Prof. Dr. M. Grohe
E. Fluck, L. Härtel, T. Novotny

Übungsblatt 5 mit Lösungen

Abgabetermin: Montag, der 10. Juni 2024 um 14:30

Prof. Dr. M. Grohe
E. Fluck, L. Härtel, T. Novotny

Hausaufgabe 4 (Das Standardmodell der Arithmetik) — 2 Punkte

Sei $\mathfrak{N}$ das Standardmodell der Arithmetik und sei

$$\varphi(x) \coloneqq \forall y \Big(\big(y \mathrel{\dot{\neq}} \dot{0} \wedge \exists z (y \mathbin{\dot{*}} z \doteq x) \big) \to \big(y \doteq \dot{1} \vee \exists z \, ((\dot{1} \mathbin{\dot{+}} \dot{1}) \mathbin{\dot{*}} z \doteq y) \big) \Big).$$

Beschreiben Sie in natürlicher Sprache und mit möglichst wenigen Worten was die Formel $\varphi(x)$ im Standardmodell der Arithmetik besagt, das heißt ergänzen Sie die folgende Aussage um möglichst wenig Wörter:

$$\mathfrak{N} \models \varphi(a) \Leftrightarrow \ldots$$

Hinweis: Eine korrekte Lösung mit maximal 10 Wörtern gibt bereits volle Punktzahl.

Lösung:

Eine Lösung in neun Wörtern:

Jeder Teiler von a ist gleich eins oder gerade.

Eine Lösung in vier Wörtern:

a ist eine Zweierpotenz.

Hausaufgabe 5 (Eigenschaften von Strukturen) **2+4 Punkte**

a) Sei $\mathfrak{G}$ ein Graph. Geben Sie einen $\{E\}$-Satz $\varphi_{k\text{-IS}}$, sodass

$$\mathfrak{G} \models \varphi_{k\text{-IS}} \Leftrightarrow \mathfrak{G} \text{ enthält ein Independent Set der Größe } k.$$

b) Geben Sie einen $\{\dot{\circ}, e\}$-Satz φ_{Gruppe}, sodass für alle $\{\dot{\circ}, e\}$-Strukturen $\mathfrak{A}$ gilt:

$$\mathfrak{A} \models \varphi_{\text{Gruppe}} \Leftrightarrow \mathfrak{A} \text{ ist eine Gruppe.}$$

Erklären Sie ihre Idee.

Lösung:

a) $\varphi_{k\text{-IS}} \coloneqq \exists x_1 \ldots \exists x_k \left(\bigwedge_{1 \le i < j \le k} \left(x_i \mathrel{\dot{\neq}} x_j \wedge \neg E(x_i, x_j) \right) \right)$

Die Formel besagt, dass es k verschiedene Elemente gibt, die paarweise nicht verbunden sind.

b) $\varphi_{\text{Gruppe}} \coloneqq \varphi_{\text{assoziativ}} \wedge \varphi_{\text{neutral}} \wedge \varphi_{\text{inverse}}$, wobei

$$\begin{aligned} \varphi_{\text{assoziativ}} &\coloneqq \forall x \forall y \forall z \left((x \mathbin{\dot{\circ}} y) \mathbin{\dot{\circ}} z \doteq x \mathbin{\dot{\circ}} (y \mathbin{\dot{\circ}} z) \right), \\ \varphi_{\text{neutral}} &\coloneqq \forall x \left(x \mathbin{\dot{\circ}} e \doteq x \wedge e \mathbin{\dot{\circ}} x \doteq x \right), \\ \varphi_{\text{inverse}} &\coloneqq \forall x \exists y \left(x \mathbin{\dot{\circ}} y \doteq e \wedge y \mathbin{\dot{\circ}} x \doteq e \right). \end{aligned}$$

Die Formel $\varphi_{\text{assoziativ}}$ beschreibt, dass die Funktion $\dot{\circ}^{\mathfrak{A}}$ assoziativ ist, die Formel φ_{neutral} beschreibt, dass $e^{\mathfrak{A}}$ ein neutrales Element bezüglich $\dot{\circ}^{\mathfrak{A}}$ ist, und die Formel φ_{inverse} beschreibt, dass es zu jedem Element im Universum A ein Element gibt, dass sowohl links-invers als auch rechts-invers bezüglich $\dot{\circ}^{\mathfrak{A}}$ ist.

Hausaufgabe 6 (Existentiell Positive Formeln) 7 Punkte

Sei σ eine abzählbare Symbolmenge, sei Var eine abzählbare Menge an Variablen und sei

$$\Sigma_{\mathrm{EP}(\sigma)} \coloneqq \sigma \cup \mathrm{Var} \cup \{\vee, \wedge, \exists, \doteq, \ , \ , (,)\}.$$

Die Menge $\mathrm{EP}(\sigma) \subseteq \Sigma^*_{\mathrm{EP}(\sigma)}$ aller *existentiell positiven σ-Formeln der Logik der ersten Stufe* ist rekursiv wie folgt definiert.

(i) Für alle $\theta_1, \theta_2 \in \mathrm{T}(\sigma)$ ist $\theta_1 \doteq \theta_2 \in \mathrm{EP}(\sigma)$.

(ii) Für alle $k \in \mathbb{N}$, alle k-stelligen Relationssymbole $R \in \sigma$ und alle $\theta_1, \dots, \theta_k \in \mathrm{T}(\sigma)$ ist $R(\theta_1, \dots, \theta_k) \in \mathrm{EP}(\sigma)$.

(iii) Wenn $\varphi, \psi \in \mathrm{EP}(\sigma)$, dann $(\varphi \wedge \psi), (\varphi \vee \psi) \in \mathrm{EP}(\sigma)$.

(iv) Für alle $x \in \mathrm{Var}$, wenn $\varphi \in \mathrm{EP}(\sigma)$, dann $\exists x \varphi \in \mathrm{EP}(\sigma)$.

Beweisen Sie die folgende Aussage:
Seien $\mathfrak{A}, \mathfrak{B}$ σ-Strukturen und sei $\pi \colon A \to B$ ein Homomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$.

(1) Für alle σ-Terme $\theta(x_1, \dots, x_n) \in \mathrm{T}(\sigma)$ und alle $a_1, \dots, a_n \in A$ gilt

$$\pi\big(\theta^{\mathfrak{A}}(a_1, \dots, a_n)\big) \quad = \quad \theta^{\mathfrak{B}}\big(\pi(a_1), \dots, \pi(a_n)\big).$$

(2) Für alle existentiell positiven σ-Formeln $\varphi(x_1, \dots, x_n) \in \mathrm{EP}(\sigma)$ und alle $a_1, \dots, a_n \in A$ gilt

$$\mathfrak{A} \models \varphi(a_1, \dots, a_n) \quad \Rightarrow \quad \mathfrak{B} \models \varphi\big(\pi(a_1), \dots, \pi(a_n)\big).$$

Hinweis: Die Definition eines Homomorphismus ist auf Blatt 4 in Tutoriumsaufgabe 2 gegeben.
Orientieren Sie sich in ihrem Beweis am Beweis des Isomorphielemmas (4.31).

Lösung:

(1) Wir beweisen (1) per Induktion über den Aufbau von θ.

$\theta = x_i$:
Es gilt $\theta^{\mathfrak{A}}(a_1, \dots, a_n) = a_i$ und damit

$$\pi\big(\theta^{\mathfrak{A}}(a_1, \dots, a_n)\big) = \pi(a_i) = \theta^{\mathfrak{B}}\big(\pi(a_1), \dots, \pi(a_n)\big).$$

$\theta = f(\theta_1, \dots, \theta_k)$ für ein k-stelliges $f \in \sigma$ und $\theta_1, \dots, \theta_k \in \mathrm{T}(\sigma)$:
Es gilt für alle $b_1, \dots, b_k \in A$:

$$\pi\big(f^{\mathfrak{A}}(b_1, \dots, b_k)\big) = f^{\mathfrak{B}}\big(\pi(b_1), \dots, \pi(b_k)\big),$$

da π ein Homomorphismus ist. Nach Induktionsannahme gilt für alle $i \in [k]$

$$\pi\big(\theta_i^{\mathfrak{A}}(a_1, \dots, a_n)\big) \quad = \quad \theta_i^{\mathfrak{B}}\big(\pi(a_1), \dots, \pi(a_n)\big).$$

Also

$$\begin{aligned}\pi\big(\theta^{\mathfrak{A}}(a_1,\dots,a_n)\big) &= \pi\Big(f^{\mathfrak{A}}\big(\theta_1^{\mathfrak{A}}(a_1,\dots,a_n),\dots,\theta_k^{\mathfrak{A}}(a_1,\dots,a_n)\big)\Big)\\ &= f^{\mathfrak{B}}\Big(\pi\big(\theta_1^{\mathfrak{A}}(a_1,\dots,a_n)\big),\dots,\pi\big(\theta_1^{\mathfrak{A}}(a_1,\dots,a_n)\big)\Big)\\ &= f^{\mathfrak{B}}\Big(\theta_1^{\mathfrak{B}}\big(\pi(a_1),\dots,\pi(a_n)\big),\dots,\theta_k^{\mathfrak{B}}\big(\pi(a_1),\dots,\pi(a_n)\big)\Big)\\ &= \theta^{\mathfrak{B}}\big(\pi(a_1),\dots,\pi(a_n)\big).\end{aligned}$$

Das beweist Aussage (1).

(2) Wir beweisen (2) per Induktion über den Aufbau von φ.

$\varphi(x_1,\dots,x_n) = \theta_1 \doteq \theta_2$ für $\theta_1,\theta_2 \in \mathrm{T}(\sigma)$:
Für $i = 1,2$ gilt dann $\mathrm{var}(\theta_i) \subseteq \{x_1,\dots,x_n\}$, also können wir $\theta_i(x_1,\dots,x_n)$ schreiben. Nach Aussage (1) gilt $\pi\big(\theta_i^{\mathfrak{A}}(a_1,\dots,a_n)\big) = \theta_i^{\mathfrak{B}}\big(\pi(a_1),\dots,\pi(a_n)\big)$. Also

$$\begin{aligned}\mathfrak{A} \models \varphi(a_1,\dots,a_n) &\iff \theta_1^{\mathfrak{A}}(a_1,\dots,a_n) = \theta_2^{\mathfrak{A}}(a_1,\dots,a_n)\\ &\quad\Rightarrow \pi\big(\theta_1^{\mathfrak{A}}(a_1,\dots,a_n)\big) = \pi\big(\theta_2^{\mathfrak{A}}(a_1,\dots,a_n)\big)\\ &\iff \theta_1^{\mathfrak{B}}\big(\pi(a_1),\dots,\pi(a_n)\big) = \theta_2^{\mathfrak{B}}\big(\pi(a_1),\dots,\pi(a_n)\big)\\ &\iff \mathfrak{B} \models \varphi\big(\pi(a_1),\dots,\pi(a_n)\big).\end{aligned}$$

$\varphi = R(\theta_1 \dots,\theta_k)$ für ein k-stelliges $R \in \sigma$ und $\theta_1,\dots,\theta_k \in \mathrm{T}(\sigma)$:
Es gilt für alle $b_1,\dots,b_k \in A$:

$$(b_1,\dots,b_k) \in R^{\mathfrak{A}} \Rightarrow \big(\pi(b_1),\dots,\pi(b_k)\big) \in R^{\mathfrak{B}},$$

da π ein Homomorphismus ist. Für $i \in [k]$ gilt nach Aussage (1) $\pi\big(\theta_i^{\mathfrak{A}}(a_1,\dots,a_n)\big) = \theta_i^{\mathfrak{B}}\big(\pi(a_1),\dots,\pi(a_n)\big)$. Also

$$\begin{aligned}\mathfrak{A} \models \varphi(a_1,\dots,a_n) &\iff \big(\theta_1^{\mathfrak{A}}(a_1,\dots,a_n),\dots,\theta_k^{\mathfrak{A}}(a_1,\dots,a_n)\big) \in R^{\mathfrak{A}}\\ &\quad\Rightarrow \Big(\pi\big(\theta_1^{\mathfrak{A}}(a_1,\dots,a_n)\big),\dots,\pi\big(\theta_k^{\mathfrak{A}}(a_1,\dots,a_n)\big)\Big) \in R^{\mathfrak{B}}\\ &\iff \Big(\theta_1^{\mathfrak{A}'}\big(\pi(a_1),\dots,\pi(a_n)\big),\dots,\theta_k^{\mathfrak{A}'}\big(\pi(a_1),\dots,\pi(a_n)\big)\Big) \in R^{\mathfrak{B}}\\ &\iff \mathfrak{B} \models \varphi\big(\pi(a_1),\dots,\pi(a_n)\big).\end{aligned}$$

$\varphi = \psi_1 \wedge \psi_2$ für $\psi_1,\psi_2 \in \mathrm{EP}(\sigma)$:
Für $i = 1,2$ gilt dann $\mathrm{frei}(\psi_i) \subseteq \mathrm{frei}(\varphi) \subseteq \{x_1,\dots,x_n\}$, also können wir $\psi_i(x_1,\dots,x_n)$ schreiben. Nach Induktionsannahme gilt

$$\mathfrak{A} \models \psi_i(a_1,\dots,a_n) \Rightarrow \mathfrak{B} \models \psi_i\big(\pi(a_1),\dots,\pi(a_n)\big).$$

Also

$$\begin{aligned}\mathfrak{A} \models \varphi(a_1,\dots,a_n) &\iff \mathfrak{A} \models \psi_1(a_1,\dots,a_n) \text{ und } \mathfrak{A} \models \psi_2(a_1,\dots,a_n)\\ \Rightarrow \mathfrak{B} \models \psi_1\big(\pi(a_1),\dots,\pi(a_n)\big) &\text{ und } \psi_2\big(\pi(a_1),\dots,\pi(a_n)\big)\\ &\iff \mathfrak{B} \models \varphi\big(\pi(a_1),\dots,\pi(a_n)\big).\end{aligned}$$

$\varphi = \psi_1 \vee \psi_2$ für $\psi_1, \psi_2 \in \mathrm{EP}(\sigma)$:
Analog zum Fall $\varphi = \psi_1 \wedge \psi_2$.

$\varphi = \exists y \psi$ für ein $y \in \mathrm{Var}$ und ein $\psi \in \mathrm{EP}(\sigma)$:
Dann gilt $\mathrm{frei}(\psi) \subseteq \mathrm{frei}(\varphi) \cup \{y\} \subseteq \{x_1, \ldots, x_n, y\}$. Wir können also $\psi(x_1, \ldots, x_n, y)$ schreiben. Nach Induktionsannahme gilt für alle $b \in A$

$$\mathfrak{A} \models \psi(a_1, \ldots, a_n, b) \Rightarrow \mathfrak{B} \models \psi\big(\pi(a_1), \ldots, \pi(a_n), \pi(b)\big).$$

Also

$$\begin{aligned}
\mathfrak{A} \models \varphi(a_1, \ldots, a_n) \iff & \text{es gibt ein } b \in A \text{, so dass } \mathfrak{A} \models \psi(a_1, \ldots, a_n, b) \\
\Rightarrow & \text{es gibt ein } b \in A \text{, so dass } \mathfrak{B} \models \psi\big(\pi(a_1), \ldots, \pi(a_n), \pi(b)\big) \\
\Rightarrow & \text{es gibt ein } b' \in B \text{, so dass } \mathfrak{B} \models \psi\big(\pi(a_1), \ldots, \pi(a_n), b'\big) \\
\iff & \mathfrak{A}' \models \varphi\big(\pi(a_1), \ldots, \pi(a_n)\big).
\end{aligned}$$

Programmieraufgabe 7 (Terme und Atomare Formeln) 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 Funktion `evaluate_term(term: Term, interpretation: Interpretation[T]) -> T`, welche als Eingabe je ein Objekt der Klasse `Term` und ein Objekt der Klasse `Interpretation[T]` nimmt und den Wert des Terms vom Typ `T` unter der gegebenen Interpretation ausgibt, wenn diese die gleiche Symbolmenge wie der Term hat, und sich sonst beliebig verhalten darf.

Schreiben Sie anschließend die Funktion `evaluate_atom(atom: AtomicFormula, interpretation: Interpretation[Any]) -> bool`, welche als Eingabe je ein Objekt der Klasse `AtomicFormula` und ein Objekt der Klasse `Interpretation[Any]` nimmt und den Wahrheitswert der atomaren Formel unter der gegebenen Interpretation ausgibt, wenn diese die gleiche Symbolmenge wie die atomare Formel hat, und sich sonst beliebig verhalten darf.

Für Fragen, z.B. zum neuen Modul `first_order.py`, stehen wir wie immer im Diskussionsforum zur Verfügung!

Lösung:

Diese Aufgabe ist vergleichbar mit der Bonusaufgabe von Blatt 0: mit Python kann man eine Lösung schreiben, die sehr viel Arbeit den eingebauten Funktionen überlässt.

Für `evaluate_term` verwenden wir ein `match`-Statement um zwischen `Variable` und `FunctionTerm(function_symbol, arguments)` zu unterscheiden. Diese lösen wir rekursiv auf, indem wir die entsprechende Funktion `interpretation.structure.functions[function_symbol]` auf `evaluate_term` von `arguments` aufrufen.

Die `evaluate_atom`-Funktion ist ziemlich ähnlich mit je einem `match`-Case für `Equality`, `RelationFormula(relation_symbol, terms)` und `TruthConstant(value)`. Hierbei verhält sich das `relation_symbol` wie eine Funktion, die auf `evaluate_term` von `terms` aufgerufen werden kann.