Klausur Formale Systeme

Fakultät für Informatik

2. Klausur zum WS 2010/2011

Prof. Dr. Bernhard Beckert

08. April 2011

Name: ____________________

Vorname: ____________________

Matrikel-Nr.: ____________________

Platz: **Platz**

Klausur-ID: **Id**

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (17)	A2 (7)	A3 (5)	A4 (7)	A5 (10)	A6 (4)	A7 (10)	Σ (60)

Bewertungstabelle bitte frei lassen!

Zum Bestehen der Klausur sind 20 der erreichbaren 60 Punkte hinreichend.

Bonus: ______ **Gesamtpunkte:** ☐

1 Zur Einstimmung (4 + 3 + 6 + 4 = 17 Punkte)

a. Bitte kreuzen Sie in den folgenden Tabellen die für die Formeln in der jeweiligen Logik zutreffende Eigenschaft an. Für korrekte Antworten erhalten Sie einen Punkt, für falsche Antworten wird ein Punkt abgezogen. Dabei werden jedoch nie weniger als 0 Punkte pro Tabelle vergeben.

PL1 (Prädikatenlogik 1. Stufe)	keine Formel der PL1	erfüllbar (aber nicht allgemeing.)	allgemeingültig (und erfüllbar)	unerfüllbar
$\big(\neg\exists x\ p(x)\big) \;\to\; \big((\forall z\ (p(z) \to q(z))) \land (\forall y\ (p(y) \to \neg q(y)))\big)$			X	
$\forall r\ symmetrisch(r) \to \forall x \forall y\ (r(x,y) \leftrightarrow r(y,x))$	X			
$\forall x\ (\mathbf{1} \to p(x))$		X		
$\forall x\ (((\exists x\ p(x)) \land q(x)) \to \forall x(p(x) \land q(x)))$		X		

$p, q, r, symmetrisch$ sind Prädikatensymbole, die übrigen Bezeichner sind Variablen.

Modallogik	erfüllbar (aber nicht allgemeing.)	allgemeingültig (und erfüllbar)	unerfüllbar
$\Diamond(A \land B) \lor \Box\neg B \lor \Box\neg A$	X		
$\Diamond\Diamond A \to \Diamond A$	X		
$\neg\Box B \land \neg\Diamond\neg A \land \Box(A \to B)$			X

A, B sind aussagenlogische Variablen.

b. Bitte kreuzen Sie in der folgenden Tabelle das Zutreffende an. Für korrekte Antworten erhalten Sie einen Punkt, für falsche Antworten wird ein Punkt abgezogen. Dabei werden jedoch nie weniger als 0 Punkte für diese Teilaufgabe vergeben.

	Richtig	Falsch
Die Frage, ob für einen beliebigen Büchi-Automaten $\mathcal{B}$ mit Alphabet V $L^\omega(\mathcal{B}) = V^\omega$ gilt, ist entscheidbar.	X	
Die Menge der unerfüllbaren Formeln der Prädikatenlogik erster Stufe ist rekursiv aufzählbar.	X	
Für jede prädikatenlogischen Formel F erster Stufe gibt es eine erfüllbarkeitsäquivalente Pränexnormalform zu F.	X	
Hat eine beliebige prädikatenlogische Formel erster Stufe ein unendliches Modell, so auch ein endliches.		X
Seien A, B aussagenlogische Variablen. Zu der LTL-Formel $A\,\mathbf{U}\,(B\,\mathbf{U}\,(A\,\mathbf{U}\,(B\,\mathbf{U}\,A)))$ gibt es eine äquivalente LTL-Formel, die als Modaloperator nur $\mathbf{V}$ verwendet.	X	
Seien A, B beliebige Formelmengen für die $A \models B$ gilt. Dann gibt es eine Formel $f \in A$, so dass $A\backslash\{f\} \not\models B$		X

c. Kompaktheitssatz der Prädikatenlogik.

Füllen Sie die drei Lücken im folgenden Text.

Der Kompaktheitssatz für die Prädikatenlogik lautet:

> Eine unendliche Formelmenge Ψ ist genau dann erfüllbar,
>
> wenn jede ihrer *endlichen* Teilmengen erfüllbar ist.

Gegeben sei die prädikatenlogische Signatur $\Sigma = (\{\}, \{p\}, \alpha)$ mit $\alpha(p) = 2$.
Geben Sie zwei Eigenschaften prädikatenlogischer Modelle (D, I) an, für die man mit Hilfe des Kompaktheitssatzes zeigen kann, dass es keine prädikatenlogische Formel gibt, die diese Eigenschaft formalisiert:

i. D ist endlich

ii. Der durch $(D, I(p))$ beschriebene Graph ist zyklenfrei/zusammenhängend.

2 Normalformen (7 Punkte)

Transformieren Sie die folgende Formel schrittweise in **Skolemnormalform**.

$$\forall v \exists x \Big(\big(\forall z r(z,v) \rightarrow \forall x p(x,z) \big) \rightarrow (q(x) \lor r(v,x)) \Big)$$

Markieren Sie den Schritt, bei dem die Äquivalenz zur ursprünglichen Formel verloren geht.
Hinweis: Bei einer Formel in Skolemnormalform ist die Matrix in konjunktiver Normalform.

Variante 1:

$$\forall v \exists x \Big(\big(\forall z (r(z,v) \rightarrow \forall x\ p(x,z)) \big) \rightarrow (q(x) \lor r(v,x)) \Big)$$

i. **Bereinigen:**

$$\forall v \exists x \Big(\big(\forall z (r(z,v) \rightarrow \forall x_1\ p(x_1,z)) \big) \rightarrow (q(x) \lor r(v,x)) \Big)$$

ii. **Pränexnormalform:**

$$\begin{aligned}
& \forall v \exists x \Big(\big(\forall z (r(z,v) \rightarrow \forall x_1\ p(x_1,z)) \big) \rightarrow (q(x) \lor r(v,x)) \Big) \\
\equiv\ & \forall v \exists x \Big(\neg \big(\forall z (\neg r(z,v) \lor \forall x_1\ p(x_1,z)) \big) \lor (q(x) \lor r(v,x)) \Big) \\
\equiv\ & \forall v \exists x \Big(\big(\exists z (r(z,v) \land \neg \forall x_1\ p(x_1,z)) \big) \lor (q(x) \lor r(v,x)) \Big) \\
\equiv\ & \forall v \exists x \Big(\big(\exists z (r(z,v) \land \exists x_1\ \neg p(x_1,z)) \big) \lor (q(x) \lor r(v,x)) \Big) \\
\equiv\ & \forall v \exists x \exists z \exists x_1 \Big(\big(r(z,v) \land \neg p(x_1,z) \big) \lor (q(x) \lor r(v,x)) \Big)
\end{aligned}$$

iii. **Skolemisieren:**

$$\begin{aligned}
& \forall v \exists x \exists z \exists x_1 \Big(\big(r(z,v) \land \neg p(x_1,z) \big) \lor (q(x) \lor r(v,x)) \Big) \\
\overset{\circ}{\equiv}\ & \forall v \Big(\big(r(g(v),v) \land \neg p(h(v),g(v)) \big) \lor (q(f(v)) \lor r(v,f(v))) \Big)
\end{aligned}$$

iv. **Matrix in KNF:**

$$\begin{aligned}
& \forall v \Big(\big(r(g(v),v) \land \neg p(h(v),g(v)) \big) \lor (q(f(v)) \lor r(v,f(v))) \Big) \\
\equiv\ & \forall v \Big(\big(r(g(v),v) \lor q(f(v)) \lor r(v,f(v)) \big) \land \big(\neg p(h(v),g(v)) \lor q(f(v)) \lor r(v,f(v)) \big) \Big)
\end{aligned}$$

Variante 2:

$$\forall v \exists x \Big(\big(\forall z (r(z,v)) \rightarrow \forall x \ p(x,z)\big) \rightarrow (q(x) \vee r(v,x)) \Big)$$

i. **All-Abschluss:**

$$\forall z \forall v \exists x \Big(\big(\forall z (r(z,v)) \rightarrow \forall x \ p(x,z)\big) \rightarrow (q(x) \vee r(v,x)) \Big)$$

ii. **Bereinigen:**

$$\forall z_1 \forall v \exists x \Big(\big(\forall z (r(z,v)) \rightarrow \forall x_1 \ p(x_1,z_1)\big) \rightarrow (q(x) \vee r(v,x)) \Big)$$

iii. **Pränexnormalform:**

$$\begin{aligned} & \forall z_1 \forall v \exists x \Big(\big(\forall z (r(z,v)) \rightarrow \forall x_1 \ p(x_1,z_1)\big) \rightarrow (q(x) \vee r(v,x)) \Big) \\ \equiv \ & \forall z_1 \forall v \exists x \Big(\neg\big(\neg(\forall z \ r(z,v)) \vee \forall x_1 \ p(x_1,z_1)\big) \vee (q(x) \vee r(v,x)) \Big) \\ \equiv \ & \forall z_1 \forall v \exists x \Big(\big(\forall z (r(z,v)) \wedge \neg \forall x_1 \ p(x_1,z_1)\big) \vee (q(x) \vee r(v,x)) \Big) \\ \equiv \ & \forall z_1 \forall v \exists x \Big(\big(\forall z (r(z,v)) \wedge \exists x_1 \ \neg p(x_1,z_1)\big) \vee (q(x) \vee r(v,x)) \Big) \\ \equiv \ & \forall z_1 \forall v \exists x \forall z \exists x_1 \Big(\big(r(z,v) \wedge \neg p(x_1,z_1)\big) \vee (q(x) \vee r(v,x)) \Big) \end{aligned}$$

iv. **Skolemisieren:**

$$\begin{aligned} & \forall z_1 \forall v \exists x \forall z \exists x_1 \Big(\big(r(z,v) \wedge \neg p(x_1,z_1)\big) \vee (q(x) \vee r(v,x)) \Big) \\ \overset{\circ}{\equiv} \ & \forall z_1 \forall v \forall z \Big(\big(r(z,v) \wedge \neg p(g(z_1,v,z),z_1)\big) \vee (q(f(z_1,v)) \vee r(v,f(z_1,v))) \Big) \end{aligned}$$

v. **Matrix in KNF:**

$$\begin{aligned} & \forall z_1 \forall v \forall z \Big(\big(r(z,v) \wedge \neg p(g(z_1,v,z),z_1)\big) \vee (q(f(z_1,v)) \vee r(v,f(z_1,v))) \Big) \\ \equiv \ & \forall z_1 \forall v \forall z \Big(\big(r(z,v) \vee q(f(z_1,v)) \vee r(v,f(z_1,v))\big) \wedge \big(\neg p(g(z_1,v,z),z_1) \vee q(f(z_1,v)) \vee r(v,f(z_1,v))\big) \Big) \end{aligned}$$

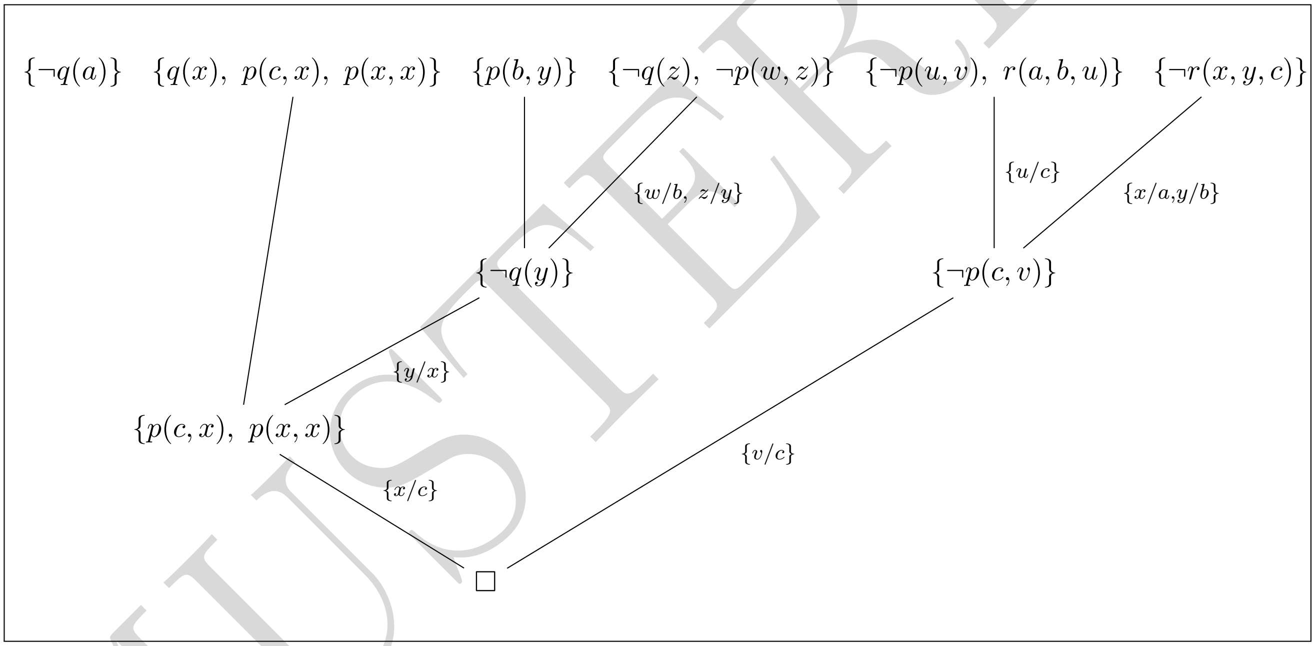
3 Resolutionskalkül (5 Punkte)

Seien

- p ein zweistelliges Prädikatensymbol,
- q ein einstelliges Prädikatensymbol,
- r ein dreistelliges Prädikatensymbol,
- a, b, c Konstanten,
- u, v, w, x, y, z Variablen.

Zeigen Sie mittels Resolution, dass die Menge folgender Klauseln unerfüllbar ist. **Notieren Sie** Ihren Beweis so, dass bei jeder neu entstehenden Klausel klar erkennbar ist, aus welchen Elternklauseln sie entsteht.

$\{\neg q(a)\}$ $\{q(x),\ p(c,x),\ p(x,x)\}$ $\{p(b,y)\}$ $\{\neg p(u,v),\ r(a,b,u)\}$ $\{\neg r(x,y,c)\}$ $\{\neg q(z),\ \neg p(w,z)\}$



4 Tableaukalkül (7 Punkte)

Zeigen Sie mit Hilfe des in der Vorlesung vorgestellten Tableaukalküls die

Allgemeingültigkeit

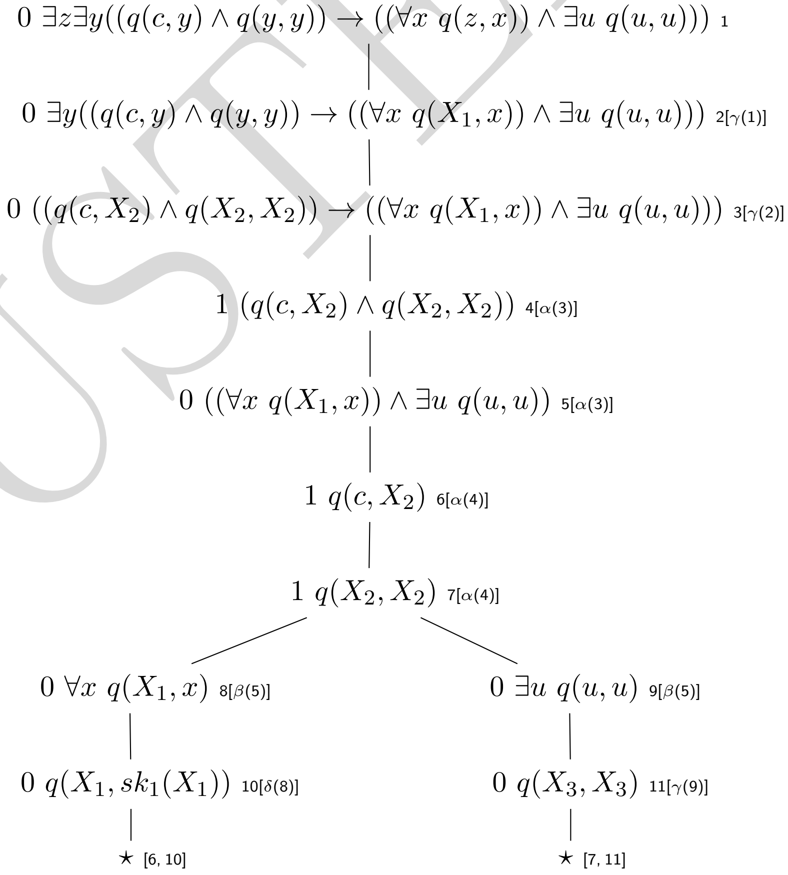
der folgenden Formel:

$$\exists z \exists y \big(\ (q(c,y) \land q(y,y)) \ \rightarrow \ ((\forall x\ q(z,x)) \land \exists u\ q(u,u)) \ \big)$$

Hierbei ist c eine Konstante.

Geben Sie dazu ein geschlossenes Tableau zu dieser Formel mit entsprechendem Vorzeichen an. Notieren Sie dabei:

- bei jeder Erweiterung, durch welche Regelanwendung eine Formel auf dem Tableau entstanden ist,
- bei Abschlüssen die beiden Partner,
- die schließende Substitution.



Die schließende Substitution lautet: $\{X_1/c, X_2/sk_1(c), X_3/sk_1(c)\}$.

5 Formalisieren Prädikatenlogik (2+4+4 = 10 Punkte)

Gegeben sei die prädikatenlogische Signatur $\Sigma = (\{\}, \{\text{tm}\}, \alpha)$ mit $\alpha(\text{tm}) = 2$. Das Modell (V, I) zu dieser Signatur sei gegeben durch:

- $V = \wp(\mathbb{N})$ (Potenzmenge von $\mathbb{N}$) und
- $I(tm) = \{(s,t) \mid s \subseteq t, t \subseteq \mathbb{N}\}$ (Dabei steht *tm* für **Teil**m**enge).

Geben Sie je eine Formel der Prädikatenlogik ϕ, χ bzw. ψ 1. Stufe (PL1) an, so dass

a. $\text{val}_{V,I,\beta}(\phi(x)) = W$ gdw. $\beta(x) = \varnothing$

$\phi(x) \equiv \forall y \; tm(x,y)$

b. $\text{val}_{V,I,\beta}(\chi(x)) = W$ gdw. $\beta(x)$ eine einelementige Menge ist.
Hinweis: verwenden Sie die in Teilaufgabe a. definierte Formel ϕ.

$\chi(x) \equiv \neg\exists v(tm(v,x) \land \neg(v \doteq x) \land \neg\phi(v))$

c. $\text{val}_{V,I,\beta}(\psi(x,y,z)) = W$ gdw. $\beta(x) \setminus \beta(y) = \beta(z)$.
Hinweis: verwenden Sie die in Teilaufgabe b. definierte Formel χ.

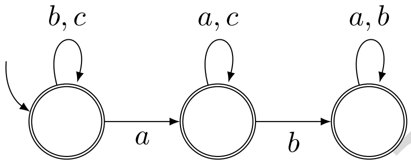
$\psi(x) \equiv \forall v(\chi(v) \to (tm(v,z) \leftrightarrow (tm(v,x) \land \neg tm(v,y))))$

6 Büchi-Automaten (4 Punkte)

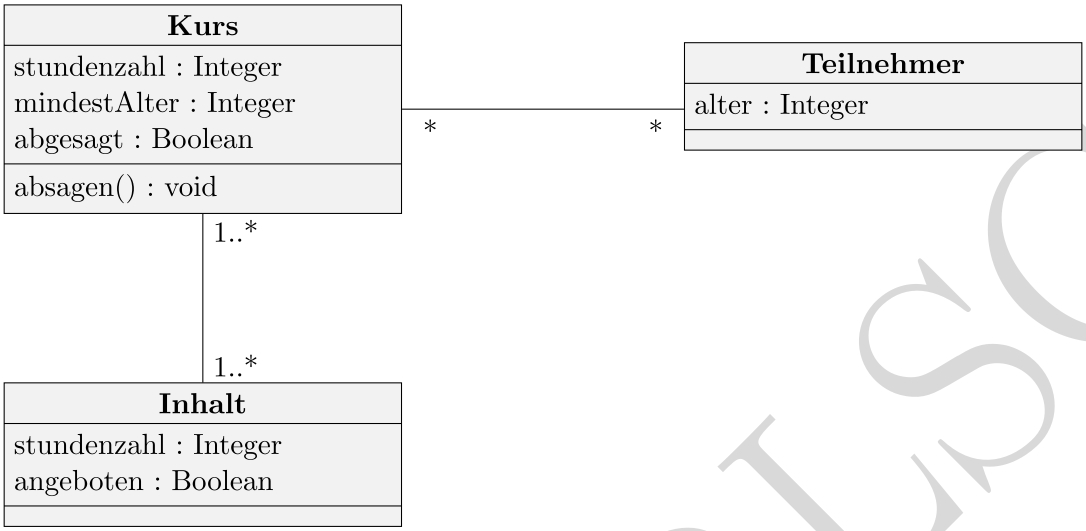
Gegeben sei ein endliches Alphabet $V = \{a, b, c\}$.

Geben Sie einen Büchi-Automaten $\mathcal{B}$ an, so dass die von $\mathcal{B}$ akzeptierte Sprache $L^\omega(\mathcal{B})$ genau diejenigen Wörter $w \in V^\omega$ enthält, für die gilt:

> Es gibt keine Vorkommen von a und c (a vor c), so dass dazwischen ein b liegt.



UML-Diagramm zu Aufgabe 7



Übersicht über wichtige OCL-Operationen

Folgende Operationen sind auf alle Gesamtheiten (Mengen, Multimengen und Listen) anwendbar.

Operation	**Ergebnis** (bei Anwendung auf M)
`size()`	die Anzahl der Elemente in M.
`including(o)`	die Gesamtheit, die M erweitert um o entspricht.
`collect(v|exp)`	die Gesamtheit, die entsteht, wenn der Audruck *exp* für jedes Element in M ausgewertet wird.
`intersection(N)`	der Durchschnitt von M und N.
`union(N)`	die Vereinigung von M und N.
`includes(o)`	wahr genau dann, wenn o ein Element in M ist.
`includesAll(N)`	wahr genau dann, wenn jedes Element n der Gesamtheit N auch ein Element in M ist.
`isEmpty()`	wahr genau dann, wenn M kein Element enthält.
`exists(v|b)`	wahr genau dann, wenn es ein Element v in M gibt, so dass dafür der boolesche Ausdruck b zu wahr auswertet.
`forAll(v|b)`	wahr genau dann, wenn für jedes Element v in M der boolesche Ausdruck b zu wahr auswertet.
`select(v|b)`	die Gesamtheit der Elemente v von M, die b erfüllen.
`iterate(v ; init | exp)`	Für jedes Element v in M wird *exp* ausgewertet unter Verwendung des in *init* initialisierten Akkumulators.

7 Object Constraint Language (2+3+5 = 10 Punkte)

Auf der linken Seite ist ein UML-Klassendiagramm abgebildet. Es modelliert den Zusammenhang zwischen Personen, deren besuchten Kursen, sowie den Inhalten der Kurse: Ein Teilnehmer kann beliebig viele Kurse besuchen und ein Kurs besteht aus beliebig vielen Teilnehmern. Ein Kurs setzt sich aus mindestens einem Inhalt zusammen.

Vervollständigen Sie die folgenden OCL-Constraints gemäß der angegebenen natürlichsprachlichen Bedeutung.

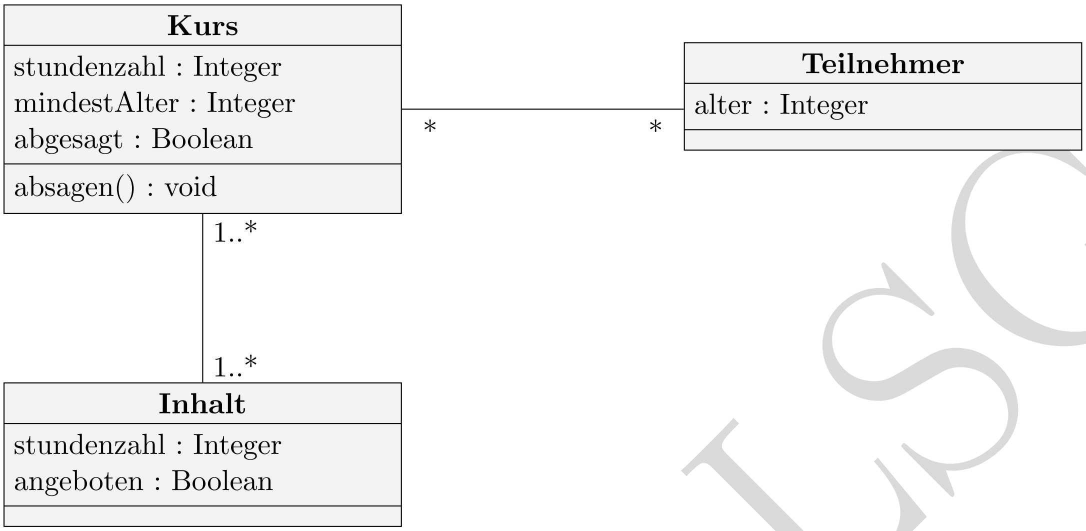
a. Das Alter eines Teilnehmers ist immer größer oder gleich dem `mindestAlter` jedes Kurses, den er besucht.

```
context Teilnehmer
inv: kurs->forall(self.alter >= mindestAlter)
```

b. Die Stundenzahl eines Kurses entspricht der Summe der Stundenzahlen der Inhalte, die zu dem Kurs gehören. Verwenden Sie hierbei die Funktion `iterate`, um diese Summe zu bestimmen.

```
context Kurs
inv: stundenzahl = inhalt->iterate(i: Inhalt; res: Integer = 0 | res + i.stundenzahl)
```

UML-Diagramm zu Aufgabe 7



c. Die Methode `absagen() : void` der Klasse `Kurs` ist wie folgt spezifiziert:

Wenn zu Beginn der Methode gilt:

- Kein Teilnehmer besucht diesen Kurs.

Dann gilt nach Ausführung der Methode:

- Der Kurs ist abgesagt. (Das entsprechend benannte Feld `abgesagt` ist auf `true` gesetzt.)
- Jeder bisher zum Kurs gehörende Inhalt ist als nicht mehr angeboten gekennzeichnet, sofern nun alle Kurse abgesagt sind, die diesen Inhalt abdecken. (Das Feld `angeboten` ist dann auf `false` gesetzt.)

```
context Kurs::absagen() : void
pre:
```

```
teilnehmer->isEmpty()
```

```
post:
```

```
Kurs.abgesagt and
inhalt->forall(i | i.kurs->forall(k | k.abgesagt)) implies not(i.angeboten))
```