Humboldt-Universität zu Berlin
Lehrstuhl Logik in der Informatik
Prof. Dr. Nicole Schweikardt

4. Februar 2022

Logik in der Informatik

Wintersemester 2021/2022

Präsenzaufgaben

für die letzte Übungsstunde

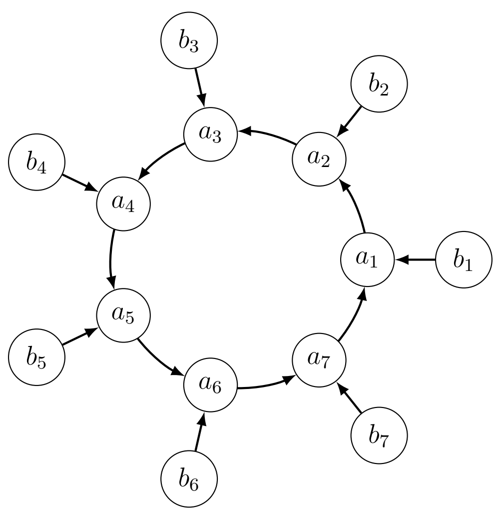
Aufgabe 1: (Endlichkeitssatz)

Sei $\sigma := \{E\}$ die Signatur, die aus dem 2-stelligen Relationssymbol E besteht.

Definition: Für eine σ-Struktur $\mathcal{A} = (A, E^{\mathcal{A}})$ und eine natürliche Zahl $n \geq 2$ sagen wir, dass $\mathcal{A}$ *eine Krone der Länge n besitzt*, wenn es Elemente $a_1, \ldots, a_n, b_1, \ldots, b_n \in A$ mit $|\{a_1, \ldots, a_n, b_1, \ldots, b_n\}| = 2n$ gibt, so dass die Relation $E^{\mathcal{A}}$ die folgenden Kanten enthält:

- (a_i, a_{i+1}) für alle $i \in \{1, \ldots, n-1\}$ und (a_n, a_1) und
- (b_i, a_i) für alle $i \in \{1, \ldots, n\}$.

Eine Krone der Länge 7 sieht zum Beispiel wie folgt aus:



(a) Geben Sie für jede natürliche Zahl $n \geq 2$ einen $\mathsf{FO}[\sigma]$-Satz φ_n an, sodass für jede σ-Struktur $\mathcal{A}$ gilt: $\mathcal{A} \models \varphi_n \iff \mathcal{A}$ enthält eine Krone der Länge n.

(b) Geben Sie eine Menge Ψ von $\mathsf{FO}[\sigma]$-Sätzen an, die die Klasse aller σ-Strukturen $\mathcal{A}$ axiomatisiert, für die gilt: Es gibt keine natürliche Zahl $n \geq 2$, so dass $\mathcal{A}$ eine Krone der Länge n besitzt.

(c) Verwenden Sie den Endlichkeitssatz der Logik erster Stufe, um Folgendes zu beweisen: Die Klasse aller σ-Strukturen $\mathcal{A}$, die eine Krone der Länge ≥ 2 besitzen, ist *nicht* erststufig axiomatisierbar. Präzise: Zeigen Sie, dass es keine Menge Φ von $\mathsf{FO}[\sigma]$-Sätzen gibt, so dass für jede σ-Struktur $\mathcal{A}$ gilt:

$$\mathcal{A} \models \Phi \iff \text{es gibt ein } n \in \mathbb{N} \text{ mit } n \geq 2, \text{ so dass } \mathcal{A} \text{ eine Krone der Länge } n \text{ besitzt.}$$

Aufgabe 2: (Logik-Programmierung)

In dieser Aufgabe bezeichnet AL' die Menge aller aussagenlogischen Formeln, die *keine Aussagensymbole* enthalten. Wir repräsentieren Formeln $\varphi \in \mathsf{AL}'$ wie folgt durch Terme $\mathtt{t}_\varphi \in \mathsf{T}_{\mathsf{LP}}$ der Logik-Programmierung:

- *Atomare Formeln:*
$$\mathtt{t}_{\mathbf{0}} \;:=\; \mathtt{0} \qquad \text{und} \qquad \mathtt{t}_{\mathbf{1}} \;:=\; \mathtt{1}$$
- *Rekursive Regeln:* Für Formeln $\varphi, \psi \in \mathsf{AL}'$ ist
$$\begin{aligned} \mathtt{t}_{\neg\varphi} &:= \mathtt{n(t}_\varphi\mathtt{)}, \\ \mathtt{t}_{(\varphi\vee\psi)} &:= \mathtt{o(t}_\varphi\mathtt{,\ t}_\psi\mathtt{)} \\ \mathtt{t}_{(\varphi\wedge\psi)} &:= \mathtt{u(t}_\varphi\mathtt{,\ t}_\psi\mathtt{)}. \end{aligned}$$

Beispielsweise wird die Formel $\big((\mathbf{1} \wedge \mathbf{0}) \vee \neg\mathbf{0}\big)$ durch den folgenden Term repräsentiert:

`o(u(1, 0), n(0))`

Betrachten Sie das folgende Logik-Programm Π:

```
true(1).
false(0).
true(n(F))      :- false(F).
false(n(F))     :- true(F).
true(o(F, G))    :- true(F).
true(o(F, G))    :- true(G).
false(o(F, G))   :- false(F), false(G).
true(u(F, G))   :- true(F), true(G).
false(u(F, G)) :- false(F).
false(u(F, G)) :- false(G).
```

(a) Geben Sie einen Beweisbaum für den Term

`true(o(u(1, 0), n(0)))`

aus Π an.

(b) Ist der folgende Term aus Π ableitbar? Begründen Sie Ihre Antwort.

`false(n(u(1,0)))`

(c) Geben Sie die Bedeutung $\mathcal{B}(\Pi)$ von Π an.

(d) Schreiben Sie ein Logik-Programm Π', so dass gilt:

$$\mathcal{B}(\Pi') \quad = \quad \Big\{ \mathtt{dual(t}_\varphi\mathtt{,\ t}_{\tilde{\varphi}}\mathtt{)} :\ \varphi \in \mathsf{AL}' \Big\}.$$

Erinnerung: Für eine Formel $\varphi \in \mathsf{AL}'$ ist $\tilde{\varphi}$ die zu φ *duale Formel*, die aus φ entsteht, indem man überall $\mathbf{0}$ durch $\mathbf{1}$, $\mathbf{1}$ durch $\mathbf{0}$, $\wedge$ durch $\vee$ und $\vee$ durch $\wedge$ ersetzt.