Prof. Dr. M. Grohe

E. Fluck, L. Härtel, T. Novotny

Übungsblatt 10 mit Lösungen

Abgabetermin: Montag, der 15. Juli 2024 um 14:30

Hausaufgabe 3 (EF Spiele) 3+4 Punkte

Geben Sie für die folgende Paare von Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ jeweils das kleinste $r \in \mathbb{N}$ an, sodass $\mathfrak{A} \not\equiv_r \mathfrak{B}$ und beschreiben Sie eine Gewinnstrategie für (H) im Spiel $\mathrm{EF}_r(\mathfrak{A}, \mathfrak{B})$, sowie eine Gewinnstrategie für (D) im Spiel $\mathrm{EF}_{r-1}(\mathfrak{A}, \mathfrak{B})$. Falls solches $r \in \mathbb{N}$ nicht existiert, geben Sie eine Gewinnstrategie für (D) im Spiel $\mathrm{EF}_r(\mathfrak{A}, \mathfrak{B})$ für jedes $r \in \mathbb{N}$.

a) $\mathfrak{A} = (\mathcal{P}(\mathbb{N}), \dot{\subseteq}^{\mathfrak{A}})$ und $\mathfrak{B} = (\mathcal{P}(\{0,1\}), \dot{\subseteq}^{\mathfrak{B}})$, wobei $\mathcal{P}(\mathbb{N})$ ist die Potenzmenge von $\mathbb{N}$ und $\mathcal{P}(\{0,1\})$ is die Potenzmenge von $\{0,1\}$. Die Relationen $\dot{\subseteq}^{\mathfrak{A}}$ und $\dot{\subseteq}^{\mathfrak{B}}$ sind wie üblich die Teilmengenrelationen.

b) $\mathfrak{A} = (\mathbb{Q}, \dot{\leq}^{\mathfrak{A}})$ und $\mathfrak{B} = (\mathbb{R}, \dot{\leq}^{\mathfrak{B}})$, wobei $\dot{\leq}^{\mathfrak{A}}$ und $\dot{\leq}^{\mathfrak{B}}$ wie üblich definiert sind.

Lösung:

a) Das kleinste r mit $\mathfrak{A} \not\equiv_r \mathfrak{B}$ ist $r = 3$.

(H) gewinnt $\mathrm{EF}_3(\mathfrak{A}, \mathfrak{B})$ indem er in $\mathfrak{A}$ die Elemente $\{0\}, \{1\}, \{2\}$ auswählt. Diese Elemente sind bezüglich $\subseteq^{\mathfrak{A}}$ paarweise unvergleichbar, und in $\mathfrak{B}$ gibt es keine drei bezüglich $\subseteq^{\mathfrak{B}}$ paarweise unverglaichbaren Elemente.

(D) gewinnt $\mathrm{EF}_2(\mathfrak{A}, \mathfrak{B})$.

- Runde 1. Es sei (H) wählt ein Element $a_1 \in \mathfrak{A}$ aus. Falls $a_1 = \emptyset$, (D) antwortet mit $b_1 = \emptyset$. Falls $a_1 = \mathbb{N}$, (D) antwortet mit $b_1 = \{0,1\}$, und sonst antwortet (D) mit $b_1 = \{0\}$.

  Es sei (H) wählt ein Element $b_1 \in \mathfrak{B}$ aus. Falls $b_1 = \emptyset$, (D) antwortet mit $a_1 = \emptyset$. Falls $b_1 = \{0,1\}$, (D) antwortet mit $a_1 = \mathbb{N}$, und sonst antwortet (D) mit $a_1 = \{0\}$.

  Die Position $\{(a_1, b_1)\}$ ist offensichtlich ein lokaler Isomorphismus, da es keine einstellige Relationen in den Strukturen gibt.

- Runde 2. Es sei (H) wählt ein Element $a_2 \in \mathfrak{A}$ aus. Falls $a_2 \subseteq a_1$, (D) wählt $b_2 = \emptyset$ aus. Falls $a_1 \subseteq a_2$, (D) wählt $b_2 = \{0,1\}$ aus. Sonst wählt (D) $b_2 = \{1\}$ aus.

  Es sei (H) wählt ein Element $b_2 \in \mathfrak{B}$ aus. Falls $b_2 \subseteq b_1$, (D) wählt $a_2 = \emptyset$ aus. Falls $b_1 \subseteq b_2$, (D) wählt $a_2 = \mathbb{N}$ aus. Sonst wählt (D) $b_2 = \{1\}$ aus.

  Die Position $\{(a_1, b_1), (a_2, b_2)\}$ ist ein lokaler Isomorphismus, da die (D) die Teilmengenbeziehung zwischen a_1, a_2 und b_1, b_2 gleich gehalten hat.

b) Wir behaupten, dass für alle $r \in \mathbb{N}$ gilt $\mathfrak{A} \equiv_r \mathfrak{B}$.

Es sei (H) wählt ein Element $a_1 \in \mathfrak{A}$ (resp. $b_1 \in \mathfrak{B}$) aus. Dann antwortet (D) mit dem Element $b_1 = 0 \in \mathfrak{B}$ (resp. $a_1 = 0 \in \mathfrak{A}$). Dann ist klar, dass die Position $\{(a_1, b_1)\}$ ein lokaler Isomorphismus ist.

Es sei das Spiel befindet sich in der Position $\{(a_1, b_1), \dots, (a_i, b_i)\}$ und (H) wählt ein Element $a_{i+1} \in \mathfrak{A}$. Da die Menge $\{a_1, \dots, a_i\}$ durch $\dot{\leq}$ linear geordnet ist, gibt es folgende Möglichkeiten.

- $a_{i+1} < \min\{a_1, \dots, a_i\}$. Dann wählt (D) das Element $b_{i+1} \coloneqq \min\{b_1, \dots, b_i\} - 1 \in \mathfrak{B}$ aus.
- $\max\{a_1, \dots, a_i\} < a_{i+1}$. Dann wählt (D) das Element $b_{i+1} \coloneqq \max\{b_1, \dots, b_i\} + 1 \in \mathfrak{B}$ aus.
- Es gibt ein $k \in [i]$ sodass $a_{i+1} = a_k$. Dann wählt (D) das Element $b_{i+1} \coloneqq b_k$.
- Sonst gibt es $k, l \in [i]$ sodass $a_k < a_{i+1} < a_l$ und
$$\{x \in \mathbb{R} \mid a_k < x < a_l\} \cap \{a_1, \dots, a_i\} = \emptyset.$$
Dann wählt (D) das Element $b_{i+1} \coloneqq \frac{b_k + b_l}{2}$.

Falls (H) ein Element $b_{i+1} \in \mathfrak{B}$ ausgewählt hat, antwortet (D) analog mit einem Element a_{i+1}.

Dann ist klar, dass die Position $\{(a_1, b_1), \dots, (a_{i+1}, b_{i+1})\}$ ein lokaler Isomorphismus ist. Also ist dies eine Gewinnstrategie für (D) im Spiel $\mathrm{EF}_r(\mathfrak{A}, \mathfrak{B})$ für jedes $r \in \mathbb{N}$ und damit gilt $\mathfrak{A} \equiv_r \mathfrak{B}$ für jedes $r \in \mathbb{N}$.

Hausaufgabe 4 (Erststufige Definierbarkeit im Endlichen) 5 Punkte

Zeigen Sie, dass die Klasse aller 3-färbbaren Graphen nicht erststufig definierbar im Endlichen ist.
Hinweis: Siehe Übungsblatt 2 für die Definition von 3-färbbarkeit von Graphen.

Lösung:

Sei $\mathcal{K} := \{\mathfrak{G} \mid \mathfrak{G} \text{ ist ein 3-färbbarer Graph}\}$. Wir beweisen die Aussage mit einer logischen Induktion wie im Beweis von Satz 7.37 aus der Vorlesung.

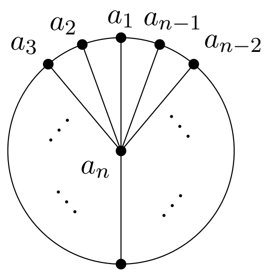
Sei $\mathcal{O}_2$ die Menge aller totalen Ordnungen gerader Länge, das heißt, die Menge aller endlichen $\{\dot{\leq}\}$-Strukturen $\mathfrak{A}_n$, sodass $\dot{\leq}^{\mathfrak{A}_n}$ eine totale Ordnung auf A ist aun $|A| \equiv 0 \pmod 2$.

Sei $\mathfrak{A}$ die $\{\dot{\leq}\}$-Struktur mit Universum $A_n = \{a_1, \dots, a_n\}$ und $a_1 \dot{\leq}^{\mathfrak{A}_n} a_2 \dot{\leq}^{\mathfrak{A}_n} \dots \dot{\leq}^{\mathfrak{A}_n} a_n$, wobei $a_i \neq a_j$ für alle $i \neq j$. Sei $\mathfrak{G}_{\mathfrak{A}_n}$ der Graph mit Knotenmenge $G_{\mathfrak{A}_n} = A_n$ und Kantenmenge

$$\begin{aligned} E^{\mathfrak{G}_{\mathfrak{A}_n}} :=& \{(a_i, a_j) \mid i, j \in [n-1] \text{ mit } |i-j| \equiv 1 \pmod{n} - 1\} \\ &\cup \{(a_n, a_i), (a_i, a_n) \mid i \in [n-1]\}. \end{aligned}$$

Der Graph sieht also wie folgt aus.

$\mathfrak{G}_{\mathfrak{A}_n}$:



Claim 1. Der Graph $\mathfrak{G}_{\mathfrak{A}_n}$ ist 3-färbbar genau dann, wenn n ungerade ist.

Beweis. Wir können eine Farbe für den Knoten v_G, und mit den zwei anderen Farben den Rest des Graphen $\mathfrak{G}$ wie in Tutoriumsaufgabe 1 färben, da $n-1$ gerade ist.

Wenn n gerade ist, ist $\mathfrak{G}_{\mathfrak{A}_n}$ nicht 3-färbbar. Angenommen es wären mit der Färbung c 3-färbbar. Die Farbe $c(a_n)$ ist eindeutig, also es gibt keinen weiteren Knoten $a_n \neq a_i \in \mathfrak{G}_{\mathfrak{A}_n}$ mit $c(a_n) = c(a_i)$. Also ist c eine 2-Färbung eines Kreises der Länge $n-1$. In Tutoriumsaufgabe 1 haben wir aber gezeigt, dass dies nicht möglich ist ($n-1$ ist ungerade). □

Wir schreiben $x \dot{<} y$ als Abkürzung für $x \dot{\leq} y \land \neg x \doteq y$. Wir definieren die folgende Formeln:

$$
\begin{aligned}
\varphi_1(x) &:= \forall z(x \dot{\leq} z) && x = a_1\\
\varphi_n(x) &:= \forall z(z \dot{\leq} x) && x = a_n\\
\varphi_{n-1}(x) &:= \big(\forall z(z \dot{\leq} x \lor \varphi_n(z))\big) \land \neg\varphi_n(x) && x = a_{n-1}\\
\varphi_{\pm 1}(x,y) &:= \big(x \dot{<} y \land \forall z(z \dot{\leq} x \lor y \dot{\leq} z)\big) && x+1 = y\\
&\quad \lor \big(y \dot{<} x \land \forall z(z \dot{\leq} y \lor x \dot{\leq} z)\big) && y+1 = x\\
\varphi_E(x,y) &:= \big(\neg\varphi_n(x) \land \neg\varphi_n(y) \to \varphi_{\pm 1}(x,y)\big) && x = y \pm 1 \text{ und } x \neq a_n \text{ und } y \neq a_n\\
&\quad \lor \big(\varphi_n(x) \leftrightarrow \neg\varphi_n(y)\big) && x = a_n\\
&\quad \lor \big(\varphi_{n-1}(x) \land \varphi_1(y)\big) \lor \big(\varphi_{n-1}(y) \land \varphi_1(x)\big)
\end{aligned}
$$

Es ist leicht zu sehen, dass für alle $a, b \in A$ gilt:

$$\mathfrak{A}_n \models \varphi_E(a,b) \iff (a,b) \in E^{\mathfrak{G}_{\mathfrak{A}_n}}.$$

Angenommen, der Satz $\psi \in L(\{E\})$ definiert die Klasse $\mathcal{K}$ im Endlichen. Sei ψ' der Satz, der aus ψ entsteht, indem man jede Subformel der Gestalt $E(z_1, z_2)$ durch die Formel $\varphi_E \frac{z_1\, z_2}{x\, y}$ ersetzt. Dann lässt sich leicht per Induktion über den Aufbau von φ zeigen, dass

$$\mathfrak{A}_n \models \psi' \iff \mathfrak{G}_{\mathfrak{A}_n} \models \psi.$$

Dann gilt

$$\mathfrak{A}_n \models \psi' \iff \mathfrak{G}_{\mathfrak{A}_n} \models \psi \iff \mathfrak{G}_{\mathfrak{A}_n} \text{ ist 3-färbbar} \iff n \text{ ist ungerade} \iff \mathfrak{A}_n \notin \mathcal{O}_2.$$

Also definiert $\neg\psi' \land \varphi_{\text{total}}$ die Klasse $\mathcal{O}_2$ im Endlichen. Das ist ein Widerspruch zum Satz 7.34.

Hausaufgabe 5 (Definierbarkeit in MSO) **3 Punkte**

Sei $\Sigma = \{a, b, c\}$, sei $L = (aba + ab)^*$, und sei

$$\mathcal{K}_L = \{\mathcal{A} \mid \text{es gibt ein } w \in L, \text{ sodass } \mathfrak{A} \cong \mathfrak{A}_w\}$$

Geben Sie einen Satz $\varphi \in \mathrm{MSO}(\sigma)$ (für ein passendes σ) an, sodass für alle σ-Strukturen $\mathfrak{A}$ gilt:

$$\mathfrak{A} \in \mathcal{K} \iff \mathfrak{A} \models \varphi.$$

Lösung:

Wir brauchen eine Formel, die besagt, dass eine σ-Struktur eine Wortstruktur ist:

$$\varphi_{\text{Wort}} \coloneqq \varphi_{\text{total endlich}} \wedge \varphi_{\text{pos}} \wedge \{\varphi_a \mid a \in \Sigma\},$$

wobei

$$\begin{aligned}
\varphi_{\text{total}} \coloneqq& \forall x(x \dot{\leq} x) \wedge \forall x \forall y \forall z(x \dot{\leq} y \wedge y \dot{\leq} z \to x \dot{\leq} z) && \text{(Reflexivität und Transitivität)}\\
\varphi_{\text{total endlich}} \coloneqq& \varphi_{\text{total}} \wedge \forall X \exists x \Big(X(x) \wedge \forall y \big(X(y) \to x \dot{\leq} y\big)\Big)\\
& \wedge \forall X \exists x \Big(X(x) \wedge \forall y \big(X(y) \to y \dot{\leq} x\big)\Big)\\
& \wedge \forall x \forall y((x \dot{\leq} y \wedge y \dot{\leq} x) \to x \doteq y) && \text{(Antisymmetrie)}\\
& \wedge \forall x \forall y(x \dot{\leq} y \vee y \dot{\leq} x), && \text{(Konnexivität)}\\
\varphi_a \coloneqq& \forall x(P_a(x) \to \bigwedge_{b \in \Sigma, b \neq a} \neg P_b(x)), && \text{(jede Position hat maximal ein Symbol)}\\
\varphi_{\text{pos}} \coloneqq& \forall x(\neg\varphi_0(x) \to (\bigvee_{a \in \Sigma} P_a(x))). && \text{(jede Position hat ein Symbol)}
\end{aligned}$$

Wir definieren die Hilfsformel

$$\begin{aligned}
\varphi_0(x) &\coloneqq \forall y(x \dot{\leq} y),\\
\varphi_{+1}(x, y) &\coloneqq \forall z \left(z \dot{\leq} y \vee x \dot{\leq} z\right),\\
\varphi_1 x &\coloneqq \exists y(\varphi_0(y) \wedge \varphi_{+1}(x, y))\\
\varphi_{+2}(x, y) &\coloneqq \exists z \left(\varphi_{+1}(z, y) \wedge \varphi_{+1}(x, z)\right),\\
\varphi_{a\text{-Partition}}(X_1, X_2) &\coloneqq \forall x\big(\neg(X_1(x) \wedge X_2(x))\big) \wedge \forall x\big(P_a(x) \leftrightarrow (X_1(x) \vee X_2(x))\big).
\end{aligned}$$

Dann können wir φ wie folgt definieren.

$$\begin{aligned}
\varphi \coloneqq \exists X_1 \exists X_2 \forall x \Big(&(\varphi_1(x) \to X_1(x)) \wedge \varphi_{a\text{-Partition}(X_1, X_2)}\\
&\wedge (X_1(x) \to \exists y(\varphi_{+1}(y, x) \wedge P_b(y)) \wedge \forall y(\varphi_{+2}(y, x) \to P_a(y)))\\
&\wedge (X_2(x) \to \forall y(\varphi_{+1}(y, x) \to X_1(y)))\Big).
\end{aligned}$$

Die erste Zeile besagt, dass es eine Partition X_1, X_2 von P_a gibt, wobei X_1 immer das "erste" a in den Teilwörtern aba oder ab enthält. Die zweite Zeile besagt, dass nach einem a in X_1 muss ein b kommen, und danach wieder ein a, oder das Wortende. Die dritte Zeile besagt, dass nach einem a in X_2 muss ein a in X_1 kommen, oder das Wortende.

Programmieraufgabe 6 (Ableitungen) 5 Punkte

- Die Abgabe der Programmieraufgabe erfolgt über **Speichern** oder **Abgabe** in VPL. Bis zur Abgabefrist könnt ihr so oft abgeben, wie ihr wollt. Wir bewerten nur die aktuellste Abgabe.
- Ihr könnt in **assignment.py** euren eigenen Code schreiben und dabei die von uns zur Verfügung gestellten Bibliotheken benutzen. Achtet allerdings darauf, keine Dateien zu löschen und die Header der Funktionen unverändert zu lassen.
- Nicht alle Importe sind möglich, manche Bibliotheken werden also einen Fehler wie z.B. `Module assignment tries to import numpy, which does not exist` liefern, wenn ihr versucht diese zu verwenden.
- Wir empfehlen, den Code mindestens einmal zu testen, mit **Ausführen** oder Strg+F11. Dies kann einige Sekunden dauern.
- Punkte und Code sind automatisch mit eurer Abgabegruppe synchronisiert.

Schreiben Sie die Funktionen
`verify_exists_left(premise: Sequent, conclusion: Sequent) -> bool` und
`verify_exists_right(premise: Sequent, conclusion: Sequent) -> bool`,
welche als Eingabe zwei Objekte der Klasse `Sequent` nehmen und als Wert `True` ausgeben, wenn die Sequenz `conclusion` mit einer Anwendung der $(\exists L)$-Regel bzw. der $(\exists R)$-Regel auf die Sequenz `premise` erhalten werden kann.

$$\frac{\texttt{premise}}{\texttt{conclusion}}$$

Schreiben Sie die Funktion `verify_proof(proof: Proof) -> bool`,
welche als Eingabe ein Objekt der Klasse `Proof` nimmt und als Wert `True` ausgibt, wenn der Beweis `proof` eine gültige Ableitung des Sequenzenkalküls darstellt.
Zur Vereinfachung können Beweise nur die Regeln (Vor), (Erw), $(\wedge L)$, $(\wedge R)$, $(\exists L)$ und $(\exists R)$ enthalten.

Lösung: