Fachbereich Informatik
Modellierung und Analyse
von Informationssystemen
Prof. Dr.-Ing. Heiko Mantel

TECHNISCHE UNIVERSITÄT DARMSTADT

Klausur für die Prüfung

Modellierung, Spezifikation und Semantik

Sommersemester 2020 – 18.9.2020

Nachname:	
Vorname:	
Matrikelnummer:	
Studiengang:	
E-mail:	
Unterschrift:	

Bitte lesen Sie die Klausurregeln auf der folgenden Seite.

Viel Erfolg!

Aufgabe	1	2	3	4	5	6	$\sum$
Punkte	(von 19)	(von 18)	(von 18)	(von 16)	(von 11)	(von 18)	(von 100)

Note: _____

Name: _____, Matrikelnr.: _____

Beiblatt zur Klausur
Modellierung, Spezifikation und Semantik

IMP

Kalkül $\mathcal{A}$: Regeln für die Herleitung des Urteils $\langle a,\ \sigma\rangle \Downarrow n$ für $a \in \mathrm{AExp}$

$$\text{rNum}\ \frac{}{\langle n,\ \sigma\rangle \Downarrow n} \qquad \text{rVar}\ \frac{}{\langle X,\ \sigma\rangle \Downarrow n}\ \sigma(X)=n \qquad \text{r}\oplus\ \frac{\langle a_1,\ \sigma\rangle \Downarrow n_1 \quad \langle a_2,\ \sigma\rangle \Downarrow n_2}{\langle (a_1 \oplus a_2),\ \sigma\rangle \Downarrow n}\ n = n_1 + n_2$$

$$\text{r}\ominus\ \frac{\langle a_1,\ \sigma\rangle \Downarrow n_1 \quad \langle a_2,\ \sigma\rangle \Downarrow n_2}{\langle (a_1 \ominus a_2),\ \sigma\rangle \Downarrow n}\ n = n_1 - n_2 \qquad \text{r}\otimes\ \frac{\langle a_1,\ \sigma\rangle \Downarrow n_1 \quad \langle a_2,\ \sigma\rangle \Downarrow n_2}{\langle (a_1 \otimes a_2),\ \sigma\rangle \Downarrow n}\ n = n_1 \cdot n_2$$

Kalkül $\mathcal{B}$: Regeln für die Herleitung des Urteils $\langle b,\ \sigma\rangle \Downarrow t$ für $b \in \mathrm{BExp}$

$$\text{rtrue}\ \frac{}{\langle \texttt{true},\ \sigma\rangle \Downarrow \texttt{true}} \qquad \text{rfalse}\ \frac{}{\langle \texttt{false},\ \sigma\rangle \Downarrow \texttt{false}}$$

$$\text{reqt}\ \frac{\langle a_1,\ \sigma\rangle \Downarrow n_1 \quad \langle a_2,\ \sigma\rangle \Downarrow n_2}{\langle (a_1\ \texttt{eq}\ a_2),\ \sigma\rangle \Downarrow \texttt{true}}\ n_1 = n_2 \qquad \text{reqf}\ \frac{\langle a_1,\ \sigma\rangle \Downarrow n_1 \quad \langle a_2,\ \sigma\rangle \Downarrow n_2}{\langle (a_1\ \texttt{eq}\ a_2),\ \sigma\rangle \Downarrow \texttt{false}}\ \neg(n_1 = n_2)$$

$$\text{rleqt}\ \frac{\langle a_1,\ \sigma\rangle \Downarrow n_1 \quad \langle a_2,\ \sigma\rangle \Downarrow n_2}{\langle (a_1\ \texttt{leq}\ a_2),\ \sigma\rangle \Downarrow \texttt{true}}\ n_1 \le n_2 \qquad \text{rleqf}\ \frac{\langle a_1,\ \sigma\rangle \Downarrow n_1 \quad \langle a_2,\ \sigma\rangle \Downarrow n_2}{\langle (a_1\ \texttt{leq}\ a_2),\ \sigma\rangle \Downarrow \texttt{false}}\ \neg(n_1 \le n_2)$$

$$\text{rnott}\ \frac{\langle b,\ \sigma\rangle \Downarrow \texttt{false}}{\langle \texttt{not}\ b,\ \sigma\rangle \Downarrow \texttt{true}} \qquad \text{rnotf}\ \frac{\langle b,\ \sigma\rangle \Downarrow \texttt{true}}{\langle \texttt{not}\ b,\ \sigma\rangle \Downarrow \texttt{false}}$$

$$\text{randt}\ \frac{\langle b_1,\ \sigma\rangle \Downarrow \texttt{true} \quad \langle b_2,\ \sigma\rangle \Downarrow \texttt{true}}{\langle (b_1\ \texttt{and}\ b_2),\ \sigma\rangle \Downarrow \texttt{true}} \qquad \text{randf1}\ \frac{\langle b_1,\ \sigma\rangle \Downarrow \texttt{false}}{\langle (b_1\ \texttt{and}\ b_2),\ \sigma\rangle \Downarrow \texttt{false}} \qquad \text{randf2}\ \frac{\langle b_2,\ \sigma\rangle \Downarrow \texttt{false}}{\langle (b_1\ \texttt{and}\ b_2),\ \sigma\rangle \Downarrow \texttt{false}}$$

$$\text{rort1}\ \frac{\langle b_1,\ \sigma\rangle \Downarrow \texttt{true}}{\langle (b_1\ \texttt{or}\ b_2),\ \sigma\rangle \Downarrow \texttt{true}} \qquad \text{rort2}\ \frac{\langle b_2,\ \sigma\rangle \Downarrow \texttt{true}}{\langle (b_1\ \texttt{or}\ b_2),\ \sigma\rangle \Downarrow \texttt{true}} \qquad \text{rorf}\ \frac{\langle b_1,\ \sigma\rangle \Downarrow \texttt{false} \quad \langle b_2,\ \sigma\rangle \Downarrow \texttt{false}}{\langle (b_1\ \texttt{or}\ b_2),\ \sigma\rangle \Downarrow \texttt{false}}$$

Kalkül $\mathcal{C}$: Regeln für die Herleitung des Urteils $\langle c,\ \sigma\rangle \to \sigma'$ für $c \in \mathrm{Com}$.

$$\text{rsk}\ \frac{}{\langle \texttt{skip},\ \sigma\rangle \to \sigma} \qquad \text{r:=}\ \frac{\langle a,\ \sigma\rangle \Downarrow n}{\langle X := a,\ \sigma\rangle \to \sigma'}\ \sigma' = \sigma[X \backslash n] \qquad \text{r;}\ \frac{\langle c_1,\ \sigma\rangle \to \sigma'' \quad \langle c_2,\ \sigma''\rangle \to \sigma'}{\langle c_1\,;\,c_2,\ \sigma\rangle \to \sigma'}$$

$$\text{rift}\ \frac{\langle b,\ \sigma\rangle \Downarrow \texttt{true} \quad \langle c_1,\ \sigma\rangle \to \sigma'}{\langle \texttt{if}\ b\ \texttt{then}\ c_1\ \texttt{else}\ c_2\ \texttt{fi},\ \sigma\rangle \to \sigma'} \qquad \text{riff}\ \frac{\langle b,\ \sigma\rangle \Downarrow \texttt{false} \quad \langle c_2,\ \sigma\rangle \to \sigma'}{\langle \texttt{if}\ b\ \texttt{then}\ c_1\ \texttt{else}\ c_2\ \texttt{fi},\ \sigma\rangle \to \sigma'}$$

$$\text{rwht}\ \frac{\langle b,\ \sigma\rangle \Downarrow \texttt{true} \quad \langle c,\ \sigma\rangle \to \sigma'' \quad \langle \texttt{while}\ b\ \texttt{do}\ c\ \texttt{od},\ \sigma''\rangle \to \sigma'}{\langle \texttt{while}\ b\ \texttt{do}\ c\ \texttt{od},\ \sigma\rangle \to \sigma'} \qquad \text{rwhf}\ \frac{\langle b,\ \sigma\rangle \Downarrow \texttt{false}}{\langle \texttt{while}\ b\ \texttt{do}\ c\ \texttt{od},\ \sigma\rangle \to \sigma}$$

CSP

Syntax von CSP

Ein Prozess wird durch einen Prozessausdruck aus der Sprache

$$P ::= \mathrm{STOP}_E \mid \mathrm{SKIP}_E \mid (x \to P) \mid (P \sqcap P) \mid (P; P) \mid (P \parallel P) \mid (P \,|||\, P) \mid id \mid (\mu X{:}\,\mathrm{E}.F(X))$$

und durch eine Menge von Gleichungen der Form

$$\{id =_E P \mid id \in \mathrm{Id}\}$$

spezifiziert, wobei Id eine Menge von Prozessbezeichnern ist.

Semantik von CSP

Prämisse	Alphabet	Menge der Spuren
	$\alpha\mathrm{STOP}_E = E$	$\mathrm{traces}(\mathrm{STOP}_E) = \{\langle\rangle\}$
	$\alpha\mathrm{SKIP}_E = E \cup \{\checkmark\}$	$\mathrm{traces}(\mathrm{SKIP}_E) = \{\langle\rangle, \langle\checkmark\rangle\}$
$x \in \alpha P$	$\alpha(x \to P) = \alpha P$	$\mathrm{traces}((x \to P)) = \{\langle\rangle\} \cup \{\langle x\rangle \frown t \mid t \in \mathrm{traces}(P)\}$
$\alpha P = \alpha Q$	$\alpha(P \sqcap Q) = \alpha P$	$\mathrm{traces}((P \sqcap Q)) = \mathrm{traces}(P) \cup \mathrm{traces}(Q)$
$\alpha P = \alpha Q$	$\alpha(P; Q) = \alpha P$	$\mathrm{traces}((P; Q)) = \{s \in \mathrm{traces}(P) \mid \checkmark \text{kommt in } s \text{ nicht vor}) \cup \{s \frown t \mid s \frown \langle\checkmark\rangle \in \mathrm{traces}(P) \wedge t \in \mathrm{traces}(Q)\}$
	$\alpha(P \parallel Q) = \alpha P \cup \alpha Q$	$\mathrm{traces}((P \parallel Q)) = \{t \in (\alpha P \cup \alpha Q)^* \mid (t \upharpoonright \alpha P) \in \mathrm{traces}(P) \wedge (t \upharpoonright \alpha Q) \in \mathrm{traces}(Q)\}$
$\alpha P = \alpha Q$	$\alpha(P \,\|||\, Q) = \alpha P$	$\mathrm{traces}((P \,\|||\, Q)) = \{s \in (\alpha P \cup \alpha Q)^* \mid \exists t \in \mathrm{traces}(P): \exists u \in \mathrm{traces}(Q): s \in \mathrm{interleaving}(t, u)\}$

wobei

$$\mathrm{interleaving} : (\alpha P)^* \times (\alpha P)^* \to \mathcal{P}((\alpha P)^*),$$

$$\mathrm{interleaving}(t, u) := \begin{cases} \{t\} & \text{wenn } u = \langle\rangle, \\ \{u\} & \text{wenn } t = \langle\rangle, \\ \{\langle x\rangle \frown s \mid s \in \mathrm{interleaving}(t', u)\} \cup \{\langle y\rangle \frown s \mid s \in \mathrm{interleaving}(t, u')\} & \text{wenn } t = \langle x\rangle \frown t' \text{ und } u = \langle y\rangle \frown u' \end{cases}$$

Intuition von CSP

Prozessausdruck	Intuition des Prozessausdrucks
STOP_E	spezifiziert, dass gar nichts passiert.
SKIP_E	spezifiziert, dass der Prozess terminiert.
$(x \to P)$	spezifiziert, dass der Prozess zuerst am Ereignis x teilnimmt, und sich anschließend wie der durch P spezifizierte Prozess verhält.
$(P \sqcap Q)$	spezifiziert, dass der Prozess sich entweder wie der Prozess P oder wie der Prozess Q verhält. Die Systemumgebung hat keine Kontrolle, ob sich der Prozess entweder wie P oder wie Q verhält.
$(P; Q)$	spezifiziert den Prozess, der sich zunächst wie der durch P spezifizierte Prozess verhält und, nachdem dieser terminiert wäre, sich wie der Prozess verhält, der durch Q spezifiziert ist.
$(P \parallel Q)$	spezifiziert einen Prozess, der die Prozesse, die durch P und Q spezifiziert sind, nebenläufig ablaufen läßt, wobei sich die Prozesse bei Ereignissen, die in beiden Alphabeten vorkommen synchronisieren.
$(P \,\|||\, Q)$	spezifiziert einen Prozess, der die Prozesse, die durch P und Q spezifiziert sind, nebenläufig ablaufen läßt, ohne dass sich die Prozesse synchronisieren.

Name: _____, Matrikelnr.: _____

Klausurregeln

- Für die Klausur haben Sie **90 Minuten** Zeit.
- Sobald Sie eine Klausur erhalten haben, dürfen Sie den Raum bis zum Ende der Klausur nicht mehr verlassen. **Ausnahme:** Wenn Sie das WC benutzen wollen, melden Sie sich. Verlassen Sie Ihren Platz *nicht* unaufgefordert. Es darf immer nur ein Studierender außerhalb des Raumes sein.
- Schreiben Sie mit Füller oder Kugelschreiber, *nicht* mit Bleistift. Schreiben Sie mit Schwarz oder Blau, *nicht* mit Rot. Schreiben Sie leserlich. Nicht lesbare Antworten werden nicht bewertet.
- Schreiben Sie auf jedes Blatt Ihren **Namen** und Ihre **Matrikelnummer**. Fällt eine Klausur auseinander, so werden Blätter, auf denen kein Name und keine Matrikelnummer stehen, nicht gewertet.
- Antworten Sie bei Aufgaben ohne Vorgaben (wie z. B. Boxen oder zu definierende Symbole) **in ganzen Sätzen**. Schreiben Sie Ihre Antworten in die freien Plätze nach den Teilaufgaben.
- Wenn Ihnen der Platz auf den Klausurblättern nicht ausreicht, erhalten Sie von uns zusätzliches Papier. Verwenden Sie **kein eigenes Papier**. Schreiben Sie Ihren Namen und Ihre Matrikelnummer auf jedes Blatt. Kennzeichnen Sie deutlich, zu welcher Aufgabe welche Antwort gehört.
- Wenn Sie **Schmierpapier** benötigen, erhalten Sie dieses von uns. Verwenden Sie kein eigenes Papier. Schreiben Sie Ihren Namen und Ihre Matrikelnummer auf jedes Blatt. Schreiben Sie „nicht bewerten" gut sichtbar oben auf das Schmierpapier.
- Am **Ende der Klausur** müssen Sie jegliches Papier mit Ausnahme Ihres vorgefertigten Blattes, das als Hilfsmittel erlaubt ist, abgeben. Ein Fortsetzen der Bearbeitung nach Ende der Klausur ist untersagt.
- Als **Hilfsmittel** sind ein beidseitig handbeschriebenes DIN-A4-Blatt (*kein* Ausdruck, *keine* Kopien, muss deutlich lesbar mit Ihrem Namen und Ihrer Matrikelnummer am oberen Rand auf beiden Seiten markiert sein) und eine Uhr zugelassen. Es sind *keine* Folien, Bücher, elektronischen Geräte, Gesprächspartner, usw. zugelassen. Schalten Sie Ihre Mobiltelefone, Smartwatches o.Ä aus. Mobiltelefone, Smartwatches, o.Ä. müssen während der gesamten Klausur abgeschaltet sein und dürfen nicht in unmittelbarer Reichweite aufbewahrt werden. Betriebsbereite Mobiltelefone, Smartwatches, o.Ä. werden als Täuschungsversuch gewertet.
- **Täuschungsversuche** sind zu unterlassen. Versuchte Täuschung kann zu Nichtbestehen der Klausur führen und ggf. weitere Konsequenzen haben. Bei einem Täuschungsversuch kann die Aufsicht die bis dahin erzielten Ergebnisse einsammeln. Die Klausur wird dann höchstens „unter Vorbehalt" weitergeführt.
- **Äußere Störungen** sind unverzüglich bei der Aufsicht zu rügen.

Besondere Klausurregeln

- Prüfen Sie zu Beginn der Klausur, dass sie 19 **paarweise verschiedene Seiten** der Klausur, ein Beiblatt und zwei Zusatzbögen haben.
- Lassen Sie alle Klausurunterlagen während Toilettengängen verdeckt auf Ihrem Platz liegen.
- Sollten Sie aus gesundheitlichen Gründen die Klausur abbrechen, hinterlassen Sie den Grund Ihres Abbruchs bitte schriftlich auf einem der Zusatzblätter und signalisieren Sie der Aufsicht Ihren Abruch. Lassen Sie alle Klausurunterlagen verdeckt auf Ihrem Platz liegen. Bitte setzen Sie sich unmittelbar nach Abbruch mit Ihrem Studienbüro zum weiteren Vorgehen in Verbindung, insbesondere bezüglich eines eventuell benötigten ärztlichen Attests. Geben Sie dabei insbesondere an, dass Sie die Klausur nach Beginn der Bearbeitung abgebrochen haben. Die Entscheidung über den Rücktritt trifft die Prüfungskommission.

Bitte lesen Sie auch den Auszug aus den Corona-Richtlinien der TU Darmstadt auf der nächsten Seite.

Name: _____, Matrikelnr.: _____

Auszug aus den Corona-Richtlinien der TU Darmstadt

- In Situationen, in denen Maßnahmen der physischen Distanzierung nur schwer eingehalten werden können, wird das Tragen einer Mund-Nasen-Bedeckung dringend empfohlen.
- Beachten Sie bitte die Einhaltung des Mindestabstandes beim Eintreten und Verlassen der Gebäude und der Prüfungsräume.
- Die Identitätsfeststellung und Anwesenheitsüberprüfung wird von den Aufsichtspersonen an Ihren Prüfungsplätzen vorgenommen. Legen Sie zur Feststellung Ihrer Identität Ihren Studierendenausweis und Ihren Personalausweis an den freien Platz rechts neben sich. Sollte sich rechts neben Ihnen kein Platz befinden, dann bitte links. Das Aufsichtspersonal wird die freien Reihen zwischen Ihnen nutzen, um die Kontrolle durchzuführen. Bitte entfernen Sie bei der Identitätsfeststellung kurz Ihre Mund-Nasen-Bedeckung.
- Bitte halten Sie sich an die Anweisungen des aufsichtsführenden Personen zum Ablauf (Beginn, Ende). Aufgrund der besonderen Situation ist die Beantwortung individueller Nachfragen zur Aufgabenstellung durch einzelne Prüflinge während der Klausur leider nicht möglich.
- Nach Verlassen der Prüfungsräume begeben Sie sich bitte wieder zügig und einzeln zu den Gebäudeausgängen und vermeiden Sie Ansammlungen.

Bitte beachten Sie, dass dieser Auszug der Corona-Richtlinien nicht vollständig ist. Auch alle weiteren von der TU Darmstadt ausgegebenen Richtlinien sind zu befolgen.

Name: _____, Matrikelnr.: _____

Aufgabe 1: Formale Modellierung mit Symbolen, Mengen und Funktionen (19 Punkte)

In dieser Aufgabe sollen Sie das folgende Modell einer Internet-of-Things (IoT) Infrastruktur betrachten. In der IoT-Infrastruktur werden verschiedene Geräte betrieben, die jeweils einen von drei Gerätetypen haben. Geräte werden von Nutzern verwendet. Die IoT-Infrastruktur wird durch folgende Mengen und Funktionen modelliert:

GERÄT	ist eine Menge, welche die Geräte modelliert, die in der IoT-Infrastruktur betrieben werden.
GERÄTE-TYP $:= \{sens, track, mon\}$	modelliert die verschiedenen Typen, die ein Gerät haben kann, wobei *sens* den Typ Sensor modelliert, *track* den Typ Tracker modelliert und *mon* den Typ Monitor modelliert.
NUTZER	ist eine Menge, welche die Nutzer der IoT-Infrastruktur modelliert.
typ-von : GERÄT $\rightarrow$ GERÄTE-TYP	modelliert für jedes Gerät, von welchem Typ das Gerät ist.
nutzer-von : GERÄT $\rightarrow \mathcal{P}$(NUTZER)	modelliert für jedes Gerät, welche Nutzer das Gerät verwenden.

Ansonsten bleiben diese Mengen und Funktionen unterspezifiziert.

- **Notationskonvention:** Sie können die aus den Übungen bekannte Schreibweise (x, xs) als Abkürzung für die Folge $(x, x_1, ..., x_n)$ verwenden, wenn $xs = (x_1, ..., x_n)$ gilt.

- **Zur Information:** Alle vier nachfolgenden Aufgabenteile sind unabhängig voneinander lösbar. Sie dürfen in Ihrer Lösung Mengen, Relationen und Funktionen wiederverwenden, die in den vorigen Aufgabenstellungen deklariert wurden, auch wenn Sie diese Aufgabenteile nicht bearbeitet haben.

(3P) (A). Modellieren Sie, dass ein Gerät von einem bestimmten Typ ist durch formale Definition einer Funktion ist-von-typ : (GERÄT $\times$ GERÄTE-TYP) $\rightarrow \{\mathfrak{w}, \mathfrak{f}\}$. Die Funktion soll für ein Gerät g und einen Typ t genau dann $\mathfrak{w}$ zurückgeben, falls das Gerät g vom Typ t ist, und genau dann $\mathfrak{f}$ zurückgeben, falls das Gerät g nicht vom Typ t ist.

Antwort:

(4P) (B). Modellieren Sie welche Geräte der IoT-Infrastruktur vom Typ Monitor sind durch formale Definition einer Menge MONITOR $\subseteq$ GERÄT.

Antwort:

Name: _____, Matrikelnr.: _____

(C). Modellieren Sie, dass zwei Geräte von den selben Nutzern verwendet werden durch formale Definition einer binären Äquivalenzrelation SELBE-NUTZER $\subseteq$ (GERÄT $\times$ GERÄT).

Antwort:

(D). Modellieren Sie, welche Geräte einer gegebenen Liste von Geräten gemeinsame Nutzer mit einem weiteren gegebenen Gerät haben durch formale Definition einer rekursiven Funktion gemeinsame-nutzer : (GERÄT $\times$ GERÄT*) $\to$ GERÄT*. Die Funktion soll, gegeben ein Gerät g und eine Folge von Geräten gs, eine Folge aller Geräte aus der Folge gs zurückgeben, die von mindestens einem Nutzer verwendet werden, der auch das Gerät g verwendet.

Antwort:

Name: _____, Matrikelnr.: _____

Aufgabe 2: Formale Modellierung von Anforderungen (18 Punkte)

In dieser Aufgabe sollen Sie Anforderungen an das Modell der IoT-Infrastruktur aus Aufgabe 1 modellieren, das um die folgende Menge und die folgende Funktion erweitert ist:

$\mathsf{MONITOR} \subseteq \mathsf{GERÄT}$	ist die in Aufgabe 1(B) eingeführte Menge aller Monitore, die in der IoT-Infrastruktur betrieben werden. In dieser Aufgabe ist ein Monitor ein Gerät, das andere Geräte überwacht.
$\text{überwacht} : \mathsf{MONITOR} \to \mathcal{P}(\mathsf{GERÄT})$	modelliert für jeden Monitor, welche Geräte von diesem Monitor überwacht werden.

Ansonsten bleiben diese Menge und diese Funktion unterspezifiziert.

▸ **Zur Information:** Sie können diese Aufgabe auch dann lösen, wenn Sie Aufgabe 1 nicht bearbeitet haben. Alle vier nachfolgenden Aufgabenteile sind unabhängig voneinander lösbar. Sie dürfen in Ihrer Lösung alle Mengen, Relationen und Funktionen wiederverwenden, die in der Aufgabenstellung von Aufgabe 1 (inklusive Teilaufgaben) deklariert wurden, auch wenn Sie Aufgabe 1 nicht vollständig bearbeitet haben.

(3 P) (A). Die Anforderung „Jeder Monitor überwacht maximal 10 Geräte." sei durch folgende Relation modelliert:

$\mathsf{ANFORDERUNG1} \subseteq (\mathsf{MONITOR} \to \mathcal{P}(\mathsf{GERÄT}))$

$\text{überwacht} \in \mathsf{ANFORDERUNG1}$ genau dann, wenn die prädikatenlogische Formel φ_1 gilt.

Definieren Sie die Formel φ_1 in Prädikatenlogik mit Hilfe von mathematischen Konzepten, die in der Vorlesung behandelt wurden.

Antwort:

$\varphi_1 :=$

(4 P) (B). Die Anforderung „Jedes Gerät wird von mindestens einem Monitor überwacht." sei durch folgende Relation modelliert:

$\mathsf{ANFORDERUNG2} \subseteq (\mathsf{MONITOR} \to \mathcal{P}(\mathsf{GERÄT}))$

$\text{überwacht} \in \mathsf{ANFORDERUNG2}$ genau dann, wenn die prädikatenlogische Formel φ_2 gilt.

Definieren Sie die Formel φ_2 in Prädikatenlogik mit Hilfe von mathematischen Konzepten, die in der Vorlesung behandelt wurden.

Antwort:

$\varphi_2 :=$

(C). Die Anforderung „Jeder Sensor wird von mindestens zwei verschiedenen Monitoren überwacht." sei durch folgende Relation modelliert: (5 P)

$\mathsf{ANFORDERUNG3} \subseteq (\mathsf{MONITOR} \to \mathcal{P}(\mathsf{GERÄT}))$

überwacht $\in \mathsf{ANFORDERUNG3}$ genau dann, wenn die prädikatenlogische Formel φ_3 gilt.

Definieren Sie die Formel φ_3 in Prädikatenlogik mit Hilfe von mathematischen Konzepten, die in der Vorlesung behandelt wurden.

Antwort:

$\varphi_3 :=$

(D). Die Anforderung „Wenn ein Monitor mehr als 5 Geräte überwacht, dann wird jedes Gerät, das von diesem Monitor überwacht wird, von mindestens einem weiteren Monitor überwacht." sei durch folgende Relation modelliert: (6 P)

$\mathsf{ANFORDERUNG4} \subseteq (\mathsf{MONITOR} \to \mathcal{P}(\mathsf{GERÄT}))$

überwacht $\in \mathsf{ANFORDERUNG4}$ genau dann, wenn die prädikatenlogische Formel φ_4 gilt.

Definieren Sie die Formel φ_4 in Prädikatenlogik mit Hilfe von mathematischen Konzepten, die in der Vorlesung behandelt wurden.

Antwort:

$\varphi_4 :=$

Name: _____, Matrikelnr.: _____

Aufgabe 3: Syntax und Semantik (18 Punkte)

- **Zur Information:** Alle zwei nachfolgenden Aufgabenteile sind unabhängig voneinander lösbar.
- **Zur Information:** Alle Kalkülregeln der operationalen Semantik der Programmiersprache IMP finden Sie zusammengefasst auf dem Beiblatt „Beiblatt zur Klausur Modellierung, Spezifikation und Semantik".

(6 P) (A). Sei $\mathsf{Var} := \{\mathtt{x}\}$. Seien $\sigma, \sigma' \in \Sigma$ beliebige Zustände, so dass $\sigma(\mathtt{x}) = 3$ und $\sigma' = \sigma[x \backslash 6]$ gelten.

- Vervollständigen Sie die nachfolgende Herleitung des Urteils

$$\langle \mathtt{x} := (\mathtt{x} \odot 2); \mathtt{skip},\ \sigma \rangle \to \sigma'$$

im Kalkül $\mathcal{C}$ für die Programmiersprache IMP. Füllen Sie für Ihre Antwort nur die untenstehenden Boxen aus und modifizieren Sie keine anderen Teile der Aufgabenstellung.

Antwort:

$$\dfrac{\square \qquad \vdots\,\mathcal{H}_1 \qquad \square}{\langle \mathtt{x} := (\mathtt{x} \odot 2),\ \sigma \rangle \to \sigma' \qquad\qquad \langle \mathtt{x} := (\mathtt{x} \odot 2); \mathtt{skip},\ \sigma \rangle \to \sigma'}$$

wobei $\mathcal{H}_1$ die folgende Herleitung ist:

$$\dfrac{\square \qquad \square \qquad \square}{\langle \mathtt{x} := (\mathtt{x} \odot 2),\ \sigma \rangle \to \sigma'}$$

Name: ______________________, Matrikelnr.: ______________

(B). In dieser Aufgabe definieren Sie Kalkülregeln zur Herleitung von Kommandos in einer modifizierten Form der Programmiersprache IMP. Folgende Wertebereiche werden verwendet: (12 P)

Num	die Zahlen,	AExp	die arithmetischen Ausdrücke,
Bool	die Wahrheitswerte,	BExp	die Booleschen Ausdrücke,
Var	die Programmvariablen,	MCom	die Kommandos.

Der Wertebereich MCom ist durch folgende Grammatik in BNF definiert, wobei $a \in$ AExp und $X \in$ Var:

$$c ::= \texttt{skip} \mid X := a \mid c; c \mid \texttt{if-unchanged}\ X\ \texttt{in}\ c\ \texttt{then}\ c\ \texttt{ufi}$$

Das heißt, dass MIMP das Kommando `if-unchanged` X `in` c_1 `then` c_2 `ufi` zusätzlich zu den Kommandos von IMP enthält und dass MIMP die Kommandos `if` b `then` c_1 `else` c_2 `fi` und `while` b `do` c `od` von IMP **nicht** enthält.

Die Intuition des Kommandos `if-unchanged` X `in` c_1 `then` c_2 `ufi` ist:

- **Fall 1:** Wenn die Variable X im Zustand vor der Ausführung des Kommandos c_1 zum selben Wert auswertet wie im Zustand nach der Ausführung des Kommandos c_1, dann wird das Kommando c_2 im Zustand vor der Ausführung des Kommandos c_1 ausgewertet.
- **Fall 2:** Wenn die Variable X im Zustand vor der Ausführung des Kommandos c_1 zu einem anderen Wert auswertet als im Zustand nach der Ausführung des Kommandos c_1, dann ändert die Ausführung von `if-unchanged` X `in` c_1 `then` c_2 `ufi` den ursprünglichen Zustand nicht.

■ **Erweitern Sie den Kalkül** für die Herleitung von Instanzen des Urteils $\langle c, \sigma\rangle \to \sigma'$ (siehe Beiblatt) **um Kalkülregeln**, mit denen Instanzen des Urteils $\langle \texttt{if-unchanged}\ X\ \texttt{in}\ c_1\ \texttt{then}\ c_2\ \texttt{ufi}, \sigma\rangle \to \sigma'$ für die Sprache MIMP hergeleitet werden können. Dabei sollen die Kalkülregeln die im Aufgabentext beschriebene Intuition des neu hinzugefügten Kommandos `if-unchanged` X `in` c_1 `then` c_2 `ufi` angemessen modellieren.

Die Kommandos `if` b `then` c_1 `else` c_2 `fi` und `while` b `do` c `od` sind **nicht** Teil der Programmiersprache MIMP und können daher **nicht** in Prämissen der definierten Kalkülregeln verwendet werden.

Sie brauchen in dieser Aufgabe **nicht** für die Angemessenheit der Kalkülregeln zu argumentieren.

► **Zur Information:** Es existiert eine angemessene Lösung mit zwei Kalkülregeln.

Antwort:

Name: _____, Matrikelnr.: _____

Aufgabe 4: Determinismus von Programmiersprachen (16 Punkte)

In dieser Aufgabe sollen Sie einen Teilbeweis für den Determinismus von Ausführungen der Programmiersprache ASM für Stackoperationen führen. Die Syntax und Semantik der Programmiersprache ASM wird im restlichen Aufgabentext definiert. Dabei werden folgende Wertebereiche verwendet:

$$\begin{array}{ll} \mathsf{Val} & \text{die Werte,} \\ \Sigma_{\mathrm{ASM}} := \mathsf{Val}^* & \text{die Stackzustände,} \end{array} \quad \Big| \quad \mathsf{Com}_{\mathrm{ASM}} \quad \text{die Kommandos.}$$

Syntax. Der Wertebereich $\mathsf{Com}_{\mathrm{ASM}}$ ist durch folgende Grammatik in BNF definiert, wobei $v \in \mathsf{Val}$:

$$c ::= \mathsf{push}\ v \mid \mathsf{add}\ v \mid c\,;c$$

Die Intuition der einzelnen Kommandos ist:

- $\mathsf{push}\ v$ legt den Wert v oben auf den Stack und lässt den Stack ansonsten unverändert
- $\mathsf{add}\ v$ kann nur bei einem nicht-leeren Stack ausgeführt werden und verändert das oberste Element eines nicht-leeren Stacks durch Addition von v und lässt den Stack ansonsten unverändert
- $c_1;c_2$ führt zunächst das Kommando c_1 aus und anschließend das Kommando c_2

► **Notationskonvention**: Die aus den Übungen bekannte Schreibweise (x, xs) wird als Abkürzung für die Folge $(x, x_1, \ldots, x_n)$ verwendet wenn $xs = (x_1, \ldots, x_n)$ gilt.

Semantik. Die Semantik von ASM ist mit Hilfe des Urteils $\langle c,\ \sigma\rangle \rightrightarrows \sigma'$ definiert. Die Intuition des Urteils $\langle c,\ \sigma\rangle \rightrightarrows \sigma'$ ist, dass das Kommando c im Stackzustand σ zum Stackzustand σ' auswertet.
Der folgende Kalkül definiert die Semantik von ASM:

$$\mathsf{rASMpush}\ \frac{}{\langle \mathsf{push}\ v,\ \sigma\rangle \rightrightarrows \sigma'}\ \sigma' = (v,\sigma) \qquad \mathsf{rASMadd}\ \frac{}{\langle \mathsf{add}\ v,\ \sigma\rangle \rightrightarrows \sigma'}\ \exists v^* \in \mathsf{Val} : \exists \sigma^* \in \Sigma_{\mathrm{ASM}} : \sigma = (v^*,\sigma^*) \wedge \sigma' = (v^* + v, \sigma^*)$$

$$\mathsf{rASMseq}\ \frac{\langle c_1,\ \sigma\rangle \rightrightarrows \sigma'' \quad \langle c_2,\ \sigma''\rangle \rightrightarrows \sigma'}{\langle c_1\,;c_2,\ \sigma\rangle \rightrightarrows \sigma'}$$

Determinismus. Die Ausführung von ASM ist deterministisch, d.h. es gilt:

$$\forall c \in \mathsf{Com}_{\mathrm{ASM}} :\ P(c)$$

wobei $P(c) := \forall \sigma, \sigma', \sigma'' \in \Sigma_{\mathrm{ASM}} : (\langle c,\ \sigma\rangle \rightrightarrows \sigma' \text{ ist herleitbar } \wedge \langle c,\ \sigma\rangle \rightrightarrows \sigma'' \text{ ist herleitbar }) \Rightarrow \sigma' = \sigma''$.

■ Vervollständigen Sie den Beweis der folgenden beiden Aussagen auf dieser Seite und den nächsten beiden Seiten.

4.i) $\forall v \in \mathsf{Val} :\ P(\mathsf{add}\ v)$

4.ii) $\forall c_1, c_2 \in \mathsf{Com}_{\mathrm{ASM}} :\ P(c_1) \wedge P(c_2) \Rightarrow P(c_1\,;c_2)$

Antwort:

4.i) $\forall v \in \mathsf{Val} :\ P(\mathsf{add}\ v)$

Sei $v \in \mathsf{Val}$ beliebig aber fest. Es ist zu zeigen, dass $P(\mathsf{add}\ v)$ gilt. Seien die Stackzustände $\sigma, \sigma', \sigma'' \in \Sigma_{\mathrm{ASM}}$ beliebig aber fest. Angenommen es gilt

und es gilt

.

Da die Regel | | die letzte Regel in den Herleitungen von $\langle \mathsf{add}\ v,\ \sigma\rangle \rightrightarrows \sigma'$ und $\langle \mathsf{add}\ v,\ \sigma\rangle \rightrightarrows \sigma''$ sein muss, haben die Herleitungen die folgende Form:

Name: _____, Matrikelnr.: _____

- Form der Herleitung von $\langle \texttt{add}\ v,\ \sigma\rangle \rightrightarrows \sigma'$:

- Form der Herleitung von $\langle \texttt{add}\ v,\ \sigma\rangle \rightrightarrows \sigma''$:

Aus der Form der Herleitung von $\langle \texttt{add}\ v,\ \sigma\rangle \rightrightarrows \sigma'$ folgt, dass es einen Wert $v^* \in \mathrm{Val}$ und einen Stackzustand $\sigma^* \in \Sigma_{\mathrm{ASM}}$ gibt, so dass $\sigma = (v^*, \sigma^*) \wedge \sigma' = (v^* + v, \sigma^*)$.

Aus der Form der Herleitung von $\langle \texttt{add}\ v,\ \sigma\rangle \rightrightarrows \sigma''$ folgt, dass es einen Wert $v^{*\prime} \in \mathrm{Val}$ und einen Stackzustand $\sigma^{*\prime} \in \Sigma_{\mathrm{ASM}}$ gibt, so dass $\sigma = (v^{*\prime}, \sigma^{*\prime}) \wedge \sigma'' = (v^{*\prime} + v, \sigma^{*\prime})$.

Aus $\sigma = (v^*, \sigma^*)$ und $\sigma = (v^{*\prime}, \sigma^{*\prime})$ folgt, dass _____ gilt und dass _____ gilt. Mit $\sigma' = (v^* + v, \sigma^*)$ und $\sigma'' = (v^{*\prime} + v, \sigma^{*\prime})$ folgt, dass $\sigma' = \sigma''$ gilt. □

4.ii) $\forall c_1, c_2 \in \mathrm{Com}_{\mathrm{ASM}}:\ P(c_1) \wedge P(c_2) \Rightarrow P(c_1\,;c_2)$

Seien $c_1, c_2 \in \mathrm{Com}_{\mathrm{ASM}}$ beliebig aber fest. Angenommen es gelten $P(c_1)$ und $P(c_2)$. Es ist noch zu zeigen, dass $P(c_1\,;c_2)$ gilt. Seien die Stackzustände $\sigma, \sigma', \sigma'' \in \Sigma_{\mathrm{ASM}}$ beliebig aber fest. Angenommen es gilt

_____

und es gilt

_____.

Da die Regel _____ die letzte Regel in den Herleitungen von $\langle c_1\,;c_2,\ \sigma\rangle \rightrightarrows \sigma'$ und $\langle c_1\,;c_2,\ \sigma\rangle \rightrightarrows \sigma''$ sein muss, gibt es Stackzustände $\sigma^{*\prime}, \sigma^{*\prime\prime} \in \Sigma_{\mathrm{ASM}}$ so dass die Herleitungen die folgende Form haben:

- Form der Herleitung von $\langle c_1\,;c_2,\ \sigma\rangle \rightrightarrows \sigma'$:

Name: _____, Matrikelnr.: _____

- Form der Herleitung von $\langle c_1 ; c_2,\ \sigma\rangle \rightrightarrows \sigma''$:

Da $\langle c_1,\ \sigma\rangle \rightrightarrows \sigma^{*\prime}$ und $\langle c_1,\ \sigma\rangle \rightrightarrows \sigma^{*\prime\prime}$ herleitbar sind, folgt aus der ersten Induktionsannahme $P(c_1)$, dass

[ ] gilt.

Da $\langle c_2,\ \sigma^{*\prime}\rangle \rightrightarrows \sigma'$ und $\langle c_2,\ \sigma^{*\prime\prime}\rangle \rightrightarrows \sigma''$ herleitbar sind, folgt damit aus der zweiten Induktionsannahme $P(c_2)$, dass [ ] gilt. □

Name: _____, Matrikelnr.: _____

Aufgabe 5: Prozessalgebra CSP (11 Punkte)

- **Zur Information:** Alle zwei nachfolgenden Aufgabenteile sind unabhängig voneinander lösbar.
- **Zur Information:** Die Semantik der Prozessausdrücke der Sprache CSP finden Sie auf der Rückseite des Beiblatts „Beiblatt zur Klausur Modellierung, Spezifikation und Semantik".

(3 P) (A). Sei $E_A := \{a, b, c, d\}$ und sei die folgende Menge von Spuren Tr_A definiert:

$$Tr_A := \{\langle\rangle, \langle a\rangle, \langle a, b\rangle, \langle a, c\rangle, \langle a, b, \checkmark\rangle, \langle a, c, \checkmark\rangle, \langle a, b, d\rangle, \langle a, c, d\rangle\}$$

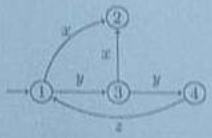
- Geben Sie einen Prozessausdruck P_A in CSP an, so dass $\text{traces}(P_A) = Tr_A$ gilt.

Antwort:

$P_A :=$

(6 P) (B). Sei folgendes Transitionssystem TS_B gegeben. Das rechts gegebene Diagramm visualisiert das Transitionssystem TS_B.

$$\begin{aligned} TS_B &:= (S^B, S_0^B, E^B, \rightarrow^B), \\ S^B &:= \{1, 2, 3, 4\} \\ S_0^B &:= \{1\} \\ E^B &:= \{x, y, z\} \\ \rightarrow^B &:= \{(1, x, 2), (1, y, 3), (3, x, 2), (3, y, 4), (4, z, 1)\} \end{aligned}$$



- Definieren Sie den Prozessbezeichner P_B durch ein Gleichungssystem aus Prozessausdrücken, so dass der Prozessausdruck P_B unter dem Gleichungssystem aus Prozessausdrücken den Prozess $(E^B, \text{E-Traces}(TS_B))$ spezifiziert.

Antwort:

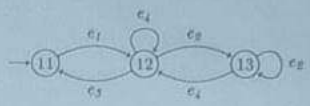
Name: _____, Matrikelnr.: _____

Aufgabe 6: Transitionssysteme und nebenläufige Ausführung (18 Punkte)

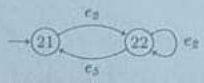
► **Zur Information:** Alle zwei nachfolgenden Aufgabenteile sind unabhängig voneinander lösbar.

(9 P) (A). Seien die beiden nachfolgenden Transitionssysteme TS_1 und TS_2 gegeben. Die Transitionssysteme werden jeweils durch das daneben stehende Diagramm veranschaulicht.

$$
\begin{aligned}
TS_1 &:= (S^{TS_1}, S_0^{TS_1}, E^{TS_1}, \rightarrow^{TS_1}) \\
S^{TS_1} &:= \{11, 12, 13\} \\
S_0^{TS_1} &:= \{11\} \\
E^{TS_1} &:= \{e_1, e_2, e_3, e_4\} \\
\rightarrow^{TS_1} &:= \{(11, e_1, 12), (12, e_2, 13), (12, e_4, 12), \\
&\qquad (12, e_3, 11), (13, e_4, 12), (13, e_2, 13)\}
\end{aligned}
$$



$$
\begin{aligned}
TS_2 &:= (S^{TS_2}, S_0^{TS_2}, E^{TS_2}, \rightarrow^{TS_2}) \\
S^{TS_2} &:= \{21, 22\} \\
S_0^{TS_2} &:= \{21\} \\
E^{TS_2} &:= \{e_2, e_4, e_5\} \\
\rightarrow^{TS_2} &:= \{(21, e_2, 22), (22, e_5, 21), (22, e_2, 22)\}
\end{aligned}
$$



■ Definieren Sie die Relation $\rightarrow^{MPC}$ als Menge der möglichen Transitionen der message-passing Komposition $MPC := (S^{MPC}, S_0^{MPC}, E^{MPC}, \rightarrow^{MPC})$ von TS_1 und TS_2. Die Definition der Menge soll als explizite Aufzählung ihrer Elemente erfolgen (extensionale Mengendefinition). Andere Formen der Definition sind hier nicht zulässig.

Antwort:

$MPC := (S^{MPC}, S_0^{MPC}, E^{MPC}, \rightarrow^{MPC})$

$S^{MPC} := S^{TS_1} \times S^{TS_2}$

$S_0^{MPC} := \{(11, 21)\}$

$E^{MPC} := E^{TS_1} \cup E^{TS_2}$

$\rightarrow^{MPC} :=$

Name: _____, Matrikelnr.: _____

(B). Seien die folgenden beiden Eigenschaften über Transitionssysteme gegeben: (9P)
Sei $TS := (S^{TS}, S_0^{TS}, E^{TS}, \rightarrow^{TS})$ ein beliebiges Transitionssystem.

$$P_1(TS) := \forall s \in S^{TS} : \exists s' \in S^{TS} : \exists e \in E^{TS} : \neg(s = s') \wedge (s, e, s) \in \rightarrow^{TS} \wedge (s, e, s') \in \rightarrow^{TS}$$

$$P_2(TS) := \forall s, s' \in S^{TS} : \forall e, e' \in \rightarrow^{TS} : ((s, e, s) \in \rightarrow^{TS} \wedge (s', e', s') \in \rightarrow^{TS}) \Rightarrow e = e'$$

Die Implikation $P_1(TS) \Rightarrow P_2(TS)$ gilt nicht für beliebige Transitionssysteme TS. Zeigen Sie dies durch Angabe eines Gegenbeispiels in Form eines Transitionssystems TS_1, für das die Implikation nicht gilt.

Antwort:

$TS_1 := (S^{TS_1}, S_0^{TS_1}, E^{TS_1}, \rightarrow^{TS_1})$

$S^{TS_1} :=$

$S_0^{TS_1} :=$

$E^{TS_1} :=$

$\rightarrow^{TS_1} :=$