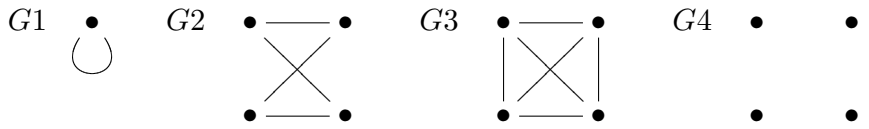
Formale Grundlagen der Informatik II
6. Übungsblatt

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik **SS 2011**
Prof. Dr. Martin Ziegler **06.07.11**
Alexander Kreuzer
Carsten Rösnick

Minitest Lösung

Gegeben seien die folgenden ungerichteten Graphen $G = (V, E)$:



In welchem der obigen Graphen gilt welcher der nachfolgenden FO-Sätze?

(a) $G\boxed{3}$ $\quad \forall x \forall y \left(\neg(x = y) \leftrightarrow Exy\right)$

(b) $G\boxed{2}$ $\quad \exists x \exists y \exists z \left(\neg(x = y) \wedge \neg(y = z) \wedge Exy \wedge Eyz \wedge \neg Ezx\right)$

(c) $G\boxed{4}$ $\quad \exists x \exists y \neg \left(x = y\right) \wedge \forall x \forall y \left(\neg(x = y) \rightarrow \neg Exy\right)$

(d) $G\boxed{1}$ $\quad \exists x \forall y \left(x = y\right)$

Begründung: Die angegebenen FO-Sätze haben folgende Bedeutung:

(a) Je zwei verschiedene Knoten sind miteinander verbunden.

(b) Es gibt drei Knoten, die keinen Kreis bilden.

(c) Der Graph enthält keine Kante, aber mindestens zwei Knoten.

(d) Der Graph besteht aus nur einem Knoten.

Gruppenübung

Aufgabe G1

(a) Bestimmen Sie die Skolem-Normalform der folgenden Formel:

$$\varphi = \exists a \forall b \forall c \exists x \exists y \forall z \left(a \cdot b + x + z > c + y\right).$$

(b) Geben Sie eine Interpretation der Skolem-Normalform von φ über der Struktur $\mathcal{N} = (\mathbb{N}, <^{\mathbb{N}}, +^{\mathbb{N}}, \cdot^{\mathbb{N}})$ der natürlichen Zahlen an.

(c) Geben Sie eine FO(S)-Formel $\varphi'(x)$ über der Signatur $S = (+, \cdot)$ an, die über der Struktur $\mathcal{N}$ äquivalent ist zu der Aussage: x ist eine Primzahl.

Lösungsskizze:

(a) Die Skolem-Normalform von φ ergibt sich wie folgt:

$$\begin{aligned} & \exists a \forall b \forall c \exists x \exists y \forall z \, (a \cdot b + x + z > c + y) \\ \equiv\ & \forall b \forall c \exists x \exists y \forall z \, (f_a \cdot b + x + z > c + y) \\ \equiv\ & \forall b \forall c \exists y \forall z \, (f_a \cdot b + g_x bc + z > c + y) \\ \equiv\ & \forall b \forall c \forall z \, (f_a \cdot b + g_x bc + z > c + h_y bc) \end{aligned}$$

Dabei sei f_a eine null-stellige, sowie g_x und h_y zwei-stellige Skolemfunktionen.

(b) Wähle als Interpretationen der Skolemfunktionen beispielsweise $f_a^{\mathcal{N}} = 1$, $g_x^{\mathcal{N}} bc = c$, $h_y^{\mathcal{N}} bc = 0$.

(c) Schreibe beispielsweise

$$\varphi'(x) = \neg(x = 1) \wedge \forall y \exists z \, (y \cdot z = x \rightarrow (y = 1 \vee y = x)) \, .$$

Aufgabe G2

Ein Pfad in einem Graph $\mathcal{G} = (V, E)$ ist eine Sequenz $\langle x_0, x_1, \ldots, x_n \rangle$ von Knoten, so dass

$$x_i E x_{i+1}$$

für alle $i < n$. Der Graph heißt *zusammenhängend*, wenn es für alle Paaren von Knoten (x, y) einen Pfad $\langle x_0, x_1, \ldots, x_n \rangle$ gibt, mit $x = x_0$ und $y = x_n$.
Zeigen Sie, dass es keine Formelmenge Γ in der Sprache der Graphen gibt, so dass $\mathcal{G} \models \Gamma$ genau dann wenn $\mathcal{G}$ zusammenhängend ist.

Lösungsskizze: Wir verwenden, dass man eine Formel $\varphi_n(x, y)$ definieren kann, die aussagt, dass es einen Pfad der Länge n vom Startzustand nach y gibt:

$$\varphi_n(y) = \exists x_0 \ldots \exists x_n \, (\, x_0 = x \wedge x_n = y \wedge \bigwedge_{i<n} x_i E x_{i+1} \,).$$

Nehmen wir an, dass es eine Formelmenge Γ gibt in der Sprache der Graphen, so dass ein Graph $\mathcal{G}$ ein Modell von Γ ist, genau dann wenn $\mathcal{G}$ zusammenhängend ist. Wir erweiteren die Signatur mit zwei Konstanten c und d und betrachten die folgende Formelmenge in der erweiterten Sprache:

$$\Gamma_\infty = \Gamma \cup \{\neg\varphi_n(c, d) \, : \, n \in \mathbb{N}\}.$$

Die Formelmenge Γ_∞ is unerfüllbar, da man in einem Modell $\mathcal{G}$ die Konstanten c und d nicht widerspruchsfrei interpreteren kann: einerseits soll der Zustand $d^{\mathcal{G}}$ von $c^{\mathcal{G}}$ aus erreichbar sein, da Γ erfüllt ist und der Graph $\mathcal{G}$ deshalb zusammenhängend sein muss; andererseits kann $d^{\mathcal{G}}$ nicht von $c^{\mathcal{G}}$ aus erreichbar sein: dann würde es einen Pfad von $c^{\mathcal{G}}$ nach $d^{\mathcal{G}}$ geben; dieser Pfad hat eine bestimmte Länge n, was unmöglich ist, da $\mathcal{G} \models \neg\varphi_n(c, d)$.
Also ist schon eine endliche Teilmenge von Γ_∞ unerfüllbar und insbesondere ist schon eine Teilmenge von der Form

$$\Gamma_n = \Gamma \cup \{\neg\varphi_k(c, d) \, : \, k < n\}$$

unerfüllbar (da jede endliche Teilmenge in einer Γ_n enthalten ist). Aber jedes Γ_n hat ein Modell, wobei es einen Pfad von $c^{\mathcal{G}}$ nach $d^{\mathcal{G}}$ gibt, aber keinen mit einer Länge kürzer als n. (Ein Modell könnte so aussehen:

$$0 \longrightarrow 1 \longrightarrow \cdots \longrightarrow n \, ,$$

wobei wir c als der 0-Knoten und d als der n-Knoten interpretieren.)
Also haben wir einen Widerspruch und schliessen, dass es keine Formel Γ geben kann, die die Zusammenhang eines Graphen ausdrückt.

Aufgabe G3
Zeigen Sie, dass eine Theorie T, die beliebig grosse endliche Modelle hat, auch ein unendliches Modell besitzt.
Geben Sie auch einen Satz an, der endliche und unendliche Modelle hat, aber keine beliebig grossen endliche Modelle.

Lösungsskizze: Wir haben gesehen, dass es eine Formel φ_n gibt, die genau dann wahr ist in einem Modell, wenn das Modell mehr als n Elemente hat, nämlich

$$\varphi_n = \exists x_1 \dots \exists x_n \bigwedge_{i \neq j} \neg x_i = x_j.$$

Betrachte $T_\infty = T \cup \{\varphi_n : n \in \mathbb{N}\}$. Da die Modelle von T_∞ genau die Modelle von T sind, die unendlich viele Elemente haben, müssen wir zeigen, dass T_∞ erfüllbar ist.
Wir beweisen, dass T_∞ erfüllbar ist, indem wir zeigen, dass jede endliche Teilmenge von T_∞ ein Modell hat. Sei Γ also eine endliche Teilmenge von T_∞. Da Γ nur endliche viele φ_n enthalten kann, gibt es ein $k \in \mathbb{N}$, so dass

$$\Gamma \subseteq T \cup \{\varphi_n : n \leq k\}.$$

Da T beliebig grosse endliche Modelle hat, gibt es ein Modell M von T mit mehr als k Elementen. Also $M \models \varphi_n$ für alle $n \leq k$, und deshalb $M \models \Gamma$.
Wir haben gesehen, dass es ein Satz ψ gibt die nur unendliche Modelle hat, und ein Satz φ, die nur dann wahr ist, wenn die Trägermenge höchstens fünf Elemente hat. Dann hat der Satz

$$\psi \vee \varphi$$

endliche und unendliche Modelle, aber keine beliebig grossen Modelle.

Hausübung

Aufgabe H1 (6 Punkte)
Welche der folgenden Eigenschaften der intendierten (d.h. hier: der auf $\mathbb{N}$ natürlichen) linearen Ordnung $(\mathbb{N}, <)$ lassen sich in $\mathrm{FO}(<)$ formalisieren, welche nicht?
Hier sind klare Begründungen (z.B. Formalisierungen oder Kompaktheitsargumente) verlangt.

(a) Jedes Element besitzt einen direkten Nachfolger.

(b) Es gibt kein letztes Element.

(c) Jedes Element hat nur endlich viele Vorgänger.

Lösungsskizze:

(a) $\forall x \exists y (x < y \wedge \forall z \neg (x < z \wedge z < y))$

(b) $\forall x \exists y \, x < y$

(c) Angenommen es gäbe einen solchen Satz φ. Wir fügen zu der Signatur eine neue Konstante c hinzu. Definiere

$$\psi_n := \exists x_1 \dots x_n \left(\bigwedge_i x_i < c \wedge \bigwedge_{i \neq j} \neg x_i = x_j \right).$$

Der Satz ψ_n sagt aus, dass c mindestens n verschiedene Vorgänger hat. Offensichtlich gilt, dass $\Phi := \{\varphi\} \cup \{\psi_n \colon n \in \mathbb{N}\}$ nicht erfüllbar ist. Nach Kompaktheitssatz gilt, dass eine endliche Teilmenge $\Phi_0 \subseteq \Phi$ bereits unerfüllbar sein muss. Allerdings sind alle endliche Teilmengen von Φ erfüllbar (z.B. ist $\mathcal{M} = (\mathbb{N}, <, c^{\mathcal{M}})$ mit $c^{\mathcal{M}} = \max\{n \colon \psi_n \in \Phi_0\}$ ein Modell von Φ_0).

Damit ist die Annahme, dass es ein solches φ gibt, falsch.

Aufgabe H2

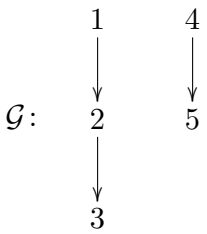
Betrachten Sie einen endlichen, gerichteten Graphen $\mathcal{G} = (V, E)$.
In der Aufgabe G3 haben wir gesehen, dass es keinen FO-Satz $\varphi(x, y)$, der genau dann wahr ist, wenn es einen Pfad von dem Knoten x zu dem Knoten y gibt. (Gäbe es eine solche FO-Formel $\varphi(x, y)$, so würde $\Gamma = \forall x, y\, \varphi(x, y)$ ausdrücken, dass $\mathcal{G}$ zusammenhängend ist.)
Wir wollen in dieser Aufgabe untersuchen, wie man zu einem allgemeinen $\mathcal{G}$ eine zwei-stellige Relation Cxy (für **c**onnected), die das aussagt, hinzufügen kann, und was das für Programme bedeutet.

(a) Was bedeutet es für ein Prolog-Programm, bzw. für die Definition von Cxy in Prolog, dass es kein $\varphi(x, y)$ mit $Cxy \leftrightarrow \varphi(x, y)$ gibt?
Was bedeutet das für eine imperative Programmiersprache wie C oder Java?

(b) Betrachten Sie die Sätze

$$\begin{aligned} &\forall x \forall y\ (Exy \to Cxy) \\ &\forall x \forall y \forall z\ ((Exz \wedge Czy) \to Cxy)\ . \end{aligned} \tag{1}$$

Was bedeuten diese Sätze umgangssprachlich?
Geben Sie einige Erweiterungen (d.h. Interpretationen der Relation C) des folgenden Graphen um eine Relation C an, die diese Sätze erfüllt.



Welche Erweiterung ist die intendierte (=gewollte, natürliche) Erweiterung, also die Erweiterung, in der Cxy bedeutet, dass es einen Pfad von x nach y gibt?

(c) Sei nun $\mathcal{G} = (V, E) = (V, E^{\mathcal{G}})$ wieder ein beliebiger endlicher, gerichteter Graph.
Zeigen Sie, dass zwei Erweiterungen (V, E, C_1), (V, E, C_2), die die Sätze aus (1) erfüllen, auch $(V, E, C_1 \cap C_2)$ die Sätze aus (1) erfüllen.
Folgern Sie daraus, dass es eine minimale Erweiterung (V, E, C_{min}), die (1) erfüllt gibt, d.h. eine Erweiterung, so dass für eine jede andere solche Erweiterung (V, E, C_1) gilt $C_{min} \subseteq C_1$.

Wir versuchen nun diese C_{min} von unten zu approximieren.

(d) Betrachten Sie dafür die Operation

$$\mathsf{F}(X) := \{\, (x, y) \mid (x, y) \in E^{\mathcal{G}} \text{ oder es gibt } z \in V \text{ mit } (x, z) \in E^{\mathcal{G}} \text{ und } (z, y) \in X \,\}.$$

Zeigen Sie, dass Φ monoton ist, d.h. das für alle $X, Y \subseteq V \times V$ gilt

$$X \subseteq Y \Longrightarrow \Phi(X) \subseteq \Phi(Y).$$

(e) Sei $C_0 := \emptyset$, $C_{n+1} = \mathsf{F}(C_n)$ und $C^* := \bigcup_{n=0}^{\infty} C_n$. Zeige, dass $C^* = C_{min}$ und damit, dass wir auf diese Weise die kleinste Erweiterung iterativ konstruiert haben.

Allgemein gilt, dass man für alle Sätze Φ der gleichen Form wie in (1) und zu jeder Struktur $\mathcal{A}$ eine minimale Erweiterung (um eine neue Relation R) finden kann, die dann auch von unten approximiert werden kann.[1]

[1] Genauer: Ein gleichungsfreier *nicht-negativer universeller Horn-Satz* ist ein Satz der Form

$$\forall x_1 \cdots \forall x_n [(\alpha_1 \wedge \cdots \wedge \alpha_m) \to \beta]\,,$$

wobei $\alpha_1, \ldots, \alpha_m, \beta$ gleichungsfreie atomare Formeln sind. Wenn Φ aus Horn-Sätze, in denen $\alpha_1, \ldots, \alpha_n$ atomare Formeln über der Signatur von $\mathcal{A}$ und R sind und β aus R besteht.

(f) Implementieren Sie die Relation C mit Hilfe der Sätze (1) in Prolog. Nehmen Sie dabei an, dass der Graph durch ein Relation V für die Knoten und eine Relation E gegeben ist.
Z.B. wäre der Graphen aus der (b) gegen durch

```
v(1). v(2). v(3). v(4). v(5).
e(1,2). e(2,3). e(4,5).
```

Sie dürfen annehmen, dass der Graph azyklisch ist, d.h. es gibt keine Pfade von einen Knoten x zurück zu x.
Da Prolog Tiefensuche verwendet um nach erfüllbaren Prädikaten zu suchen, müssen sonst besondere Vorkehrungen getroffen werden, damit ein Zyklus nur einmal durchlaufen wird.
Datalog, ein Dialekt von Prolog, approximiert die Prädikate, wie wir es in dem Aufgabenteil (e) getan haben, deswegen kann dort C für alle Graphen wie in (1) definiert werden. Damit das funktioniert ist die Verwendung von Rekursion in Datalog eingeschränkt, im Westentlichen auf Sätze vom Typ wie in der Fußnote beschrieben.

Lösungsskizze:

(a) Für ein Prolog Programm bedeutet das, dass die Relation C nicht einfach durch (möglicherweise mehrere) E alleine ausgedrück werden kann, weil das im Prinzip eine FO-Formel ist. Die Relation C kann also nur Rekursiv, d.h. in der Form

```
c(x,y) :- ..., c(z1,z2), ...
```

Für iterative Programmiersprachen heißt, dass man entweder auch Rekursion oder eine **while**-Schleife (was äquivalent ist) verwenden muss oder man brauch mehr Informationen über den Graphen, also z.B. die Anzahl der Knoten.

(b) Der erste Satz bedeutet, wenn es eine Kante zwischen zwei Knoten x, y gibt dann sind sie auch Verbunden. Der zweite bedeutet, wenn es eine Kante zwischen x und z gibt und z mit y Verbunden ist, dann ist auch x mit y verbunden.
U.a. folgende Erweiterungen sind möglich:

$$\begin{aligned} C_1 &:= \{(1,2),(2,3),(1,3),(4,5)\} \\ C_2 &:= C_1 \cup \{(2,1),(1,1)\} \\ C_3 &:= C_1 \cup \{(1,4),(1,5)\} \\ C_4 &:= \{1,2,3,4,5\} \times \{1,2,3,4,5\} \end{aligned}$$

Die Erweiterung (V, E, C_1) ist die intendierte Erweiterung.

(c) Aus dem ersten Satz von (1) folgt, dass für alle $(x, y) \in E$ gilt $(x, y) \in C_1$ und $(x, y) \in C_2$, damit $(x, y) \in C_1 \cap C_2$ und $(V, E, C_1 \cap C_2)$ erfüllt den ersten Satz aus (1).
Gilt nun für beliebige x, y, dass es ein $z \in V$ gibt, so dass $(x, z) \in E$ und $(z, y) \in C_1 \cap C_2$, dann müssen wir zeigen, dass auch $(x, y) \in C_1 \cap C_2$ gilt, damit die Struktur $(V, E, C_1 \cap C_2)$ auch den zweiten Satz aus (1) erfüllt. Dies gilt, weil mit $(z, y) \in C_1 \cap C_2$ auch $(z, y) \in C_1$ gilt und dann mit der Voraussetzung, dass (V, E, C_1) den zweiten Satz aus (1) erfüllt, auch $(x, y) \in C_1$. Genauso sieht man, dass $(x, y) \in C_2$ und damit $(x, y) \in C_1 \cap C_2$.
Es gibt eine kleinste Erweiterung, weil nach dem oben gezeigten der Schnitt über alle Erweiterungen auch eine Erweiterung ist.

(d) Wenn $X \subseteq Y$ gilt, dann gilt insbesondere $(z, y) \in X \Rightarrow (z, y) \in Y$. Damit

$$\begin{aligned} \mathsf{F}(X) &= \{\, (x,y) \mid (x,y) \in E^{\mathcal{G}} \text{ oder es gibt } z \in V \text{ mit } (x,z) \in E^{\mathcal{G}} \text{ und } (z,y) \in X \,\} \\ &\subseteq \{\, (x,y) \mid (x,y) \in E^{\mathcal{G}} \text{ oder es gibt } z \in V \text{ mit } (x,z) \in E^{\mathcal{G}} \text{ und } (z,y) \in Y \,\} \\ &= \mathsf{F}(Y) \end{aligned}$$

(e) Wir zeigen die Gleichheit von C^* mit C_{min} durch gegenseitige Inklusion. (V, E, C^*) ist eine Erweiterung, die (1) erfüllt (warum?). Da C_{min} eine minimale Erweiterung beschreibt gilt, $C_{min} \subseteq C^*$. Für C_{min} gilt $\emptyset = C_0 \subseteq C_{min}$ und $\mathsf{F}(C_{min}) = C_{min}$ (warum?). Mit Induktion und der Monotonie von F folgt, $C_i \subseteq C_{min}$ und damit $C^* \subseteq C_{min}$.

(f)
```
c(x,y) :- e(x,y).
c(x,y) :- v(z), e(x,z), c(z,y).
```