, $s \succ t$, s kommt in Γ nicht vor:
Es gilt $s \succ t, \Gamma$. Annahme: $R^* \not\models C$.
Dann gilt $R^* \models \Gamma$ und C erzeugt nicht die Regel $s \to t$.
Also ist s nach Definition 4.15 in R_C reduzibel.
Es gibt $l \to r$ in R_C, erzeugt durch $D \equiv \Gamma' \Rightarrow l = r$, so dass s mit $l \to r$ reduzierbar. Es gilt dann mit „Superposition rechts"

$$\frac{\Gamma' \Rightarrow l = r \qquad \Gamma \Rightarrow s = t}{D' \equiv \Gamma', \Gamma \Rightarrow s[r]_\pi = t}$$

Es ist $D' \in S$, da S abgeschlossen bezüglich $\mathcal{GH}$ ist.
Es gilt $R^* \models D$ und $R^* \models \Gamma'$ (da D eine Regel erzeugt und Lemma 4.10 gilt) und $R^* \models D'$ (nach IV, da $C \succ_c D'$). Dies liefert $R^* \models l = r$ und $R^* \models s = t$. Also gilt doch $R^* \models C$.

c) $C \equiv \Gamma, s = s \Rightarrow \Delta$:
Dann liefert „Ref.-Resolution" $D \equiv \Gamma \Rightarrow \Delta$. Es gilt $R^* \models D$ (da $C \succ_c D$ und $D \in S$), also $R^* \models C$.

d) $C \equiv \Gamma, s = t \Rightarrow \Delta$, $s \succ t$:
Analog zu b) liefert „Superposition links" $R^* \models C$.

Hauptergebnis zu $\mathcal{GH}$

Definition 4.16.

Eine $\mathcal{GH}$-Ableitung $(S_i)_{i\in\mathbb{N}}$ heißt **fair**, falls das Grenzsystem $S^* = \bigcup_{i\in\mathbb{N}} S_i$ unter $\mathcal{GH}$ abgeschlossen ist.

Satz 4.8.

$\mathcal{GH}$ ist korrekt und vollständig für Grund-Horn-Klauseln: Für jedes S gilt $S \overset{*}{\underset{\mathcal{GH}}{\vdash}} \Box \ \text{ gdw } \ S$ ist H-unerfüllbar.

Beweis:

a) Korrektheit: Für jede sig-Algebra $\mathcal{A}$ und jede Inferenzregel von $\mathcal{GH}$ gilt: Erfüllt $\mathcal{A}$ die Prämissen, so erfüllt $\mathcal{A}$ auch die Konklusion (Übung: Nachrechnen).

   Also: Gilt $S \overset{*}{\underset{\mathcal{GH}}{\vdash}} \Box$, so ist S unerfüllbar.

b) Vollständigkeit: Sei S H-unerfüllbar. Man betrachte eine faire $\mathcal{GH}$-Ableitung $(S_i)_{i\in\mathbb{N}}$ mit $S_0 = S$ und Grenzsystem S^*. Wäre $\Box \notin S^*$, so wäre nach dem letzten Lemma S^* und damit auch $S \subseteq S^*$ H-erfüllbar. Also gilt $\Box \in S^*$ und damit $\Box \in S_n$ für ein $n \in \mathbb{N}$. Dies zeigt $S \overset{*}{\underset{\mathcal{GH}}{\vdash}} \Box$.

Selektion negativer Gleichungen (1)

Beachte

Die Ordnungsrestriktionen für die Anwendbarkeit der Inferenzregeln in $\mathcal{GH}$ bewirken schon eine massive Einschränkung des Suchraums, d. h. der Erzeugung von „redundanten“ Klauseln, gegenüber dem Paramodulationskalkül. Dennoch ist $\mathcal{GH}$ vollständig (für Grund-Horn-Klauseln).

Ziel

Weitere Beschränkung der Anwendbarkeit der Inferenzregeln, ohne die Vollständigkeit zu verlieren.

Ansatz

Man markiert willkürlich in jeder Klausel $C \equiv \Gamma \Rightarrow \Delta$ mit $\Gamma \neq \varnothing$ genau eine Gleichung $s = t$ aus Γ. Mit diesen Markierungen lassen sich die Einschränkungen beschreiben.

In der Literatur wird die Markierung auch durch **selection functions** beschrieben.

Selektion negativer Gleichungen (2)

Definition 4.17. (Inferenzsystem $\mathcal{GH}'$)

Superposition rechts

$$\frac{\Gamma' \Rightarrow l \doteq r \qquad \Gamma \Rightarrow s \doteq t}{\Gamma', \Gamma \Rightarrow s[r]_\pi = t}$$

falls $s/\pi \equiv l$, $l \succ r, \Gamma'$, $s \succ t, \Gamma$ und keine Gleichung aus $\Gamma \cup \Gamma'$ ist markiert.

Superposition links

$$\frac{\Gamma' \Rightarrow l \doteq r \qquad \Gamma, s \doteq t \Rightarrow \Delta}{\Gamma', \Gamma, s[r]_\pi = t \Rightarrow \Delta}$$

falls $s/\pi \equiv l$, $l \succ r, \Gamma'$, $s \succ t$ und keine Gleichung aus Γ' ist markiert, $s = t$ ist markiert.

Reflexiv-Resolution

$$\frac{\Gamma, s = s \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \quad \text{falls } s = s \text{ markiert ist.}$$

Selektion negativer Gleichungen (3)

Beachte

Sei $C \equiv \Gamma$, $s = t \Rightarrow \Delta$ und sei $s = t$ markiert.

a) Man kann nur unbedingte Gleichungen $\Rightarrow l = r$ benutzen, um in $\Gamma \Rightarrow s = t$ die Bedingung Γ leer zu machen (Superposition links).

b) Man kann nur mit unbedingten Gleichungen $\Rightarrow l = r$ unbedingte Gleichungen $\Rightarrow s = t$ behandeln (Superposition rechts).

Satz 4.9.

Das Inferenzsystem $\mathcal{GH}'$ ist korrekt und vollständig:

$S \overset{*}{\underset{\mathcal{GH}'}{\vdash}} \Box$ gdw S ist H-unerfüllbar.

Beweis-Skizze:

Analog zum letzten Satz.

Abwandlung der Modellkonstruktion so, dass Klauseln mit einer markierten Gleichung keine Regel erzeugen.

Horn-Klauseln mit Variablen

Voraussetzung

Bisher: S ist Menge von Grund-Horn-Klauseln.

Jetzt: S ist Menge von Horn-Klauseln.

Beachte

Wir untersuchen die H-Erfüllbarkeit von S. Dies ist äquivalent zur H-Erfüllbarkeit von $\text{gnd}(S) = \{\sigma(C) \mid C \in S, \quad \sigma(C) \text{ eine Grundklausel}\}$.
Also: $\Gamma \Rightarrow \Delta$ steht für $\{\sigma(\Gamma) \Rightarrow \sigma(\Delta) \mid \sigma \text{ Grundsubstitution}\}$

Ansatz für ein Inferenzsystem

Reduktion auf den bisherigen Fall mit dem Lifting-Argument.

Kontrolle der Ordnungsbedingungen $l \succ r, \Gamma$ durch Constraints.

Constraints (1)

Definition 4.18.

Ein **Constraint** T ist eine Boole'sche Kombination von atomaren Constraints der Form $s = t, s > t$ oder $s \geq t$. Eine Substitution σ erfüllt den Constraint T (oder ist Lösung von T), falls $\sigma(T)$ zu true ausgewertet wird, unter der Interpretation von $=, >, \geq$ durch $\equiv, \succ, \succeq$.

1. Bemerkung

Ein Gleichheits-Constraint $s_1 = t_1 \wedge \cdots \wedge s_n = t_n$ entspricht dem Unifikationsproblem $s_1 =^? t_1, \ldots, s_n =^? t_n$. Ein Ordnungs-Constraint $s > t$ entspricht den richtbaren Instanzen $\sigma(s) \succ \sigma(t)$.

2. Bemerkung

Bei der Vervollständigung ohne Abbruch benutzt man Constraints zur Beschreibung von kritischen Paaren: $s = t$ und $l = r$ liefern $\langle t, s[r]_\pi \rangle \mid s/\pi = l \wedge l > r \wedge s > t$.

3. Bemerkung

Eine **beschränkte Klausel** $C|T$ steht für alle Grundinstanzen $\sigma(C)$ mit σ erfüllt T. Sie ist **überflüssig**, falls T unlösbar ist.

Constraints (2)

4. Bemerkung

Constraints kann man vereinfachen:

a) Gilt $s \succ t$, so ist (1) jedes σ eine Lösung zu $s > t$ und (2) $t > s$ ist unlösbar.
b) Sei $T \equiv T_1 \wedge T_2$. Ist T_2 unlösbar, so ist auch T unlösbar. Ist jedes σ eine Lösung von T_2, so ist T äquivalent zu T_1.
c) Sei $T \equiv T_1 \wedge s = t$. Existiert $\mu = \text{mgu}(s,t)$ nicht, so ist T unlösbar. Sonst ist T äquivalent zu $\mu(T_1)$. Dann ist $C|T$ äquivalent zu $\mu(C)|\mu(T_1)$.

Beispiel

Sei $T \equiv x = g(y) \wedge x = z \wedge y \geq h(z) \wedge x > a$.

Es ist
$\mu = \{x \leftarrow g(y), z \leftarrow g(y)\} = \text{mgu}\{(x, g(y)), (x, z)\}$.

Dann ist T äquivalent zu $y \geq h(g(y)) \wedge g(y) > a$.

Dies ist unlösbar wegen $y \geq h(g(y))$.

Also ist $C|T$ überflüssig.

Das Inferenzsystem $\mathcal{H}$

Das System $\mathcal{H} = \mathcal{H}_\succ$ arbeitet auf Horn-Klauseln und mit fester Ordnung $\succ$. Wir betrachten gleich die Verfeinerung mit Selektion.

Definition 4.19.

$\mathcal{H}$ besteht aus den Inferenzregeln

Superposition rechts

$$\frac{\Gamma' \Rightarrow l \doteq r \qquad \Gamma \Rightarrow s \doteq t}{\Gamma', \Gamma \Rightarrow s[r]_\pi = t \mid s/\pi = l \wedge l > r, \Gamma' \wedge s > t, \Gamma}$$

falls s/π keine Variable, und keine Gleichung aus $\Gamma \cup \Gamma'$ markiert ist.

Superposition links

$$\frac{\Gamma' \Rightarrow l \doteq r \qquad \Gamma, s \doteq t \Rightarrow \Delta}{\Gamma', \Gamma, s[r]_\pi = t \Rightarrow \Delta \mid s/\pi = l \wedge l > r, \Gamma' \wedge s > t}$$

falls s/π keine Variable und $s = t$ markiert, und keine Gleichung aus Γ' markiert ist.

Reflexiv-Resolution

$$\frac{\Gamma, s = t \Rightarrow \Delta}{\Gamma \Rightarrow \Delta \mid s = t}$$ falls $s = t$ markiert ist.

Bemerkung

Die Prämissen sind o.B.d.A. Variablen-disjunkt.

Beispiel zu $\mathcal{H}$

Sei

$\succ$ die LPO zur Präzedenz $h >_F a >_F f >_F g >_F b$.

Betrachte den Inferenzschritt mit „Superposition rechts“

$$\frac{\Rightarrow f(a,x) = f(x,x) \qquad \Rightarrow h(f(a,g(y))) = h(y)}{\begin{array}{r}\Rightarrow h(f(x,x)) = h(y) \mid f(a,x) = f(a,g(y)) \\ \wedge h(f(a,g(y))) > h(y) \wedge f(a,x) > f(x,x)\end{array}}$$

Der Constraint ist äquivalent zu $x = g(y) \wedge a > x$.

Dies ist lösbar. Also entsteht neu
$\Rightarrow h(f(x,x)) = h(y) \mid x = g(y) \wedge a > x$

Hier wurden zur Constraint-Vereinfachung auch die Eigenschaften der LPO benutzt, etwa bei der Vereinfachung von $f(a,x) > f(x,x)$ zu $a > x$.

Beachte

Die neue Klausel lässt sich durch „Constraint-Propagieren“ weiter umschreiben zu

$$\Rightarrow h(f(g(y),g(y))) = h(y) \mid a > g(y)$$

Behandlung von beschränkten Klauseln

Beachte

Die Inferenzregeln erzeugen beschränkte Klauseln $C|T$, benutzen aber unbeschränkte Klauseln C als Prämissen.

Wir erweitern also $\mathcal{H}$ um Säuberungsregeln und identifizieren $C|T$ mit C, falls T leer ist.

Säuberungsregeln

(1) Lösche $C|T$, falls T unlösbar ist.

(2) Ersetze $C|T \wedge s = t$ durch $\mu(C)|\mu(T)$, falls $\mu = \text{mgu}(s,t)$.

(3) Ersetze $C|T$ durch C, falls T kein $s = t$ enthält, aber erfüllbar ist.

Bemerkung

Ist $\succ$ eine LPO, so ist entscheidbar (aber teuer), ob T lösbar ist.

C ist eine „zulässige Vergröberung“ von $C|T$: Jede Grundinstanz $\sigma(C)$ ist gültig (siehe das folgende Lemma).

Bezeichnung

Ein $\mathcal{H}$-Inferenzschritt besteht aus der Anwendung einer Inferenzregel aus $\mathcal{H}$ zur Erzeugung einer beschränkten Klausel $C|T$ und anschließender Säuberung von $C|T$ zur Klausel C'.

Korrektheit von $\mathcal{H}$

Lemma 4.12

Das Inferenzsystem $\mathcal{H}$ ist korrekt.

Gilt $S \vdash_{\mathcal{H}} S'$ und ist $\mathcal{A}$ ein H-Modell zu S, so ist $\mathcal{A}$ auch ein H-Models zu S'.

Beweis:

Es entstehe $C|T$ mit einer Inferenzregel aus $\mathcal{H}$ und den Prämissen $C_1, \ldots, C_n$ aus S für ein $n \in \{ 1, 2 \}$. Sei $T_=$ der Gleichheitsanteil von T und $\mu = mgu(T_=)$. Dann gilt $\mathcal{A} \models \sigma(C_i)$ für $i = 1, \ldots, n$ und $\sigma(C_i)$ ist eine Grundinstanz zu C_i. Dann gilt auch $\mathcal{A} \models \sigma(\mu(C))$ für alle Grundinstanzen $\sigma(\mu(C))$ von $\mu(C)$ (Übung!).

Wird $C|T$ zu $D = \mu(C)$ gesäubert, so gilt auch $\mathcal{A} \models \sigma(D)$ für alle Grundinstanzen $\sigma(D)$.

Also ist $\mathcal{A}$ auch ein H-Modell zu $S' = S \cup \{ D \}$.

Beachte Es werde $C|T$ zu D gesäubert. Dann gilt

$$\{ \sigma(C) \mid \sigma(C) \text{ ist Grundklausel, } \sigma \text{ erfüllt } T \} \subseteq \mathsf{gnd}(D).$$

Dies wird auch impliziet im nächsten Lemma benötigt.

Zur Vollständigkeit von $\mathcal{H}$ (1)

Lemma 4.13

Sei S eine Menge von Horn-Klauseln, sei S abgeschlossen bezüglich $\mathcal{H}$ und sei $\Box \notin S$. Dann ist R_S^* ein Modell zu S, d. h. $R_S^* \models C$ für alle $C \in S$.

Beweis:

Der Beweis läuft analog zum Beweis zu Lemma 4.11 mit dem Unterschied, dass wir Markierungen berücksichtigen müssen. Wir verwenden daher den Modellkonstruktor zum Inferenzsystem $\mathcal{GH}$ mit der Einschränkung, dass Klauseln mit markierten Gleichungen keine Regeln erzeugen. Für ihn bleibt Lemma 4.10 gültig (Übung).

Sei $R = R_{\mathsf{gnd}(S)}$ das Grund-Regelsystem, das durch $\mathsf{gnd}(S) = \{\sigma(C) \mid C \in S,\ \sigma(C) \text{ Grundklausel}\}$ erzeugt wird.

Induktion nach $\succ_c$ zeigt $R^* \models \sigma(C)$ für alle $C \in S$, σ Grundsubstitution:

Sei $C \equiv \Gamma \Rightarrow \Delta$. Es gibt 4 Fälle:

(I) $\Gamma = \emptyset$: Dann ist $C \equiv \ \Rightarrow s = t$.
  1) $\sigma(s) \equiv \sigma(t)$:
  Dann gilt $R^* \models \sigma(C)$ wie in 4.11a).
  2) $\sigma(s) \not\equiv \sigma(t)$: o.B.d.A sei $\sigma(s) \succ \sigma(t)$.
  Behandlung mit „Superposition rechts“ analog zu Fall (II,2)

Zur Vollständigkeit von $\mathcal{H}$ (2)

(II) $\Gamma \neq \emptyset$: Dann gibt es eine markierte Gleichung $s = t$ in Γ.

1) $\sigma(s) \equiv \sigma(t)$:
Behandlung mit „Reflexivresolution“ analog zu 4.11c)
2) $\sigma(s) \not\equiv \sigma(t)$: o.B.d.A sei $\sigma(s) \succ \sigma(t)$.
Behandlung mit „Superposition links“ analog zu 4.11d)

Wir betrachten den letzten Fall genauer:

Sei $C \equiv \Gamma', s = t \Rightarrow \Delta$ mit $\sigma(s) \succ \sigma(t)$ und $s = t$ markiert.

Annahme $R^* \not\models \sigma(C)$.

Dann gilt $R^* \models \sigma(\Gamma' \cup \{s = t\})$ und $R^* \not\models \sigma(\Delta)$, speziell $\sigma(s) \downarrow \sigma(t)$ in R. Es gibt $D \equiv \Gamma'' \Rightarrow l = r$ in S mit C, D sind Variablen-disjunkt und D erzeugt $\sigma(l) \to \sigma(r)$ und $\sigma(l) \to \sigma(r)$ reduziert $\sigma(s)$, also $\sigma(s)/\pi \equiv \sigma(l)$. Da D $\sigma(l) \to \sigma(r)$ erzeugt, gilt $R^* \models \sigma(D)$ und $\Gamma'' = \emptyset$.

i) s/π ist keine Variable, π Position in s.
Es gilt mit „Superposition links“

$$\frac{\Rightarrow l = r \qquad \Gamma', s = t \Rightarrow \Delta}{\Gamma', s[r]_\pi = t \Rightarrow \Delta \mid s/\pi = l \wedge l > r \wedge s > t}$$

und die Konklusion $D'|T$ erzeugt nach Säuberung D'' in S mit Grundinstanz $\sigma(D')$, da σ eine Lösung zu T ist. Dann gilt $\sigma(C) \succ_c \sigma(D')$ und daher $R^* \models \sigma(D')$ nach IV. Wegen $R^* \models \sigma(D)$ gilt doch $R^* \models \sigma(C)$.

Zur Vollständigkeit von $\mathcal{H}$ (3)

ii) $\sigma(s)/\pi$ ist Teilterm von $\sigma(x)$, $x \in \mathsf{Var}(s)$, $\pi = \pi' \cdot \pi''$ mit $s/\pi' \equiv x$.
Sei σ' Substitution mit $\sigma'(y) \equiv \sigma(y)$ für $y \neq x$, $\sigma'(x) \equiv \sigma(x)[\sigma(r)]_{\pi''}$. Wegen $R^* \models \sigma(l) = \sigma(r)$ gilt $R^* \models \sigma(y) = \sigma(y')$ für alle y. Hieraus folgt $R^* \models \sigma(C)$ gdw $R^* \models \sigma'(C)$.
Es gilt $\sigma(C) \succ_c \sigma'(C)$ und daher $R^* \models \sigma'(C)$ nach IV, also gilt doch $R^* \models \sigma(C)$.

Beachte Der Fall (II,2,ii) zeigt, dass nicht jede Grundinferenz in $\mathcal{H}$ liftbar ist. Wir zeigen später: Jede „nicht-redundante“ Grundinstanz ist liftbar.

Satz 4.10.

$\mathcal{H}$ ist korrekt und vollständig für Horn-Klauseln. Für jedes S gilt $S \overset{*}{\underset{\mathcal{H}}{\vdash}} \Box$ gdw S ist H-unerfüllbar.

Beweis: Wie bei Grund-Horn-Klauseln.

Bemerkung zu Constraints

Beachte

$\mathcal{H}$ produziert beschränkte Klauseln der Form $C|T$.

Wir haben $C|T$ zu C' „gesäubert".

Also: Wir arbeiten im wesentlichen mit (unbeschränkten) Klauseln.

Bemerkung

Man kann das Inferenzsystem $\mathcal{H}$ so erweitern, dass es auf beschränkten Klauseln arbeitet.

Dies beschränkt (theoretisch) den Suchraum bei der Suche von $S \overset{*}{\underset{\mathcal{H}}{\vdash}} \square$ weiter. Die Realisierung auf dem Rechner wird aber schwieriger:

a) Der Programmieraufwand wird größer.

b) Die Datenstrukturen werden komplizierter.

c) Das Lösen (bzw. Testen auf Lösbarkeit) von Constraints ist sehr zeitaufwendig.

Also

Angabe von $\mathcal{H}$ ohne Constraints.

$\mathcal{H}$ ohne Constraints

Definition 4.20.

Das Inferenzsystem $\mathcal{H}' = \mathcal{H}'_{\succ}$ besteht aus den Regeln

Superposition rechts

$$\frac{\Gamma' \Rightarrow l \doteq r \qquad \Gamma \Rightarrow s \doteq t}{\sigma(\Gamma', \Gamma) \Rightarrow \sigma(s[r]_\pi) = \sigma(t)}$$

falls $s/\pi \notin V$, $\sigma = \text{mgu}(l, s/\pi)$, $\sigma(l) \not\preceq \sigma(r, \Gamma')$, $\sigma(s) \not\preceq \sigma(t, \Gamma)$ und keine Gleichung aus $\Gamma \cup \Gamma'$ ist markiert.

Superposition links

$$\frac{\Gamma' \Rightarrow l \doteq r \qquad \Gamma, s \doteq t \Rightarrow \Delta}{\sigma(\Gamma', \Gamma, s[r]_\pi = t) \Rightarrow \sigma(\Delta)}$$

falls $s/\pi \notin V$, $\sigma = \text{mgu}(l, s/\pi)$, $\sigma(l) \not\preceq \sigma(r, \Gamma')$, $\sigma(s) \not\preceq \sigma(t)$ und keine Gleichung aus $\Gamma \cup \Gamma'$ ist markiert, aber $s = t$ ist markiert.

$\mathcal{H}$ ohne Constraints (Forts.)

Reflexiv-Resolution

$$\frac{\Gamma, s = t \Rightarrow \Delta}{\sigma(\Gamma) \Rightarrow \sigma(\Delta)} \quad \text{falls } \sigma = \text{mgu}(s,t) \text{ und } s = t \text{ ist markiert.}$$

Satz 4.11.

$\mathcal{H}'$ ist korrekt und vollständig für Horn-Klausel-Mengen S:

Es gilt $S \overset{*}{\underset{\mathcal{H}'}{\vdash}} \Box \text{ gdw } S$ ist H-unerfüllbar.

4.3 Saturierung bis auf Redundanz

Ziel

Sei $\mathcal{K}$ ein abstrakter Kalkül, der auf Klauseln (nicht nur auf Horn-Klauseln) arbeitet.

a) Wir besprechen die zentralen Konzepte
   - Saturierung
   - Modellkonstruktion
   - Redundanz

b) Wir leiten hieraus Bedingungen für die Vollständigkeit von $\mathcal{K}$ ab.

Beachte

a) Wir suchen einen Kalkül, der
   1. sparsam ist, also wenig Inferenzschritte erzeugt und
   2. vollständig ist, also $S \overset{*}{\underset{\mathcal{K}}{\vdash}} \Box$ liefert, falls S unerfüllbar ist.

b) Wir instanziieren später $\mathcal{K}$ zum Superpositionskalkül für beliebige Klauseln.

c) Auch der Superpositionskalkül $\mathcal{H}_\succ$ für Horn-Klauseln ist (nach Verfeinerung) eine Instanz von $\mathcal{K}$.

Vorgehen

Erst Beschränkung auf Grundklauseln.

Dann Erweiterung auf beliebige Klauseln mit Lifting.

Erinnerung: Inferenzen und Modellfunktor

Sei S eine Menge von Grundklauseln,
$\succ_c$ eine totale Noethersche Ordnung auf den Grundklauseln.

Bezeichnungen

Eine **Inferenzregel** in $\mathcal{K}$ hat die Form

$$\frac{C_1 \dots C_n \quad C}{D} \quad \text{falls Anwendungsbedingung}$$

Sei S eine Klauselmenge. Eine $\mathcal{K}$-**Inferenz** zu S ist eine Instanz

$$\pi = \frac{C_1 \dots C_n \quad C}{D}$$

einer Inferenzregel aus $\mathcal{K}$ mit $\{C_1, \dots, C_n, C\} \subseteq S$ und $C \succ_c C_i$. Es heißt C die **Hauptprämisse**, C_i eine **Nebenprämisse** und D die **Konklusion**. Wir schreiben auch con$(\pi) = D$. Die $\mathcal{K}$-Inferenz heißt **monoton**, falls $C \succ_c D$ gilt. $\mathcal{K}$ ist monoton, falls jede $\mathcal{K}$-Inferenz monoton ist.

Sei $\mathcal{K}(S)$ die Menge aller $\mathcal{K}$-Inferenzen zu S gilt.

S heißt **abgeschlossen unter** $\mathcal{K}$, falls con$(\mathcal{K}(S)) \subseteq S$.

Modellfunktor

Ein Modellfunktor Φ ordnet jeder Klauselmenge S ein Herbrandmodell $\Phi(S) = I_S$ zu. (Es braucht I_S kein Modell von S zu sein.)

Redundanz

Sei

S eine Menge von Grundklauseln und $S_C = \{D \in S \mid C \succ_c D\}$.

Definition 4.21.

Eine Grundklausel C heißt **redundant bezüglich** S, falls $S_C \models C$ gilt. C ist **redundant in** S, falls außerdem $C \in S$ gilt.

$R(S) = \{C \mid C$ ist redundant bezüglich $S\}$ ist die Menge der redundanten Klauseln bezüglich S.

Eine $\mathcal{K}$-Inferenz zu S

$$\frac{C_1 \dots C_n \quad C}{D}$$

ist **redundant**, falls eine Prämisse redundant ist oder $S_C \models D$ gilt.

$\mathcal{K}_R(S)$ ist die Menge der redundanten $\mathcal{K}$-Inferenzen zu S.

Beachte

Ist C redundant in S, so gilt $S_C - R(S) \models C$.

Jedes Modell von $S - R(S)$ ist ein Modell von S.

Ist C redundant in S, so ist C kein minimales Gegenbeispiel zu I_S.

Saturiertheit bis auf Redundanz

Definition 4.22.

S heißt $\mathcal{K}$-**saturiert bis auf Redundanz**, falls $\mathcal{K}(S - R(S)) \subseteq \mathcal{K}_R(S)$ gilt, d. h., falls alle $\mathcal{K}$-Inferenzen mit nicht-redundanten Prämissen bezüglich S redundant sind.

Beachte

S heißt abgeschlossen unter $\mathcal{K}$, falls $\mathcal{K}(S) \subseteq S$ gilt. Die Forderung der Abgeschlossenheit ist sehr stark. Wir zeigen, dass die Saturiertheit bis auf Redundanz zur Herleitung der folgenden Ergebnisse ausreicht.

Lemma 4.14

Sei S $\mathcal{K}$-saturiert bis auf Redundanz. Dann ist auch $T = S - R(S)$ $\mathcal{K}$-saturiert bis auf Redundanz und jede $\mathcal{K}$-Inferenz zu T ist redundant.

Beweis:

Für jedes C und D gilt: Aus $S_C \models D$ folgt $T_C \models D$.

Sei $\dfrac{C_1 \ldots C_n \quad C}{D}$ eine $\mathcal{K}$-Inferenz zu T. Da S saturiert ist, gilt $S_C \models D$, also auch $T_C \models D$.

Hieraus folgt die Behauptung.

Verkleinerung von Gegenbeispielen

Beachte

Gilt $I_S \models S$, so ist S H-erfüllbar. Gilt $I_S \not\models S$, so gibt es ein $C \in S$ mit $I_S \not\models C$. Dann heißt C ein **Gegenbeispiel** zu I_S. Dann existiert auch ein (bezüglich $\succ_c$) minimales Gegenbeispiel zu I_S.

Definition 4.23.

$\mathcal{K}$ hat (bezüglich des Modellfunktors Φ) die **Verkleinerungseigenschaft für Gegenbeispiele**, falls gilt:

Ist S $\mathcal{K}$-saturiert bis auf Redundanz, $\Box \notin S$ und $C \in S$ ein minimales Gegenbeispiel für I_S, so gibt es eine $\mathcal{K}$-Inferenz mit Hauptprämisse C und Konklusion D mit $C \succ_c D$ und $I_S \not\models D$.

Beachte

Wir verlangen nur, dass $\mathcal{K}$ in S Gegenbeispiele verkleinert, wenn S saturiert ist.

Da $\succ_c$ Noethersch ist, sieht man leicht, dass ein S kein Gegenbeispiel enthalten kann, wenn zu jedem Gegenbeispiel $C \in S$ ein kleineres Gegenbeispiel $D \in S$ existiert.

Das Hauptresultat für Grundklauseln

Sei S eine Menge von Grundklauseln,
Φ ein beliebiger Modellfunktor.

Satz 4.12.

Hat $\mathcal{K}$ die Verkleinerungseigenschaft für Gegenbeispiele und ist S $\mathcal{K}$-saturiert bis auf Redundanz, so gilt:

S ist H-erfüllbar $\mathrm{gdw}\ \Box \notin S$.

Beachte

Sei $\mathcal{K}$ monoton und korrekt.

a) Ist π eine $\mathcal{K}$-Inferenz zu S mit Konklusion D, so ist π in $S \cup \{D\}$ redundant.

b) Ist $C \in S$ redundant, so bleibt C redundant, wenn man
- redundante Klauseln D aus S entfernt oder
- Klauseln D (mit $S \models D$) in S einfügt.

Also folgt aus Satz 4.12:

Es reicht, S so weit zu saturieren, dass für alle $\mathcal{K}$-Inferenzen π zu S mit nicht-redundanten Prämissen con(π) schon in S liegt oder aber redundant bezüglich S ist. Dann gilt: S ist H-erfüllbar $\mathrm{gdw}\ \Box \notin S$.

Beweis zu Satz 4.12

Beweis

Ist $\Box \in S$, so ist S unerfüllbar.

Sei $\Box \notin S$ und $T = S - R(S)$.

Für jedes H-Modell I gilt $I \models S \ \mathrm{gdw}\ I \models T$.

Wir zeigen $I_T \models T$, dann gilt $I_T \models S$ und S ist H-erfüllbar.

Annahme $I_T \not\models T$:

Es gibt ein minimales $C \in T$ mit $I_T \not\models C$.

T ist nach Lemma 4.14 saturiert bis auf Redundanz und $\mathcal{K}$ hat die Verkleinerungseigenschaft für Gegenbeispiele, also gibt es eine $\mathcal{K}$-Inferenz π zu T mit

$$\pi = \frac{C_1 \ldots C_n \quad C}{D} \quad , C \succ_c D, I_T \not\models D \qquad (*)$$

Nach Lemma 4.14 ist π redundant, wegen $T \cap R(T) = \emptyset$ gilt also $T_C \models D$.

Nach Wahl von C gilt $I_T \models T_C$.

Aus $I_T \models T_C$ und $T_C \models D$ folgt $I_T \models D$. $\lightning$ zu (*)

Ableitungen mit Redundanz

Definition 4.24.

$S \overset{0}{\vdash} S'$ gdw S' entsteht aus S durch

a) Hinzunahme von Grundklauseln C mit $S \models C$ oder

b) Löschen von Grundklauseln C, die in S redundant sind.

$(S_i)_{i \in \mathbb{N}}$ ist eine **Ableitung** (mit Redundanz), falls $S_i \overset{0}{\vdash} S_{i+1}$ für alle $i \in \mathbb{N}$ gilt.

Setze

$$S^\infty = \bigcup_{i \in \mathbb{N}} S_i \qquad S_\infty = \bigcup_{i \in \mathbb{N}} \bigcap_{j \geq i} S_j$$

Es ist S_∞ die Menge der **persistenten** Klauseln.

Lemma 4.15

Sei $(S_i)_{i \in \mathbb{N}}$ eine Ableitung. Dann gilt

a) $R(S_i) \subseteq R(S^\infty) = R(S_\infty)$.

b) S_∞ ist erfüllbar gdw S_0 ist erfüllbar.

Also

Redundanz und (Un-)Erfüllbarkeit bleiben während der ganzen Ableitung erhalten und übertragen sich auch auf das Grenzsystem S_∞.

Fairness (1)

Voraussetzung ab jetzt

$\mathcal{K}$ ist monoton und korrekt: Für jede $\mathcal{K}$-Inferenz $\dfrac{C_1 \ldots C_n \quad C}{D}$ gilt $C \succ_c D$ und $\{C_1, \ldots, C_n, C\} \models D$.

Definition 4.25.

Eine Ableitung $(S_i)_{i \in \mathbb{N}}$ ist $\mathcal{K}$-**fair**, falls jede $\mathcal{K}$-Inferenz mit nicht-redundanten Prämissen in S_∞ redundant in S^∞ ist, d. h., falls $\mathcal{K}(S_\infty - R(S_\infty)) \subseteq \mathcal{K}_R(S^\infty)$ gilt.

Lemma 4.16

Eine Ableitung $(S_i)_{i \in \mathbb{N}}$ ist $\mathcal{K}$-fair, falls die Konklusion jeder nicht-redundanten $\mathcal{K}$-Inferenz mit nicht-redundanten Prämissen in S_∞ in S^∞ liegt oder in S^∞ redundant ist, d. h., falls

$\text{con}(\mathcal{K}(S_\infty - R(S_\infty)) - \mathcal{K}_R(S_\infty - R(S_\infty))) \subseteq S^\infty \cup R(S^\infty)$

Also

$\mathcal{K}$-Fairness kann erreicht werden, indem man garantiert, dass alle (nicht-redundanten) $\mathcal{K}$-Inferenzen π mit Prämissen aus S_∞ gezogen werden, also $\text{con}(\pi) \in S^\infty \cup R(S^\infty)$ gilt.

Fairness (2)

Sei $\mathcal{K}$ korrekt und monoton.

Lemma 4.17

Sei $(S_i)_{i \in \mathbb{N}}$ eine $\mathcal{K}$-faire Ableitung. Dann ist das Grenzsystem S_∞ $\mathcal{K}$-saturiert bis auf Redundanz.

Satz 4.13.

$\mathcal{K}$ habe die Verkleinerungseigenschaft für Gegenbeispiele und $(S_i)_{i \in \mathbb{N}}$ sei eine $\mathcal{K}$-faire Ableitung.
Dann gilt: S_0 ist unerfüllbar $\mathrm{gdw}\ \square \in S_\infty$.

Satz 4.14.

Hat $\mathcal{K}$ die Verkleinerungseigenschaft für Gegenbeispiele, so ist $\mathcal{K}$ vollständig.

Beweis:

Sei $(S_i)_{i \in \mathbb{N}}$ eine faire Ableitung mit $S_0 = S$. Ist S unerfüllbar, so gilt $\square \in S_\infty$, also auch $\square \in S_n$ für ein $n \in \mathbb{N}$. Also gilt $S \overset{*}{\underset{\mathcal{K}}{\vdash}} \square$.

Bemerkung zum Entwickeln von Inferenzsystemen (1)

Vorgehen

1. Man entwickelt häufig zunächst ein Inferenzsystem $\mathcal{GK}$, das auf Grundklauseln arbeitet und alle benötigten Eigenschaften hat: Korrektheit, Monotonie, Verkleinerung von Gegenbeispielen (bezüglich geeignetem Modellfunktor Φ).
   Dann gilt: Ist S' $\mathcal{GK}$-saturiert bis auf Redundanz so gilt
   S' ist H-erfüllbar $\mathrm{gdw}\ \square \notin S'$.
2. Man erweitert dann $\mathcal{GK}$ zu einem Inferenzsystem $\mathcal{K}$ auf Klauseln, so dass gilt
   - $\mathcal{K}$ und $\mathcal{GK}$ stimmen auf Grundklauseln überein.
   - Jede nicht-redundante $\mathcal{GK}$-Inferenz ist liftbar zu einer $\mathcal{K}$-Inferenz.

   Dann gilt: Ist S $\mathcal{K}$-saturiert bis auf Redundanz, so gilt
   S ist H-erfüllbar $\mathrm{gdw}\ \square \notin S$.
3. Man definiert die Fairness so, dass für jede faire $\mathcal{K}$-Ableitung $(S_i)_{i \in \mathbb{N}}$ gilt
   S_0 ist H-unerfüllbar $\mathrm{gdw}\ \square \in S_n$ für ein $n \in \mathbb{N}$.

Hier

Wir bezeichnen auch $\mathcal{GK}$ mit $\mathcal{K}$, fassen also $\mathcal{GK}$ als Restriktion von $\mathcal{K}$ auf Grundklauseln auf.

Bemerkung zum Entwickeln von Inferenzsystemen (2)

Sei S eine Menge von Klauseln.
gnd$(S) = \{\sigma(C) \mid C \in S, \sigma(C)$ Grundinstanz von $C\}$.

Beachte

S ist H-erfüllbar gdw gnd(S) ist H-erfüllbar.

Ziel

Anpassen (liften) der Begriffe, so dass gilt

1. Ist S saturiert bis auf Redundanz, so gilt:
   S ist H-erfüllbar gdw $\square \notin S$.
2. Ist $(S_i)_{i \in \mathbb{N}}$ eine faire $\mathcal{K}$-Ableitung, so gilt:
   S_0 ist H-unerfüllbar gdw $\square \in S_n$ für ein $n \in \mathbb{N}$.

Man muss folgende Begriffe liften/definieren:

Die Ordnung auf Klauseln
Redundanz einer Klausel bzw. einer Inferenz
Saturiertheit bis auf Redundanz
Eine Grundinferenz ist liftbar
$\mathcal{K}$ ist monoton bzw. korrekt

Es bleiben folgende Begriffe unverändert:

Φ ist ein Modellfunktor
$\mathcal{K}$ hat die Verkleinerungseigenschaft für Gegenbeispiele

Lifting: Die Ordnung auf Klauseln

Sei $\mathcal{K}$ ein Kalkül, der auf Klauseln (mit Variablen) arbeitet.

Sei

$\succ$ eine Reduktionsordnung auf den Termen, total auf den Grundtermen.

$\succ_l$ die induzierte Ordnung auf den Literalen:
comp$(s = t) = \{s, t\}$
comp$(\neg s = t) = \{s, s, t, t\}$
$l_1 \succ_l l_2$ gdw comp$(l_1) \succ\!\!\succ$ comp(l_2)

$\succ_c$ die induzierte Ordnung auf Klauseln:
comp$(C) = \{$comp$(l) \mid l$ in $C\}$
$C \succ_c D$ gdw comp$(C) \succ\!\!\succ_l$ comp(D)

Es gilt

Aus $s \succ t$ folgt $\sigma(s) \succ \sigma(t)$.

Aus $l_1 \succ_l l_2$ folgt $\sigma(l_1) \succ_l \sigma(l_2)$.

Aus $C \succ_c D$ folgt $\sigma(C) \succ_c \sigma(D)$.

Alle Ordnungen sind Noethersch und total auf den Grundobjekten.

Inferenzen und Grundinferenzen

Sei $\mathcal{K}$ ein Inferenzsystem auf Klauseln,
S eine Menge von Klauseln.

Bemerkung

Wir betrachten nur Inferenzsysteme $\mathcal{K}$ mit

(1) Jede $\mathcal{K}$-Inferenz π hat die Form $\pi = \dfrac{C_1 \ldots C_n \quad C}{\tau(D_0)}$,
also Konklusion $D \equiv \tau(D_0)$.

(2) Sind die Prämissen Grundklauseln, so ist π eine $\mathcal{GK}$-Inferenz.

Dann heißt $\mathcal{K}$ eine **Erweiterung** zu $\mathcal{GK}$.

Bezeichnung

Sei π eine $\mathcal{K}$-Inferenz zu S wie oben

a) Sind alle Klauseln in π Grundklauseln, so ist π eine $\mathcal{K}$-**Grundinferenz**.

b) $\sigma(\pi) :\equiv \dfrac{\sigma(\tau(C_1)) \ldots \sigma(\tau(C_n)) \quad \sigma(\tau(C))}{\sigma(\tau(D_0))}$
ist eine **Grundinstanz** zu π, falls alle Klauseln in $\sigma(\pi)$ Grundklauseln sind.

Beispiel zu Inferenzen

Sei $\mathcal{K} = \mathcal{R}_{\succ}$ der Resolutionskalkül,
$\succ$ eine LPO.

Die Resolutionsregel lautet

$$\frac{A \vee C \quad \neg B \vee D}{\tau(C \vee D)} \qquad \begin{array}{l} \text{falls } \tau = \text{mgu}(A, B), \\ \tau(A) \text{ ist strikt maximal in } \tau(A \vee C), \\ \tau(\neg B) \text{ ist maximal in } \tau(\neg B \vee D). \end{array}$$

Dann ist π eine $\mathcal{K}$-Inferenz:

$$\pi \equiv \frac{p(b,x) \quad \neg p(y,z) \vee \neg p(z,y)}{\tau(\neg p(z,y))} \qquad \tau = \{y \leftarrow b, z \leftarrow x\}$$

Beachte: $\tau(\neg p(z,y)) \equiv \neg p(x,b)$.

Sei $\sigma(x) \equiv t$. Für jeden Grundterm t ist $\sigma(\pi)$ korrekt

$$\sigma(\pi) \equiv \frac{p(b,t) \quad \neg p(b,t) \vee \neg p(t,b)}{\neg p(t,b)}$$

Für $b \succeq t$ ist $\neg p(b,t) \succeq_l \neg p(t,b)$ und daher $\sigma(\pi)$ eine $\mathcal{K}$-Inferenz.

Für $t \succ b$ ist nicht $\neg p(b,t) \succeq_l \neg p(t,b)$, also ist $\sigma(\pi)$ **keine** $\mathcal{K}$-Inferenz.

Lifting: Korrektheit und Monotonie von $\mathcal{K}$

Intention

a) π steht für alle Grundinstanzen $\sigma(\pi)$.
b) Nicht jedes $\sigma(\pi)$ ist eine $\mathcal{K}$-Grundinferenz, aber jedes $\sigma(\pi)$ ist korrekt.
c) Es sollen die Ergebnisse zu $\mathcal{GK}$ auf $\mathcal{K}$ übertragen werden.

Beachte

Die Intention verlangt für jede Klauselmenge S:

a) Jede $\mathcal{K}$-Inferenz π' zu gnd(S) soll monoton sein.
b) Jede nicht-redundante $\mathcal{K}$-Inferenz π' zu gnd(S) soll zu einer $\mathcal{K}$-Inferenz π zu S „liftbar" sein.
c) Für jede $\mathcal{K}$-Inferenz π ist jede Grundinstanz $\sigma(\pi)$ korrekt.

Definition 4.26.

Eine $\mathcal{K}$-Inferenz π heißt **korrekt**, falls alle Grundinstanzen $\sigma(\pi)$ korrekt sind.

$\mathcal{K}$ heißt **korrekt**, falls jede $\mathcal{K}$-Inferenz π korrekt ist.

$\mathcal{K}$ heißt **monoton**, falls jede $\mathcal{K}$-Grundinferenz monoton ist.

Lifting: Redundanz (1)

Sei $\mathcal{K}$ ein Inferenzsystem auf Klauseln,
S eine Menge von Klauseln.

Bezeichnug

$C \trianglerighteq D$, falls $\sigma(D) \equiv C$ für eine Substitution σ.
$C \approx D$, falls $C \trianglerighteq D$ und $D \trianglerighteq C$.
$C \triangleright D$, falls $C \trianglerighteq D$ und nicht $D \trianglerighteq C$.

Beachte

a) $\triangleright$ ist eine Noethersche Partialordnung auf Klauseln.
b) Aus $C \trianglerighteq D$ folgt gnd$(C) \subseteq$ gnd(D).
c) Aus $C \approx D$ folgt gnd$(C) =$ gnd(D) und dass C und D bis auf Variablenumbenennung übereinstimmen. Wir identifizieren dann C und D.

Definition 4.27.

a) C ist **redundant** bezüglich S, falls jedes $C' \in$ gnd(C) bezüglich gnd(S) redundant ist oder es ein $D \in S$ gibt mit $C \triangleright D$.
Sei $R(S) = \{C \mid C$ ist bezüglich S redundant $\}$.
b) Eine $\mathcal{K}$-Inferenz π zu S ist **redundant**, falls jede Grundinstanz $\sigma(\pi)$, die eine $\mathcal{K}$-Inferenz ist, redundant ist.

Lifting: Redundanz (2)

Lemma 4.18

a) Ist C redundant bezüglich S, so ist C auch redundant bezüglich $S - \{C\}$.

b) Für $S' = \text{gnd}(S)$ gilt $S' - R(S') \subseteq \text{gnd}(S - R(S))$. Also: Jede nicht-redundante Klausel in $\text{gnd}(S)$ ist Instanz einer nicht-redundanten Klausel in S.

Beweis:

a) Ist $C \vartriangleright D$ für ein $D \in S$, so ist $D \in S - \{C\}$. Sei also jedes $C' \in \text{gnd}(C)$ redundant bezüglich $\text{gnd}(S)$. Induktion nach $\succ_c$ zeigt, dass dann jedes $C' \in \text{gnd}(C)$ redundant bezüglich $\text{gnd}(S - \{C\})$ ist.

b) Sei $T' = S' - R(S')$ und $T = S - R(S)$. Sei $C' \in T'$, also $C' \in \text{gnd}(C)$ für ein $C \in S$.
Sei $C \vartriangleright D$ für kein $D \in S$. Wegen $C' \notin R(S')$ ist $C' \notin R(S)$.
Sei $C \vartriangleright D$ für ein $D \in S$. Dann gibt es ein $\vartriangleright$-minimales $D_1 \in S$ mit $C \vartriangleright D_1$. Dann ist $C' \in \text{gnd}(C) \subseteq \text{gnd}(D_1)$ und $D_1 \vartriangleright D_0$ für kein $D_0 \in S$. Wegen $C' \notin R(S')$ ist $D_1 \notin R(S)$.

Beispiele zur Redundanz

Beispiel 1

Ist $C \in S$ und τ eine beliebige Substitution und $D \not\equiv \Box$, so ist $D \vee \tau(C)$ redundant bezüglich S.

Beweis: Sei σ eine Grundsubstitution. Es ist $\sigma \circ \tau(C) \in \text{gnd}(S)$, $\sigma(D \vee \tau(C)) \succ_c \sigma \circ \tau(C)$ und $\sigma \circ \tau(C) \models \sigma(D \vee \tau(C))$. Also ist $\sigma(D \vee \tau(C))$ redundant in $\text{gnd}(S)$ und daher $D \vee \tau(C)$ redundant in S.

Beispiel 2

S enthalte die Unit-Klauseln $g(a) = a$ und $g(x) = f(x)$. Sei $g(x) \succ f(x)$.
Sei

$$\pi_1 = \quad \frac{g(a) = a \quad g(g(a)) = f(g(a))}{g(a) = f(g(a))}$$

$$\pi_2 = \quad \frac{g(a) = a \quad g(a) = f(a)}{a = f(a)}$$

Beide Inferenzen sind korrekt.

π_1 ist redundant, denn $g(a) = a,\ g(a) = f(a) \models g(a) = f(g(a))$, also $\text{gnd}(S)_{g^2(a) = f(g(a))} \models g(a) = f(g(a))$.

π_2 ist i. Allg. nicht redundant. Gilt aber $a = f(a) \in \text{gnd}(S)$, so ist π_2 redundant, denn $g(a) = f(a) \succ_c a = f(a)$.

Liftbare Inferenzen

Definition 4.28.

Eine $\mathcal{K}$-Inferenz π' zu gnd(S) heißt **liftbar**, falls es eine $\mathcal{K}$-Inferenz π zu S gibt, so dass π' eine Grundinstanz zu π ist. Wir sagen dann: π' ist liftbar zu π.

$\mathcal{K}$ heißt **liftend**, falls für jedes S gilt: Jede nicht-redundante $\mathcal{K}$-Inferenz π' zu gnd(S) ist liftbar zu einer $\mathcal{K}$-Inferenz π zu S mit nicht-redundanten Prämissen.

Beispiel 2 (Fortsetzung)

Sei $\pi \equiv \quad \dfrac{g(a) = a \quad g(x) = f(x)}{\tau(a = f(x))} \quad \tau(x) = a$

Es sind π_1 und π_2 $\mathcal{K}$-Inferenzen zu gnd(S).

π_1 ist nicht liftbar zu π.

π_2 ist liftbar zu π, es ist $\pi_2 = \sigma(\pi)$ mit $\sigma = $ id.

Beachte, dass π_1 redundant ist und dass nur nicht-redundante Grundinferenzen liftbar sein sollen.

Lifting: $\mathcal{K}$-Saturiertheit

Sei S eine Menge von Klauseln.

Definition 4.29.

S heißt $\mathcal{K}$-**saturiert bis auf Redundanz**, falls $\mathcal{K}(S - R(S)) \subseteq \mathcal{K}_R(S)$ gilt.

Lemma 4.19

Ist $\mathcal{K}$ liftend und S $\mathcal{K}$-saturiert bis auf Redundanz, so ist auch $S' = $ gnd(S) $\mathcal{K}$-saturiert bis auf Redundanz.

Beweis:
Zu zeigen ist $\mathcal{K}(S' - R(S')) \subseteq \mathcal{K}_R(S')$.
Annahme: Es gibt $\pi' \in \mathcal{K}(S' - R(S'))$, $\pi' \notin \mathcal{K}_R(S')$.
Da π' nicht redundant und $\mathcal{K}$ liftend ist, ist π' zu einer $\mathcal{K}$-Inferenz π mit nicht-redundanten Prämissen liftbar. Es gibt also eine $\mathcal{K}$-Inferenz zu S, so dass π' eine Grundinstanz zu π ist: $\pi' = \sigma(\pi)$.

$$\pi' = \quad \frac{C_1' \ldots C_n' C'}{D'} \qquad \pi = \quad \frac{C_1 \ldots C_n C}{\tau(D_0)}$$

mit $C_i' \equiv \sigma(\tau(C_i))$, $C' \equiv \sigma(\tau(C))$, $D' \equiv \sigma(\tau(D_0))$ und $C_i, C \in S - R(S)$. Da S $\mathcal{K}$-saturiert ist, gilt $\pi \in \mathcal{K}_R(S)$.
Also ist $\pi' = \sigma(\pi) \in \mathcal{K}_R(S')$ nach Definition 4.27b). $\lightning$

Das Hauptresultat für Klauseln

Sei S eine Menge von Klauseln.
Φ ein Modellfunktor.
$\mathcal{K}$ ein Inferenzsystem auf Klauseln.
$\mathcal{K}(S)$ (bzw. $\mathcal{K}_R(S)$) die Menge der (redundanten) $\mathcal{K}$-Inferenzen zu S.

Satz 4.15.

Hat $\mathcal{K}$ (bezüglich Φ) die Verkleinerungseigenschaft für Gegenbeispiele, ist $\mathcal{K}$ liftend und ist S $\mathcal{K}$-saturiert bis auf Redundanz, so gilt

S ist H-erfüllbar gdw $\Box \notin S$.

Beweis:
Aus den Voraussetzungen folgt nach Lemma 4.19, dass gnd(S) bis auf Redundanz $\mathcal{K}$-saturiert ist.

Also gilt mit Satz 4.12:
S ist H-erfüllbar gdw gnd(S) ist H-erfüllbar
gdw $\Box \notin$ gnd(S) gdw $\Box \notin S$.

Beachte

Nur nicht-redundante $\mathcal{K}$-Grundinferenzen π' müssen zu einer $\mathcal{K}$-Inferenz π mit nicht-redundanten Prämissen liftbar sein.

$\mathcal{K}$-Ableitungen auf Klauseln

Bemerkung

Nachdem der Redundanzbegriff auf Klauseln geliftet ist, ist klar, was eine Ableitung $(S_i)_{i\in\mathbb{N}}$ mit Redundanz ist:

Für alle $i \in \mathbb{N}$ gilt (i) $S_{i+1} = S_i \cup \{D\}$ mit $S_i \models D$ oder
(ii) $S_{i+1} = S_i - \{C\}$ mit $C \in R(S_i)$.

Ist $\mathcal{K}$ korrekt und ist D die Konklusion einer $\mathcal{K}$-Inferenz zu S_i, so gilt $S_i \models D$. Wir sprechen dann von einem $\mathcal{K}$-Schritt der Ableitung.

Seien S_∞ und S^∞ erklärt wie früher.

Lemma 4.20

Sei $(S_i)_{i\in\mathbb{N}}$ eine $\mathcal{K}$-Ableitung und sei $S'_i = $ gnd(S_i). Dann gilt für alle $i \in \mathbb{N}$

a) $R(S_0) \subseteq R(S_i) \subseteq R(S^\infty) = R(S_\infty)$
$R(S'_0) \subseteq R(S'_i) \subseteq R(S'^\infty) = R(S'_\infty)$
b) S'_0 ist H-erfüllbar gdw S'_i ist H-erfüllbar für alle $i \in \mathbb{N}$
gdw S'_∞ ist H-erfüllbar.
c) S_0 ist H-erfüllbar gdw S_∞ ist H-erfüllbar.

Lifting: Fairness von $\mathcal{K}$-Ableitungen

Definition 4.30.

Eine $\mathcal{K}$-Ableitung $(S_i)_{i\in\mathbb{N}}$ auf Klauseln heißt **fair**, falls $\mathcal{K}(S_\infty - R(S_\infty)) \subseteq \mathcal{K}_R(S^\infty)$ gilt.

Lemma 4.21

Eine $\mathcal{K}$-Ableitung $(S_i)_{i\in\mathbb{N}}$ ist fair, falls $\mathcal{K}$ monoton ist und con$(\mathcal{K}(S_\infty - R(S_\infty)) - \mathcal{K}_R(S_\infty - R(S_\infty))) \subseteq S^\infty \cup R(S^\infty)$ gilt.

Ist $(S_i)_{i\in\mathbb{N}}$ eine faire $\mathcal{K}$-Ableitung, so ist S_∞ $\mathcal{K}$-saturiert bis auf Redundanz.

Beweis: Übung.

Satz 4.16.

$\mathcal{K}$ sei monoton, korrekt und liftend, $\mathcal{K}$ habe die Verkleinerungseigenschaft für Gegenbeispiele und $(S_i)_{i\in\mathbb{N}}$ sei eine faire $\mathcal{K}$-Ableitung auf Klauseln. Dann gilt: S_0 ist H-unerfüllbar gdw $\square \in S_n$ für ein $n \in \mathbb{N}$.

Beweis:

S_∞ ist $\mathcal{K}$-saturiert bis auf Redundanz.

Also gilt S_0 ist H-unerfüllbar gdw $\square \in S_\infty$ gdw $\square \in S_n$ für ein $n \in \mathbb{N}$.

Beispiel $\mathcal{K} = \mathcal{R}_\succ$

Sei $\mathcal{K} = \mathcal{R}_\succ$ der Resolutionskalkül aus Abschnitt 4.1,
Φ der Modellfunktor aus Abschnitt 4.1.

Beachte

Nach dem hier entwickelten Ansatz gilt:

Definition 4.6 definiert $\mathcal{GR}_\succ$ auf den Grundklauseln.
Definition 4.8 erweitert $\mathcal{GR}_\succ$ zu $\mathcal{R}_\succ$ auf den Klauseln.

Es gilt

a) In jeder Grundinferenz ist die Hauptprämisse (die rechte Prämisse) größer als die Nebenprämisse.
b) $\mathcal{R}_\succ$ ist monoton und korrekt.
c) $\mathcal{R}_\succ$ verkleinert Gegenbeispiele in jeder Menge S.
d) Jede $\mathcal{R}_\succ$-Grundinferenz ist liftbar. Nicht-liftbare Inferenzen treten nicht auf, da die Gleichheit nicht zugelassen ist.

Also erfüllt $\mathcal{R}_\succ$ die Voraussetzungen von Satz 4.16.

Beispiel $\mathcal{K} = \mathcal{H}_\succ$

Sei $\mathcal{K} = \mathcal{H}_\succ$ der Superpositionskalkül für Horn-Klauseln,
Φ der Modellfunktor aus Abschnitt 4.2.

Es gilt

a) In jeder Grundinferenz π ist die Hauptprämisse größer als die Nebenprämisse. $\mathcal{H}_\succ$ ist eine Erweiterung von $\mathcal{GH}_\succ$.
b) $\mathcal{H}_\succ$ ist monoton und korrekt.
c) $\mathcal{H}_\succ$ verkleinert Gegenbeispiele in jeder Menge S von Grundklauseln.
d) Jede nicht-redundante $\mathcal{H}_\succ$-Grundinferenz ist liftbar. Es gibt aber nicht-liftbare Grundinferenzen.

Also erfüllt $\mathcal{H}_\succ$ die Voraussetzungen von Satz 4.16.

4.4 Superposition für beliebige Klauseln

Sei

S eine Menge beliebiger Klauseln.

$\succ$ eine Reduktionsordnung, die total ist auf den Grundtermen.

$\succ_c$ die durch $\succ$ bestimmte Ordnung auf Klauseln wie stets.

Markierung (Selektion)

In jeder Klausel darf man beliebig maximal ein negatives Literal markieren.

Ziel

Angabe des Superpositionskalküls $\mathcal{S}$, der korrekt und vollständig ist.

Dabei Ausnutzung der Ergebnisse aus dem letzten Abschnitt.

Erinnerung an Ordnungen

Sei

$\succ$ eine Noethersche Ordnung auf Literalen, mit

1. Die Einschränkung auf die Grundliterale ist total.
2. Die Einschränkung auf die Terme ist eine Grundreduktionsordnung.

Die Komplexität von Literalen ist die Multimenge

$$\text{comp}(l) = \begin{cases} \{s, t\} & \text{falls } l \text{ ist } s = t \\ \{s, s, t, t\} & \text{falls } l \text{ ist } \neg s = t \end{cases}$$

Sei $l \succ_l l'$ gdw $\text{comp}(l) \succ\!\!\succ \text{comp}(l')$.

Die Komplexität einer Klausel C ist die Multimenge

$$\text{comp}(C) = \{\text{comp}(l) \mid l \text{ in } C\}$$

Sei $C \succ_c D$ gdw $\text{comp}(C) \succ\!\!\succ_l \text{comp}(D)$.

Beachte

Dies ist für Horn-Klauseln wie früher.
Aus $s \succ t$ und $s = t \succ_l \neg u = v$ folgt $s \succ u, v$.

Superposition auf Grundklauseln

Sei S eine Menge von Grundklauseln.

Intention

Der Superpositionskalkül lässt nur Inferenzen zwischen maximalen Literalen in Klauseln zu:

a) Superposition positiv (bzw. negativ):
Überlappung einer maximalen Gleichung in eine maximale (negierte) Gleichung.

b) Reflexiv-Resolution:
Resolution einer maximalen negierten Gleichung mit $x = x$.

c) Es gibt zwei Faktorisierungsregeln:
Sie arbeiten mit maximalen Literalen.

Weitere Einschränkung der Redundanz:

Man kann mit Markierungen von negativen Literalen (Selektion) arbeiten. In den Beispielen sind die markierten Literale unterstrichen.

Bezeichnung

Das Literal l heißt (**strikt**) **maximal** in $C \vee l$, falls $l \succeq_l l'$ (bzw. $l \succ_l l'$) für alle l' aus C gilt.

Der Superpositionskalkül $\mathcal{GS}$ (1)

Sei S Menge von Grundklauseln.

Definition 4.31.

$\mathcal{GS} = \mathcal{GS}_\succ$ besteht aus folgenden Inferenzregeln.

Positive Superposition

$$\frac{C \lor s \doteq t \quad D \lor u \doteq v}{C \lor D \lor u[t]_p = v}$$ falls $u/p \equiv s$ und (i)-(vi)

Negative Superposition

$$\frac{C \lor s \doteq t \quad D \lor \neg u \doteq v}{C \lor D \lor \neg u[t]_p = v}$$ falls $u/p \equiv s$ und (i)-(iii), (iv'), (v)

Reflexiv-Resolution

$$\frac{D \lor \neg u = v}{D}$$ falls $u \equiv v$ und (iv')

Ordnungsfaktorisierung

$$\frac{C \lor u = v \lor s = t}{C \lor s = t}$$ falls $\{s, t\} = \{u, v\}$ und (iii')

Gleichheitsfaktorisierung

$$\frac{C \lor u \doteq v \lor s \doteq t}{C \lor \neg v = t \lor u = t}$$ falls $u \equiv s$ und (ii), (iii') und (vi)

Der Superpositionskalkül $\mathcal{GS}$ (2)

Definition der Nebenbedingungen

(i) $s \succ t$.

(ii) $u \succ v$.

(iii) $s = t$ ist strikt maximal in $C \lor s = t$ und C enthält kein markiertes Literal.

(iii') $s = t$ ist maximal in $C \lor s = t$ und C enthält kein markiertes Literal.

(iv) $u = v$ ist strikt maximal in $D \lor u = v$ und D enthält kein markiertes Literal.

(iv') $\neg u = v$ ist markiert oder in dieser Prämisse ist nichts markiert und $\neg u = v$ ist maximal in $D \lor \neg u = v$.

(v) $p \in \mathrm{Pos}(u)$.

(vi) $u = v \succ_l s = t$.

Beispiele zu $\mathcal{GS}$

Beispiel 1 Nachbildung der Resolution

(1)	$s = t \vee C$	(3)	$\underline{\neg t = t} \vee C \vee D$
(2)	$\neg s = t \vee D$	(4)	$C \vee D$

Ist s der größte Term, so folgen (3) und (4) aus (1) und (2) in $\mathcal{GS}$. Wir lassen den Zwischenschritt (3) im Folgenden weg.

Beispiel 2

Sei $\succ$ die LPO zu $p >_F a >_F b$ und $S = \{(1), (2), (3)\}$.

Wir zeigen $S \overset{*}{\vdash} \Box$.

(1)	$\neg p(a) = \top \vee \neg p(b) = \top$	(4)	$\neg p(b) = \top$
(2)	$\neg p(a) = \top \vee p(b) = \top$	(5)	$p(b) = \top$
(3)	$p(a) = \top$	(6)	$\Box$

Beispiel 3 Bedeutung der Gleichheitsfaktorisierung

Sei $\succ$ die LPO zu $a >_F b >_F c$ und $S = \{(1), (2), (3)\}$.

(1)	$a = b \vee a = c$	(4)	$\underline{\neg b = c} \vee a = c$
(2)	$b = c$	(5)	$a = c$
(3)	$\neg a = c \vee \neg a = c$	(6)	$\neg a = c$
		(7)	$\Box$

Negative Superposition von (1) auf (3) liefert eine Tautologie. Sie liefert kein neues Wissen. Sonst ist auf S nur die Gleichheitsfaktorisierung anwendbar. Sie liefert (4) aus (1). Die restlichen Klauseln entstehen durch Resolution.

Korrektheit und Monotonie von $\mathcal{GS}$

Lemma 4.22

$\mathcal{GS}$ ist korrekt und monoton: Für jede Inferenz $\dfrac{C_1 \ldots C_n C}{D}$ gilt $\{C_1, \ldots C_n, C\} \models D$ und $C \succ_c D$ und $C \succ_c C_i$.

Beweis:

Wir zeigen dies beispielhaft für die positive Superposition

$$\frac{C \vee s = t \quad D \vee u = v}{D' :\equiv C \vee D \vee u[t]_p = v}$$

Es ist $s \succ t$ nach (i). Wegen (v) folgt daraus $u \succ u[t]_p$ und somit $u = v \succ_l u[t]_p = v$. Aus (vi) und (iii) folgt $u = v \succ_l l$ für alle l in $C \vee s = t$. Daher gilt $D \vee u = v \succ_c C \vee s = t$ und $D \vee u = v \succ_c D'$.

Sei I ein H-Modell mit $I \models C \vee s = t$ und $I \models D \vee u = v$. Gilt $I \models l_1$ oder $I \models l_2$ für ein $l_1 \in C$ bzw. $l_2 \in D$, so gilt $I \models D'$. Gilt $I \models l$ für kein $l \in C \vee D$, so gilt $I \models u = v \wedge s = t$. Dann gilt $I \models u[t]_p = v$, also auch $I \models D'$.

Analog zeigt man, dass auch die anderen Inferenzregeln korrekt und monoton sind.

Zur Vollständigkeit von $\mathcal{GS}$

Ziel

Vollständigkeitsergebnis aus letztem Abschnitt anwenden.

Ansatz

a) Angabe eines Modell-Funktors I.
   Sei S eine Menge von **Grundklauseln** mit $\Box \notin S$. Beschreibung des Modells I_S durch ein konvergentes Grundregelsystem.

b) Nachweis, dass $\mathcal{GS}$ hinreichend mächtig ist, also dass $\mathcal{GS}$ die Verkleinerungseigenschaft für Gegenbeispiele hat.

Dann

Abschnitt 4.3 liefert, dass $\mathcal{GS}$ vollständig ist für Grundklauselmengen.

Idee der Modellkonstruktion

Sei S Menge von Grundklauseln.

Definition 4.32.

Eine Grundklausel $C \vee s = t$ heißt **reduktiv** für $s \to t$, falls $s = t$ strikt maximal in der Klausel ist und $s \succ t$ gilt.

Bemerkung zur Konstruktion von $I_S = R_S^*$

R_S ist eine Menge von Regeln $s \to t$ mit $C \equiv C' \vee s = t$ in S.

R_S wird induktiv nach $\succ_c$ definiert.

R_S ist konvergent. Nicht jedes C produziert eine Regel.

Idee der Modellkonstruktion

Sei $C \equiv C' \vee s = t$ in S und reduktiv für $s = t$ und noch nicht $I_S \models C$. Es ist $C \succ_c C'$.

Sei $\tilde{C}$ die Konjunktion der negierten Literale in C'.

Dann entspricht C der positiv/negativ bedingten Regel $s \to t$ if $\tilde{C}$.

Nach Induktionsvoraussetzung ist schon definiert, ob I_S die Bedingung $\tilde{C}$ erfüllt oder nicht.

Es wird $s \to t$ in R_S aufgenommen, falls I_S die Bedingung $\tilde{C}$ erfüllt.

Definition der Modellkonstruktion

Sei S Menge von Grundklauseln.

Definition 4.33. (Definition des Modellfunktors Φ)

Sei $C \equiv C' \vee s = t$ und sei das Regelsystem R_C der durch die $D \in S$ mit $C \succ_c D$ produzierten Regeln schon definiert. Die Klausel C **produziert** die Regel $s \to t$, falls (1) bis (5) gilt:

(1) C ist reduktiv für $s \to t$,
(2) C ist ein Gegenbeispiel für R_C^*,
(3) s ist irreduzibel in R_C,
(4) C' ist ein Gegenbeispiel für $(R_C \cup \{s \to t\})^*$ und
(5) kein Literal in C ist markiert.

Sei $R_S = \{s \to t \mid C \in S, C \text{ produziert } s \to t\}$.
Es ist $\Phi(S) = I_S = R_S^*$.

Eigenschaften des Modellfunktors (1)

Sei S gegeben und $R = R_S$.

Lemma 4.23

R und alle R_C sind konfluent und terminierend. Es gilt $R_C \subseteq R$.

Lemma 4.24

Sei $C \equiv C' \vee s = t$ reduktiv für $s \to t$.

a) Für alle $l \to r$ in $R - R_C$ gilt $l \succeq s$.
b) Für $s \succ u, v$ gilt $R_C^* \models u = v \quad \text{gdw} \quad R^* \models u = v$.
c) Aus $R_C^* \models C$ folgt $R^* \models C$.

Beweis:

a) Wird $l \to r$ aus $R - R_C$ durch $D \equiv D' \vee l = r$ produziert, so gilt $D \succeq_c C$. Dann gilt $l \succeq s$.
b) Ist $s \succ w$, so ist w in $R - R_C$ irreduzibel nach a).
Für $s \succ u, v$ gilt also $R_C^* \models u = v \quad \text{gdw} \quad u \downarrow v$ in R_C
$\text{gdw} \quad u \downarrow v$ in $R \quad \text{gdw} \quad R^* \models u = v$.
c) Sei $C \equiv \Gamma \Rightarrow \Delta$. Dann ist $s \succ u', v'$ für alle $u' = v'$ in Γ. Aus $R_C^* \models C$ folgt $R_C^* \not\models \Gamma$ oder $R_C^* \models u = v$ für ein $u = v$ in Δ.
Gilt $R_C^* \not\models \Gamma$, so gilt $R^* \not\models \Gamma$ nach b), also $R^* \models C$.
Gilt $R_C^* \models u = v$ für ein $u = v$ aus Δ, so gilt $u \downarrow v$ in R_C, also $u \downarrow v$ in R,
also $R^* \models C$.

Eigenschaften des Modellfunktors (2)

Sei $C \equiv C' \vee s = t$ reduktiv für $s \to t$,
$R_0 = R_C \cup \{s \to t\}$ und s irreduzibel in R_C.

Beachte

C' kann kein Literal der Form $\neg s = v$, wohl aber eines der Form $s = v$ enthalten. Hierfür wird die Bedingung (4) in Definition 4.33 benötigt: $R_0^* \not\models C'$.

Lemma 4.25

a) Aus $R^* \models s = t$ und $R^* \models C'$ folgt $R_0^* \models C'$.
b) Gilt $R_0^* \models C'$ und nicht $R^* \models C$, so gibt es ein $s = v$ in C' mit $R^* \models t = v$.
c) Produziert C die Regel $s \to t$, so gilt $R^* \not\models C'$.

Beweis zu Lemma 4.25

R_0 ist konfluent und terminierend.
Für $s \succ u, v$ gilt wie in Lemma 4.24 b)

$$R_0^* \models u = v \text{ gdw } R_C^* \models u = v \text{ gdw } R^* \models u = v \qquad (+)$$

a) Es gelte $R^* \models s = t$ und $R^* \models C'$.
Gilt $R^* \models u = v$ für ein $u = v$ in C' mit $s \succ u, v$, so gilt $R_0^* \models C'$ nach $(+)$. Sonst gibt es ein $s = v$ in C' mit $R^* \models s = v$. Dann gilt $R^* \models t = v$ wegen $R^* \models s = t$, also $R_0^* \models t = v$ wegen $s \succ t, v$ und $(+)$, also $R_0^* \models s = v$.
Hieraus folgt $R_0^* \models C'$.

b) Wegen $R_0^* \models C'$ und $R^* \not\models C$ und $(+)$ gibt es $s = v$ in C' mit $R_0^* \models s = v$ und $R^* \not\models s = v$.
Es gilt dann $R_0^* \models t = v$, also auch $R^* \models t = v$ wegen $s \succ t, v$ und $(+)$.

c) Produziert C die Regel $s \to t$, so gilt $R_0^* \not\models C'$ und $s \to t$ liegt in R. Aus a) folgt jetzt $R^* \not\models C'$.

Vollständigkeit für Grundklauseln

Satz 4.17.

Sei S eine Menge von Grundklauseln und $\Box \notin S$. Sei $I_S = R^*$.

Ist $C \in S$ das kleinste Gegenbeispiel für I_S, so gibt es eine $\mathcal{GS}$-Inferenz in S mit Hauptprämisse C und Konklusion D mit

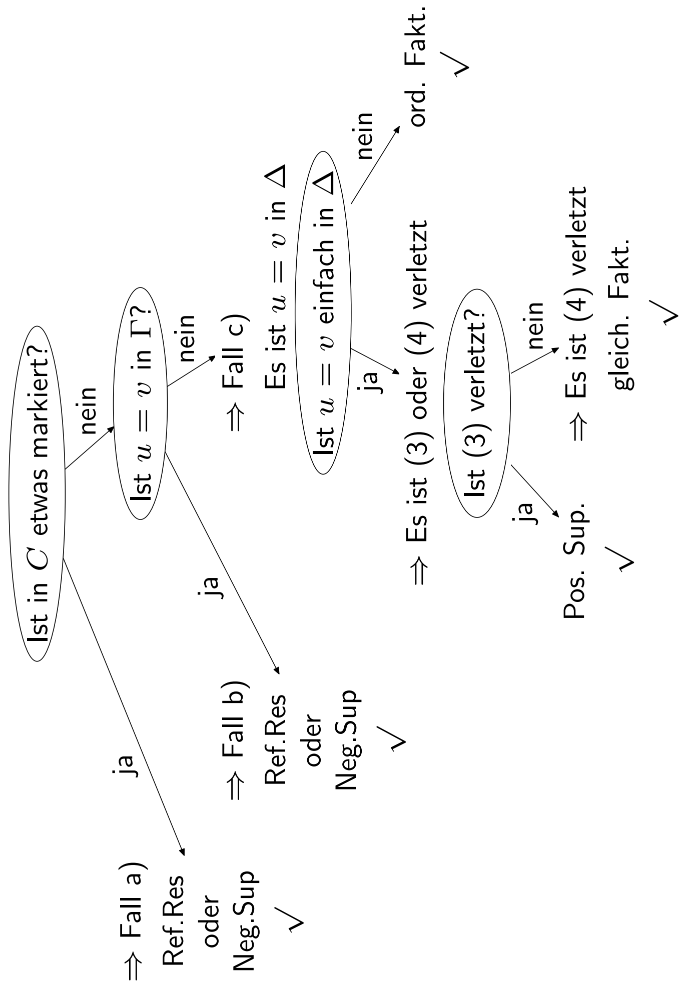
1. $C \succ_c D$ und D ist ein Gegenbeispiel für I_S.
2. Ist die Inferenz eine Superpositions-Inferenz, so ist die Nebenprämisse produktiv.

Also

$\mathcal{GS}$ hat die Verkleinerungseigenschaft für Gegenbeispiele.

Struktur des Beweises zu Satz 4.17

Sei $C \in S$ minimales Gegenbeispiel zu I_S, $C \equiv \Gamma \Rightarrow \Delta$ und $u = v$ das maximale Atom unter allen $l \in C$.



Beweis zu Satz 4.17 (1)

Sei $C \equiv \Gamma \Rightarrow \Delta$ und $u = v$ das Atom im größten Literal in C und $u \succeq v$.

Es gilt $R^* \not\models C$, also $R^* \models s = t$ für alle $s = t$ in Γ und $R^* \not\models s = t$ für alle $s = t$ in Δ.

a) Sei $\neg l = r$ aus C markiert und $l \succeq r$.

Gilt $l \equiv r$, so produziert „Ref.-Resolution" aus C ein D mit $C \succ_c D$ und $R^* \not\models D$.

Sei also $l \succ r$ und sei $C \equiv C' \vee \neg l = r$.

Wegen $l = r$ in Γ gilt $l \downarrow r$ in R, also ist l mit einer Regel $s \to t$ in R reduzierbar. Es werde $s \to t$ durch $C_1 \equiv C_0 \vee s = t$ produziert.

Wegen $l \equiv l[s]_p$ liefert „Neg. Superposition" aus C und C_1 die Klausel

$$D \equiv C' \vee C_0 \vee \neg l[t]_p = r.$$

Es gilt $R^* \not\models C'$ wegen $R^* \not\models C$.

Da C_1 $s \to t$ produziert, ist C_1 reduktiv für $s \to t$ und es gilt $R^* \models s = t$. Nach Lemma 4.25c) gilt $R^* \not\models C_0$.

$R^* \models l = r$ und $R^* \models s = t$ liefert $R^* \models l[t]_p = r$.

Also gilt $R^* \not\models D$. Es gilt $C \succ_c D$, da $\mathcal{GS}$ monoton ist.

Beweis zu Satz 4.17 (2)

b) Sei nichts in C markiert und $u = v$ in Γ.

Wie in a) mit $u = v$ statt $l = r$ liefert entweder „Ref.-Resolution" oder „Neg. Superposition" ein D mit $C \succ_c D$ und $R^* \not\models D$.

c) Sei nichts in C markiert und $u = v$ nicht in Γ.

Es ist $u = v$ in Δ, also $R^* \not\models u = v$, also $u \succ v$.

Tritt $u = v$ mehrfach in Δ auf, so liefert „Ord.-Faktorisierung" ein D mit $C \succ_c D$ und $R^* \not\models D$.

Es trete also $u = v$ nur einfach in Δ auf.

Dann ist $C \equiv C' \vee u = v$ reduktiv für $u \to v$.

Es ist $R_C^* \not\models C$ wegen $R^* \not\models C$ nach Lemma 4.24c).

Es ist C nicht produktiv wegen $R^* \not\models C$, also ist eine der Bedingungen (3) oder (4) aus Definition 4.33 verletzt.

Ist (3) verletzt, so ist u in R_C reduzibel. Dann liefert analog zu a) „Pos. Superposition" ein D mit $C \succ_c D$ und $R^* \not\models D$. (Beachte dazu $C \succ_c C_1$, also $u = v \succeq_l s = t$, und $R^* \models s = t$. Also gilt $u = v \succ_l s = t$ wegen $R^* \not\models C$.)

Sei (3) nicht verletzt, wohl aber (4), es gelte also $R_0^* \models C'$ mit $R_0 = R_C \cup \{u \to v\}$. Es gilt nach Lemma 4.25b)
$C \equiv C'' \vee u = t \vee u = v$ und $R^* \models v = t$.

„Gleich.-Faktorisierung" liefert $D \equiv C'' \vee \neg v = t \vee u = t$ und $C \succ_C D$. Es gilt $R^* \not\models C'' \vee u = t$ wegen $R^* \not\models C$. Es gilt $R^* \not\models \neg v = t$, wegen $R^* \models v = t$. Also gilt $R^* \not\models D$.

1. Beispiel

Ziel Demonstration von

a) der Verkleinerung von Gegenbeispielen.

b) der Bedeutung von (4) in Definition 4.33

Beachte

$s = t \vee C$ und $\neg s = t \vee D$ liefert in 2 $\mathcal{GS}$-Schritten unter den entsprechenden Ordnungsbedingungen $C \vee D$. Dies wird hier benutzt.

Sei $S = \{(1),(2),(3)\}$ und $a \succ b \succ c$.

S	1.Modell	2.Modell	3.Modell
$(1)\ b = c$	$1)\ b \to c$	$1)\ b \to c$	$1)\ b \to c$
$(2)\ a = b \vee a = c$	$2)\ -$	$3)\ \surd$	
$(3)\ \neg a = c$		$4)\ -$	
$(4)\ \neg b = c \vee a = c$		$2)\ a \to c$	
$(5)\ \neg b = c$			$2)\ -$
$(6)\ \Box$			

Hier steht $-$ für „die Modellkonstruktion erfüllt C nicht".
$\surd$ für „das bisher erzeugte Modell erfüllt schon C".

Also S ist unerfüllbar.

Beachte

Die Bedingung (4) aus Definition 4.33 verhindert, dass (2) produktiv ist. Daher ist (2) das kleinste Gegenbeispiel beim 1.Modell. Es kann nur mit einer Inferenzregel (mit (2) als Hauptprämisse) verkleinert werden: „Gleich.-Faktorisierung" liefert (4).

2. Beispiel

Sei $S = \{(1),(2), (3)\}$.
$\succ$ die LPO zu $p >_F q >_F \top$.
$s \neq t$ steht für $\neg s = t$.

Dann

Modellbildung + Saturierung liefern

S	1. Modell	2. Modell	3. Modell
$(1)\ p = \top \vee q = \top$	$1)\ p \to \top$	$2)\ \surd$	
$(2)\ p = \top \vee q \neq \top$	$2)\ \surd$	$3)\ p \to \top$	
$(3)\ p \neq \top$	$3)\ -$	$4)\ -$	
$(4)\ q = \top$		$1)\ q \to \top$	$1)\ q \to \top$
$(5)\ q \neq \top$			$2)\ -$
$(6)\ \Box$			

Also

S ist unerfüllbar.

Hinweis

Vergleiche dies mit Satz 4.17. In jedem Inferenzschritt wird aus dem kleinsten Gegenbeispiel zum aktuellen Modell I_S ein noch kleineres Gegenbeispiel erzeugt.

Der Beweisprozeß für Grundklauseln

Problem

Eingabe: S, eine Menge von Grundklauseln.
Frage: Ist S H-erfüllbar?

Ansatz

Saturiere S bezüglich dem Kalkül $\mathcal{GS}$.

Nutze dabei den Redundanzbegriff.

Satz 4.18.

Sei $(S_i)_{i\in\mathbb{N}}$ eine $\mathcal{GS}$-faire Ableitung mit $S_0 = S$.

Es gilt: S ist H-unerfüllbar $\mathrm{gdw}\ \Box \in S_n$ für ein $n \in \mathbb{N}$.

Beweis:

$\mathcal{GS}$ hat nach Satz 4.17 die Verkleinerungseigenschaft für Gegenbeispiele.

Also gilt nach Satz 4.13 für jede $\mathcal{GS}$-faire Ableitung

$$\begin{aligned} S_0 \text{ ist H-unerfüllbar } & \mathrm{gdw}\ \Box \in S_\infty \\ & \mathrm{gdw}\ \Box \in S_n \text{ für ein } n \in \mathbb{N}. \end{aligned}$$

Satz 4.19.

$\mathcal{GS}$ ist korrekt und vollständig für Grundklauseln.

Superposition auf Klauseln

Ziel

Liften des Superpositionskalküls $\mathcal{GS}$ auf Grundklauseln zum Superpositionskalkül $\mathcal{S}$ auf Klauseln.

Sei S eine Menge von Klauseln.

Liften der Inferenzregeln

Ansatz: Die Ordnungsbedingungen in der Anwendungsbedingung einer Regel so anpassen, dass jede Grundinferenz, die in $\mathcal{GS}$ gezogen werden muss, in $\mathcal{S}$ abgedeckt ist. Beachte, dass $\succ$ auf den Termen mit Variablen nicht total ist.

Beispiel

Sei $s' \equiv \sigma(s), t' \equiv \sigma(t)$ und in $\mathcal{GS}$ die Bedingung $s' \succ t'$ vorhanden. Gilt $t \succeq s$, so gilt $s' \succ t'$ sicher nicht.

Übersetzt man $s' \succ t'$ in $\mathcal{GS}$ zu $t \not\succeq s$ in $\mathcal{S}$, so sind sicher alle σ mit $\sigma(s) \succ \sigma(t)$ durch $t \not\succeq s$ abgedeckt.

Bezeichnung

Das Literal l heißt (**strikt**) **maximal** in $C \vee l$, falls $l' \not\succ_l l$ (bzw. $l' \not\succeq_l l$) für alle l' aus C gilt.

Der Superpositionskalkül $\mathcal{S}$ (1)

Definition 4.34.

$\mathcal{S} = \mathcal{S}_{\succ}$ besteht aus folgenden Inferenzregeln.

Positive Superposition

$$\frac{C \vee s \doteq t \quad D \vee u \doteq v}{\sigma(C \vee D \vee u[t]_p = v)} \qquad \text{falls } \sigma = \mathsf{mgu}(u/p, s) \text{ und (i)-(vi)}$$

Negative Superposition

$$\frac{C \vee s \doteq t \quad D \vee \neg u \doteq v}{\sigma(C \vee D \vee \neg u[t]_p = v)} \qquad \text{falls } \sigma = \mathsf{mgu}(u/p, s) \text{ und (i)-(iii), (iv'), (v)}$$

Reflexiv-Resolution

$$\frac{D \vee \neg u = v}{\sigma(D)} \qquad \text{falls } \sigma = \mathsf{mgu}(u, v) \text{ und (iv')}$$

Ordnungsfaktorisierung

$$\frac{C \vee u \doteq v \vee s = t}{\sigma(C \vee s = t)} \qquad \text{falls } \sigma = \mathsf{mgu}(u = v, s = t) \text{ und (iii')}$$

Gleichheitsfaktorisierung

$$\frac{C \vee u \doteq v \vee s \doteq t}{\sigma(C \vee \neg v = t \vee u = t)} \qquad \text{falls } \sigma = \mathsf{mgu}(u, s) \text{ und (ii), (iii') und (vi)}$$

Der Superpositionskalkül (2)

Definition der Nebenbedingungen

(i) $\sigma(t) \not\succeq \sigma(s)$.

(ii) $\sigma(v) \not\succeq \sigma(u)$.

(iii) $\sigma(s = t)$ ist strikt maximal in $\sigma(C \vee s = t)$ und C enthält kein markiertes Literal.

(iii') $\sigma(s = t)$ ist maximal in $\sigma(C \vee s = t)$ und C enthält kein markiertes Literal.

(iv) $\sigma(u = v)$ ist strikt maximal in $\sigma(D \vee u = v)$ und D enthält kein markiertes Literal.

(iv') $\neg u = v$ ist markiert oder in dieser Prämisse ist nichts markiert und $\sigma(\neg u = v)$ ist maximal in $\sigma(D \vee \neg u = v)$.

(v) $p \in \mathsf{Pos}(u)$ und $u/p \notin V$.

(vi) $\sigma(s = t) \not\succeq_l \sigma(u = v)$.

Voraussetzungen wie stets:

Die Prämissen sind Variablen-disjunkt.

Beispiele (1)

Sei S gegeben durch

(1)	$\neg p(x) = \top \vee \neg p(y) = \top \vee \neg g(x,y) = g(y,x)$	
(2)	$p(x) = \top \vee p(y) = \top \vee q(x,y,z) = \top$	
(3)	$\neg q(x,y,c) = \top$	

Sei $>$ die LPO mit $g >_F q >_F p >_F p >_F \top >_F c$. Dies liefert

(4)	$\neg p(x) = \top \vee \neg p(x) = \top$	Ref.Res (1)
(5)	$p(x) = \top \vee p(y) = \top \vee \neg\top = \top$	Neg.Sup (2,3)
(6)	$p(x) = \top \vee p(y) = \top$	Ref.Res (5)
(7)	$\neg\top = \top \vee p(z) = \top$	Gleich.Fak (6)
(8)	$p(z) = \top$	Ref.Res (7)
(9)	$\neg\top = \top \vee \neg p(x) = \top$	Neg.Sup (8,4)
(10)	$\neg p(x) = \top$	Ref.Res (9)
(11)	$\neg\top = \top$	Neg.Sup (8,10)
(12)	$\Box$	Ref.Res(11)

Beachte: Ohne die Regel „Gleichheitsfaktorisierung“ ist $\Box$ nicht aus S herleitbar.

Beispiele (2)

Sei $S = \{(1),(2)\}$ und $>$ die LPO zu $g >_F p >_F \top$. Dies liefert

(1)	$p(x) = \top \vee p(y) = \top$	
(2)	$\neg p(x) = \top \vee \neg p(y) = \top \vee$ $\neg g(x,y) = g(y,x)$	
(3)	$\neg p(x) = \top \vee \neg p(x) = \top$	Ref.Res (2)
(4)	$p(y) = \top$	Ord.Fakt (1)
(5)	$\neg\top = \top \vee \neg p(x) = \top$	Neg.Sup (4,3)
(6)	$\neg p(x) = \top$	Ref.Res (5)
(7)	$\neg\top = \top$	Neg.Sup (4,6)
(8)	$\Box$	Ref.Res (7)

Übung: Man zeige, dass ohne die Regel „Ordnungsfaktorisierung“ $\Box$ aus S nicht herleitbar ist.

Korrektheit und Monotonie von $\mathcal{S}$

Bemerkung

Eine Grundinstanz $\sigma(\pi)$ einer $\mathcal{S}$-Inferenz π muss keine $\mathcal{GS}$-Inferenz sein, da die Ordnungsbedingung verletzt sein kann. Wir zeigen, dass $\sigma(\pi)$ korrekt ist, also gezogen werden darf. Ist $\sigma(\pi)$ eine $\mathcal{GS}$-Inferenz, so muss $\sigma(\pi)$ gezogen werden.

π ist **monoton**, falls jede $\mathcal{GS}$-Inferenz $\sigma(\pi)$ monoton ist.

Lemma 4.26

$\mathcal{S}$ ist korrekt und monoton: Für jede Inferenz

$\pi \equiv \dfrac{C_1 \ldots C_n C}{D \equiv \tau(D_0)}$ gilt $\{C_1, \ldots, C_n, C\} \models D$.

Ist $\sigma(\pi)$ eine $\mathcal{GS}$-Inferenz, so gilt $\sigma(\tau(C)) \succ_c \sigma(\tau(C_i))$ und $\sigma(\tau(C)) \succ_c \sigma(\tau(D_0))$.

Beweis

Der Beweis läuft analog zum Beweis zu Lemma 4.22: Jede Grundinstanz $\sigma(\pi)$ ist korrekt: $\{\sigma(\tau(C_1)), \ldots, \sigma(\tau(C_n)), \sigma(\tau(C))\} \models \sigma(\tau(D_0))$. Ist $\sigma(\pi)$ eine $\mathcal{GS}$-Inferenz, so ist $\sigma(\pi)$ monoton.

Nicht liftbare $\mathcal{S}$-Inferenzen

Erinnerung

Seien $C'_1, \ldots, C'_n, C' \in \mathrm{gnd}(S)$ und $\pi' \equiv \dfrac{C'_1, \ldots, C'_n C'}{D'}$ eine $\mathcal{GS}$-Inferenz.

π' ist liftbar, falls es $C_1, \ldots, C_n, C \in S$ mit $C'_i \in \mathrm{gnd}(C_i)$, $C' \in \mathrm{gnd}(C)$ und eine $\mathcal{S}$-Inferenz $\pi = \dfrac{C_1 \ldots C_n C}{D}$ gibt mit $D' \in \mathrm{gnd}(D)$.

Beachte

1. Im Resolutionskalkül $\mathcal{R}$ und $\mathcal{R}_\succ$ ist jede Grundinferenz liftbar.
2. In $\mathcal{H}$ und $\mathcal{S}$ ist nicht jede Grundinferenz liftbar:
   Es bestehe S aus den beiden Unit-Klauseln $f(x,x) = x$ und $a = b$. Sei $a \succ b$.
   Folgende Inferenz ist nicht liftbar
   $$\frac{a = b \quad f(a,a) = a}{f(a,b) = a}$$
3. a) Die Grundinferenz aus 2. ist redundant in $\mathrm{gnd}(S)$:
   Es gilt $f(b,b) = b \in \mathrm{gnd}(S)$ und $\{a = b, f(b,b) = b\} \models f(a,b) = a$ und $f(a,a) = a \succ_c f(b,b) = b, a = b$.
   b) Die Konklusion $f(a,b) = a$ entsteht durch eine Variablenüberlappung von $a = b$ in $f(x,x) = x$.

$\mathcal{S}$ ist liftend (1)

Lemma 4.27

Jede nicht-redundante $\mathcal{GS}$-Inferenz in gnd(S) ist liftbar zu einer $\mathcal{S}$-Inferenz in S mit nicht-redundanten Prämissen.

Beweis-Skizze:

Der problematische Fall einer $\mathcal{GS}$-Inferenz ist die Superpositions-Inferenz. Wir betrachten hier als typischen Spezialfall die positive Superposition. Sei π' nicht-redundant und von der Form

$$\pi' \equiv \quad \frac{C'_0 \equiv C' \vee s' = t' \quad D'_0 \equiv D' \vee u' = v'}{C' \vee D' \vee u'[t']_p = v'} \quad s' \equiv u'/p,$$

Da π' nicht-redundant ist, sind C'_0 und D'_0 nicht-redundante Grundklauseln. Nach Lemma 4.18b) gibt es nicht-redundante Klauseln $C_0 \equiv C \vee s = t$ und $D_0 \equiv D \vee u = v$ mit $C'_0 \in$ gnd(C_0) und $D'_0 \in$ gnd(D_0). Seien o.B.d.A C_0 und D_0 Variablen-disjunkt. Dann gibt es eine Substitution σ mit $C'_0 \equiv \sigma(C_0)$, $D'_0 \equiv \sigma(D_0)$, $\sigma(s) \equiv s'$ und $\sigma(u) \equiv u'$, also $\sigma(u)/p \equiv \sigma(s)$.

$\mathcal{S}$ ist liftend (2)

a) Sei $p \in$ Pos(u) und u/p keine Variable.
Dann existiert $\mu =$ mgu$(u/p, s)$. Sei π die Inferenz

$$\pi \equiv \quad \frac{C \vee s = t \quad D \vee u = v}{\mu(C \vee D \vee u[t]_p = v)}$$

Die Ordnungsbedingungen zu π' garantieren, dass die Ordnungsbedingungen zu π erfüllt sind. Dann ist π eine $\mathcal{S}$-Inferenz mit nicht-redundanten Prämissen und π' ist zu π liftbar.

b) Sei $p \notin$ Pos(u) oder $p \in$ Pos(u) und u/p eine Variable. Wir zeigen, dass dieser Fall nicht auftreten kann, da π' redundant ist:
Sei $\sigma(x)/p \equiv s'$ und sei τ gegeben durch $\tau(x) \equiv \sigma(x)[t']_p$ und $\tau(y) \equiv \sigma(y)$ sonst.
Dann gilt $\sigma(w) \succeq \tau(w)$ und $s' = t' \models \sigma(w) = \tau(w)$ für jeden Term w und daher $\{C' \vee s' = t', \tau(D \vee u = v)\} \models C' \vee D' \vee u'[t']_p = v'$.
Außerdem gilt $\sigma(x) \succ \tau(x)$, also $\sigma(u) \succ \tau(u)$ und somit $D' \vee u' = v' \succ \tau(D \vee u = v)$. Da $D' \vee u' = v'$ die Hauptprämisse ist, gilt außerdem $D' \vee u' = v' \succ C' \vee s' = t'$.
In diesem Fall ist also die Inferenz π' redundant in gnd(S).

Hauptergebnis zu $\mathcal{S} = \mathcal{S}_{\succ}$

Sei

S eine Menge von Klauseln.

Lemma 4.28

Sei $(S_i)_{i\in\mathbb{N}}$ eine $\mathcal{S}$-faire $\mathcal{S}$-Ableitung. Dann ist S_∞ $\mathcal{S}$-saturiert bis auf Redundanz.

Beweis: Dies folgt aus Lemma 4.21.

Satz 4.20.

$\mathcal{S}$ ist korrekt und vollständig für Klauseln:

a) Es gilt $S \overset{*}{\underset{\mathcal{S}}{\vdash}} \Box \text{ gdw } S$ ist H-unerfüllbar.

b) Für jede $\mathcal{S}$-faire $\mathcal{S}$-Ableitung $(S_i)_{i\in\mathbb{N}}$ mit $S_0 = S$ gilt:
S ist H-unerfüllbar $\text{gdw } \Box \in S_n$ für ein $n \in \mathbb{N}$.

Beweis:

a) folgt aus b).

b) folgt aus Satz 4.16.

4.5 Ausnutzung der Redundanz

Sei $\mathcal{K}$ ein korrektes Inferenzsystem auf Klauseln.

Beachte

Ausnutzung des Redundanz-Begriffs ist von zentraler Bedeutung für die Effizienz von Saturierungsprozeduren.

Es gibt zwei Redundanzbegriffe

- Redundanz einer Klausel

- Redundanz einer Inferenz

Redundante Klauseln kann man beliebig einfügen und löschen.

Redundante Inferenzen kann man, muss man aber nicht ziehen.

Redundante Klauseln dienen zum Säubern der Datenbasis S.

Also

Wir betrachten Paare (S, T) mit der Eigenschaft: Jedes $C \in T$ ist bezüglich S redundant, d. h., $T \subseteq R(S)$.

Beachte

$\mathcal{K}$-Inferenzen, bei der eine Prämisse in T liegt, sind redundant.

Inferenzen mit expliziter Redundanz

Definition 4.35.

Eine **Klauselmenge mit Redundanz** ist ein Paar (S,T) mit $T \subseteq R(S)$.

Eine **Inferenz mit expliziter Redundanz** ist $\Pi = \dfrac{(S,T)}{(S',T')}$ mit $T \subseteq R(S)$ impliziert $T' \subseteq R(S')$. Π heißt **korrekt**, falls $S \models S'$ und $S - S' \subseteq R(S')$ gilt.

Beachte

Wir betrachten nur Paare (S,T) mit $T \subseteq R(S)$.

Aus $S \cup T \models S'$ folgt also $S \models S'$.

Bemerkung

Sei $\Pi = \dfrac{(S,T)}{(S',T')}$ eine Inferenz mit expliziter Redundanz.

a) Ist $T_0 \subseteq R(S)$, $S' = S - T_0$ und $T' = T \cup T_0$, so ist Π korrekt.

b) Ist $S \subseteq S'$, $S \models S'$ und $T' \subseteq T$, so ist Π korrekt.

c) Sei π eine $\mathcal{K}$-Inferenz mit Prämissen aus $S \cup T$ und Konklusion D. Ist $S' = S \cup \{D\}$ und $T' = T$, so ist Π eine $\mathcal{K}$-**Inferenz mit expliziter Redundanz**. Sie ist korrekt, da $\mathcal{K}$ als korrekt vorausgesetzt ist.

Kontrahierende Inferenzen Π

Ziel

Entwicklung von Kriterien zur „Säuberung“ von S.

Definition 4.36.

Eine Inferenz $\Pi = \dfrac{(S,T)}{(S',T')}$ heißt **kontrahierend**, falls $S \gg_c S'$ gilt.

Lemma 4.29

Folgende Inferenzen Π sind korrekt und kontrahierend:

a) Tautologie-Elimination

$$\frac{(S \cup \{C\},T)}{(S,T)} \quad \text{falls } C \text{ eine Tautologie ist}$$

b) Faktorisierung / Reflexiv-Resolution

$$\frac{(S \cup \{C \vee l \vee \cdots \vee l\},T)}{(S \cup \{C \vee l\},T)} \qquad \frac{(S \cup \{C \vee \neg s = s\},T)}{(S \cup \{C\},T)}$$

c) Subsumption

$$\frac{(S \cup \{C,D\},T)}{(S \cup \{C\},T)} \quad \begin{array}{l}\text{falls es ein } \sigma \text{ gibt mit } \sigma(C) \subseteq D \text{ oder}\\ C \equiv s = t \text{ und}\\ D \equiv D' \vee u[\sigma(s)]_p = u[\sigma(t)]_p\end{array}$$

Bemerkungen zu kontrahierenden Inferenzen (1)

Tautologien und Faktorisierung

$C \vee s = s$ und $C \vee l \vee \neg l$ sind Tautologien.

Diese Tests auf Anwendbarkeit der Tautologie-Elimination, der Faktorisierung und der Reflexiv-Resolution sind rein syntaktisch. Sie sind effizient programmierbar, also werden diese Kriterien für Säuberungsschritte mit Vorrang abgeprüft.

Subsumption

a) C subsumiert $C \vee D$.
$s = t$ subsumiert $C \cup u[\sigma(s)]_p = u[\sigma(t)]_p$.
Diese Kriterien sind leicht abprüfbar.

b) C subsumiert $\sigma(C)$, falls $\sigma(C)$ eine echte Instanz von C ist, also $\sigma(C) \rhd C$ gilt. Dieses Kriterium ist auch relativ leicht abprüfbar.

c) Generell ist der Test, ob C D subsumiert schwer. Also wird der volle Test nur selten ausgeführt.

d) Sei $C \in T$ und C subsumiert D. Dann ist auch
$\Pi = \dfrac{(S \cup \{D\}, T)}{(S, T)}$ korrekt und kontrahierend.

Bemerkungen zu kontrahierenden Inferenzen (2)

- Kontrahierende und sichere Inferenzen erstzen in der Regel eine Klause C aus S durch eine kleinere oder sie löschen C.
- Im Gegensatz dazu generieren die $\mathcal{K}$-Inferenregel aus S eine neue Klausel. Diese Inferenzen heißen daher auch expandierend.
- Manche $\mathcal{K}$-Inferenzen π generieren eine neue Klausel D, die die Hauptprämisse C redundant macht. Solche $\mathcal{K}$-Inferenzen erzeugen eine kontrahierende Inferenz Π, die das Generieren und Löschen in einen Schritt macht. Siehe dazu z.B. die Simplifikationsregeln der KB-Vervollständigung und Lemma 4.30.
- Aus Effizienzgründen sollten kontrahierende Inferenzen mit höherer Priorität angewandt werden als expandierende. Man beachte dazu, dass C nicht in S_∞ liegt, wenn C gelöscht oder ersetzt wird. Deswegen brauchen keine S-Inferenzen gezogen werden, die C als Prämisse enthalten.
- Wird C gelöscht, so kann man C in T aufnehmen und daher für kontrahierende Inferenzen benutzen.
- Wir geben weiter kontrahierende Inferenzen an.

Sichere Inferenzen (1)

Definition 4.37.

Eine Inferenz $\pi = \dfrac{C_1 \ldots C_n C}{D}$ heißt **sicher**, falls C redundant in $\{C_1, \ldots, C_n, D\}$ ist.

Beispiele für sichere $\mathcal{K}$-Inferenzen

a) Sei $\mathcal{K} = \mathcal{R}_{\succ}$ der Resolutionskalkül

$$\frac{C \vee A \vee A}{C \vee A}$$ ist sicher.

b) Sei $\mathcal{K} = \mathcal{H}_{\succ}$ der Superpositionskalkül für Horn-Klauseln. Folgende Superposition-Inferenzen (Rewriting) sind sicher.

$$\frac{\Rightarrow l \doteq r \quad \Gamma \Rightarrow s[\sigma(l)]_p \doteq t}{\Gamma \Rightarrow s[\sigma(r)]_p = t}$$

$$\frac{\Rightarrow l \doteq r \quad \Gamma, s[\sigma(l)]_p \doteq t \Rightarrow \Delta}{\Gamma, s[\sigma(r)]_p = t \Rightarrow \Delta}$$

Lemma 4.30

Sei π eine sichere und korrekte Inferenz mit Prämissen aus $S \cup T$, Hauptprämisse C, Konklusion D und $C \succ_c D$. Dann ist Π korrekt und kontrahierend:

$$\Pi = \frac{(S \cup \{C\}, T)}{(S \cup \{D\}, T)}$$

Sichere -Inferenzen (2)

Lemma 4.31

Folgende Inferenzen sind korrekt und sicher:

Contextual Rewriting

$$\frac{C \vee s \doteq t \quad D \vee l[\sigma(s)]_p}{D \vee l[\sigma(t)]_p}$$ falls $\sigma(s) \succ \sigma(t)$ und $\sigma(C) \models D$

Constant Rewriting

$$\frac{C \vee \neg s \doteq t \vee l[s]_p}{C \vee \neg s = t \vee l[t]_p}$$ falls $s \succ t$

Beachte

In der Regel Contextual Rewriting gilt $\sigma(C) \models D$, falls $\sigma(C) \subseteq D$ gilt. Dies ist leicht zu prüfen. Contextual Rewriting ist für den Spezialfall $C = \Box$ die normale Simplifikation und besonders einfach zu implementieren. Simplifikation mit höchster Priorität steigert enorm die Effizienz. Contextual Rewriting und Constant Rewriting mit Klauseln aus $S \cup T$ wirken säubernd auf (S, T).

Die $\mathcal{S}$-Inferenzen Reflexiv-Resolution und Ordnungsfaktorisierung mit $\sigma = $ id sind auch sicher und korrekt. Sie fallen unter die Faktorisierung in Lemma 4.29.

Sichere Inferenzen (3)

Lemma 4.32

Folgende Inferenzregel ist sicher und kontrahierend:

$$\frac{S \cup \{C \vee D\}}{S \cup \{D\}}, \quad \text{falls } C \neq \Box \text{ und es ein } \sigma \text{ gibt, so dass } \models \sigma(C \Rightarrow D) \text{ und } \models \sigma(D) \Rightarrow D \text{ gilt}$$

Beweis:

Sicherheit: D macht $C \vee D$ redundant

Korrektheit: Es reicht zu zeigen: Ist $\mathcal{A}$ ein H-Modell mit $\mathcal{A} \models C \vee D$, so gilt $\mathcal{A} \models D$.

Sei $\mathcal{A} \models C \vee D$ und τ eine Grundsubstitution.

Zu zeigen ist $\mathcal{A} \models \tau(D)$.

Es gilt $\mathcal{A} \models \tau(\sigma(C \vee D))$ wegen $\mathcal{A} \models C \vee D$.

i) Sei $\mathcal{A} \models \tau(\sigma(D))$. Es folgt dann $\mathcal{A} \models \tau(D)$ wegen $\models \sigma(D) \Rightarrow D$.

ii) Sei $\mathcal{A} \models \tau(\sigma(C))$. Es folgt dann $\mathcal{A} \models \tau(\sigma(D))$ wegen $\models \sigma(C \Rightarrow D)$, also auch $\mathcal{A} \models \tau(D)$ nach (i).

Bemerkungen zu Lemma 4.32

Beachte

Es gilt $\models C_1 \Rightarrow C_2$, falls $C_1 \subseteq C_2$ gilt. Die Seitenbedingung im Lemma 4.32 ist erfüllt, falls

a) $\sigma = id$ und $\models C \Rightarrow D$ oder

b) $\models \sigma(C) \Rightarrow D$ und $\sigma(D) \equiv D$.

Beispiele

1. $\dfrac{s = t \vee u[s]_p = u[t]_p}{u[s]_p = u[t]_p}$
2. $\dfrac{p(x) = \top \vee p(f(y)) = \top}{p(f(y)) = \top}$
3. $\dfrac{p(y) = \top \vee p(f(y)) = \top}{p(f(y)) = \top}$
4. $\dfrac{s = t \vee u = v \vee f(s) = f(t) \vee g(u) = g(v) \vee D'}{f(s) = f(t) \vee g(u) = g(v) \vee D'}$

1. ist sicher und korrekt nach a)
2. ist sicher und korrekt nach b) mit $\sigma = \{x \leftarrow f(y)\}$
3. ist sicher, aber nicht korrekt! (Übung)
4. ist sicher und korrekt nach a)

Das Hauptresultat zu $\mathcal{K}$

Definition 4.38.

Eine **Ableitung mit Redundanz** ist eine Folge $(S_i, T_i)_{i\in\mathbb{N}}$, so dass

$$\frac{(S_i, T_i)}{(S_{i+1}, T_{i+1})} \qquad \text{für alle } i \in \mathbb{N} \text{ eine korrekte Inferenz ist.}$$

Sie heißt $\mathcal{K}$-**fair**, falls $(S_i)_{i\in\mathbb{N}}$ $\mathcal{K}$-fair ist.

Lemma 4.33

Eine Ableitung $(S_i, T_i)_{i\in\mathbb{N}}$ ist $\mathcal{K}$-fair, falls $\mathcal{K}$ korrekt und monoton ist und con$(\mathcal{K}(S_\infty - R(S_\infty)) - \mathcal{K}_R(S_\infty - R(S_\infty))) \subseteq S^\infty \cup T^\infty \cup R(S^\infty \cup T^\infty)$ gilt. Dann ist S_∞ $\mathcal{K}$-saturiert bis auf Redundanz.

Satz 4.21.

Sei $\mathcal{K}$ monoton, korrekt und liftend. $\mathcal{K}$ habe die Verkleinerungseigenschaft für Gegenbeispiele und $(S_i, T_i)_{i\in\mathbb{N}}$ sei eine $\mathcal{K}$-faire Ableitung. Dann gilt

$$S_0 \text{ ist H-unerfüllbar } \mathrm{gdw}\ \Box \in S_n \text{ für ein } n \in \mathbb{N}.$$

Grundzusammenführbarkeit (1)

Ziel

Übertragung des Redundanzkriteriums Grundzusammenführbarkeit aus Abschnitt 3.2 auf $\mathcal{S}$.

Bezeichnung

Sei $E = \{u_1 = v_1, \ldots, u_n = v_n\}$ ein Gleichungssystem und s, t Grundterme.

$s \rightarrow^0 t$ in $E^\succ$, falls es ein σ und ein $u_i \doteq v_i$ aus E gibt mit $\sigma(u_i) \succ \sigma(v_i)$, $s = t \succ_l \sigma(u_i) = \sigma(v_i)$ und $s \rightarrow t$ mit der Regel $\sigma(u_i) \rightarrow \sigma(v_i)$.

Definition 4.39.

$s = t$ heißt **in $E^\succ$ stark grundzusammenführbar**, falls für jede Grundinstanz $\sigma(s) = \sigma(t)$ gilt

- $\sigma(s) \equiv \sigma(t)$ oder
- $\sigma(s) \succ \sigma(t)$ und $\sigma(s) \rightarrow^0 s_0 \downarrow \sigma(t)$ in $E^\succ$ oder
- $\sigma(t) \succ \sigma(s)$ und $\sigma(t) \rightarrow^0 t_0 \downarrow \sigma(s)$ in $E^\succ$.

$s = t$ heißt **in S stark grundzusammenführbar**, falls es Unit-Klauseln $u_i = v_i$ in S gibt, so dass $s = t$ in $E^\succ = \{u_1 = v_1, \ldots, u_n = v_n\}^\succ$ stark grundzusammenführbar ist.

Grundzusammenführbarkeit (2)

Beachte

Sei $E^{\succ} = \{\sigma(u) \to \sigma(v) \mid u \doteq v \text{ in } E, \sigma(u) \succ \sigma(v)\}$. Gilt $s \to s_0 \downarrow t_0 \leftarrow t$ in $E^{\succ}$ und $s \to s_0$ und $t \to t_0$ jeweils bei Position $p \neq \lambda$, so ist $s = t$ in $E^{\succ}$ stark grundzusammenführbar.

Lemma 4.34

Ist $s = t$ stark grundzusammenführbar in S, so ist $s = t$ redundant in S.

Lemma 4.35

Ist $s = t$ in $S \cup T$ stark grundzusammenführbar, so ist die Inferenz

$$\frac{(S \cup \{C \vee s = t\}, T)}{(S, T \cup \{s = t\})}$$

korrekt und kontrahierend.

Hauptergebnis für $\mathcal{S}$

Sei S_0 eine Menge von Klauseln.

Bezeichnung

Ein **Saturierungsprozess** von S_0 besteht aus der Berechnung einer $\mathcal{S}$-fairen korrekten Ableitung $(S_i, T_i)_{i \in \mathbb{N}}$.

Ein Ableitungsschritt (S_i, T_i) nach (S_{i+1}, T_{i+1}) besteht dabei aus (1) oder (2):

(1) dem Ziehen einer $\mathcal{S}$-Inferenz π zu S_i und
$S_{i+1} = S_i \cup \{\text{con}(\pi)\}$,

(2) dem Ziehen einer korrekten und kontrahierenden Inferenz

$$\frac{(S_i, T_i)}{(S_{i+1}, T_{i+1})}$$

Beachte

Schritte nach (1) sind nötig, um Fairness zu erreichen. Schritte nach (2) sind wichtig für die Säuberung der ersten Komponente in den (S, T), also für die Effizienz des Saturierungsprozesses.

Satz 4.22.

Sei $\mathcal{A}$ ein Saturierungsprozess mit Eingabe $(S_0, \emptyset)$.

Es ist S_0 genau dann H-unerfüllbar, wenn $\mathcal{A}$ ein (S_n, T_n) liefert mit $\square \in S_n$.

Literatur

Buch

Chang, C.-L. und Lee, R.:

Symbolic Logic and Mechanical Theorem Proving, Academic Press 1990.

Robinson, A. und Voronkov, A. (Eds.):

Handbook of Automated Reasoning, Volume I, Elsevier 2001

Übersichtsartikel

Ganzinger, H. und Bachmair, L.:

Equational Reasoning in Saturation-Based Theorem Proving, in B. Bibel, D. Schmitt (Eds.) Automated Deduction - A Basis for Applications Vol. I, Kluwer 1998, pp. 353-397.

Ganzinger, H. und Niewenhuis, R.:

Constraints and Theorem Proving, in H. Comon, C. Marche, R. Treinen (Eds.): Constraints in Computational Logics: Theory and Applications, Springer 2000, LNCS 2002, pp. 159-201.

Nonnengart, A. und Weidenbach, Chr.:

Computing Small Clause Normal Forms, in A. Robinson, A. Voronkov (Eds.): Handbook of Automated Reasoning, Elsevier 2001.

Zeitschriftenartikel

Avenhaus, J., Hillenbrand, Th., Löchner, B.:

On Using Ground Joinable Equations in Equational Theorem Proving, erscheint in J. Symb. Comput.

Bachmair, L., Deshowitz, N., Plaisted, D.:

Completion without Failure, in H. Ait Kaci, M. Nivat (Eds.): Resolution of Equations in Algebraic Structures, Vol. 2, Academic Press 1989, pp. 1-30.

Bachmair, L., Ganzinger, H.:

Rewrite-Based Equational Theorem Proving with Selection and Simplification, J. of Logic and Comput., 1994, pp. 217-247.

Weitere Bücher

Bibel, W.:

Deduktion: Automatisierung der Logik, Oldenbourg 1992.

Loveland, D.:

Automated Theorem Proving: A Logical Basis, North Holland 1978.