Übungen Logik und formale Methoden – Prädikatenlogik

[Einige der Übungen sind von Bodo Igler.]

A Objekte und Prädikate

1. WM 2014

Gegeben seien folgende Prädikate:

$B(x, y)$: x besiegt y
$F(x)$: x ist Fußball-Nationalmannnschaft
$T(x, y)$: x ist Torwart von y

und folgende Konstanten:

de: deutsche Nationalmannschaft
br: brasilianische Nationalmannschaft
us: Nationalmannschaft der USA

Drücken Sie folgende Aussagen in der Prädikatenlogik aus:

(a) Jede Fußball-Nationalmannschaft hat einen Torwart.

(b) Wenn de gegen us gewinnt, dann verliert de nicht jedes Spiel.

(c) br schlägt jedes Team, gegen das de verliert, mit Ausnahme von sich selbst.

Zusatzaufgabe: Spezifizieren Sie die Situation mit fünf Mannschaften in Alloy und erzeugen Sie mit dem Alloy-Analyzer mögliche Verläufe der WM.

2. Prädikate finden

Finden Sie geeignete Prädikate, um folgendes in der Prädikatenlogik auszudrücken:

(a) Alle roten Dinge sind in der Schachtel.

(b) Nur rote Dinge sind in der Schachtel.

(c) Kein Tier ist zugleich eine Katze und ein Hund.

(d) Jeder Preis wurde von (irgend)einem Spieler gewonnen.

(e) Ein Spieler hat jeden Preis gewonnen.

(f) Keines der roten Dinge ist blau.

3. Formeln interpretieren

Was bedeuten folgende Formeln der Prädikatenlogik im Universum der natürlichen Zahlen $\mathbb{N}$? (Die Beispiele sind aus einem grundlegenden Werk der theoretischen Informatik.)

(a) $\forall q \exists p \forall x \forall y (p > q \land (x > 1 \land y > 1 \to x \times y \neq p))$

(b) $\forall a \forall b \forall c \forall n ((a > 0 \land b > 0 \land c > 0 \land n > 2) \to a^n + b^n \neq c^n)$

(c) $\forall q \exists p \forall x \forall y (p > q \land (x > 1 \land y > 1 \to (x \times y \neq p \land x \times y \neq p + 2)))$

B Die formale Sprache der Prädikatenlogik

4. Terme und Syntaxbaum

Die Menge F sei d, f, g mit einer Konstanten d, einem Funktionssymbol f mit 2 Argumenten und einem Funktionssymbol g mit 3 Argumenten.

Welche der folgenden Zeichenfolgen sind Terme über F. Zeichnen Sie den Syntaxbaum der Zeichenfolgen, die Terme sind.

(a) $g(d, d)$

(b) $f(x, g(y, z), d)$

(c) $g(x, f(y, z), d)$

(d) $g(x, h(y, z), d)$

(e) $f(f(g(d, x), f(g(d, x), y, g(y, d)), g(d, d)), g(f(d, d, x), d)z)$

5. Syntaktisch korrekte Formeln

m sei eine Konstante, f ein Funktionssymbol mit einem Argument und S und B zwei Prädikate mit jeweils zwei Argumenten. Welche der folgenden Zeichenfolgen ist eine (syntaktisch korrekte) Formel der Prädikatenlogik? Begründung, wenn nicht.

(a) $S(m, x)$

(b) $B(m, f(m))$

(c) $f(m)$

(d) $B(B(m, x), y)$

(e) $S(B(m), z)$

(f) $(B(x, y) \rightarrow (\exists z S(x, y)))$

(g) $(S(x, y) \rightarrow S(y, f(f(x))))$

(h) $(B(x) \rightarrow B(B(x)))$

6. Spielen mit Formeln

Gegeben sei die Formel ϕ

$$\exists x(P(y, z) \land (\forall y(\neg Q(y, x) \lor P(y, z))),$$

mit Prädikaten P und Q in zwei Argumenten.

(a) Zeichnen Sie den Syntaxbaum von ϕ.

(b) Identifizieren Sie alle freien und gebundenen Variablen in ϕ.

(c) Gibt es eine Variable in ϕ, die gebunden *und* frei vorkommt?

(d) Gegeben seien eine Variable w und die Funktionssymbole $f(x)$ und $g(y, z)$ mit einem bzw. zwei Argumenten.
Berechnen Sie $\phi[w/x]$, $\phi[w/y]$, $\phi[f(x)/y]$ und $\phi[g(y, z)/z]$.

(e) Schreiben Sie die Formel in der Syntax von Logic Workbench *lwb*.

C Semantik der Prädikatenlogik

7. Modelle erfüllen die Formel?

Gegeben sei die Formel ϕ als $\forall x \exists y \exists z (P(x,y) \land P(z,y) \land P(x,z) \to P(z,x))$ Welches der folgenden Modell erfüllt ϕ?

(a) Das Modell mit dem Universum $\mathcal{A}$ gewählt als die natürlichen Zahlen und der Definition von P als $\{(m,n) | m < n\}$.

(b) Das Modell mit den natürlichen Zahlen $\mathbb{N}$ als Universum und P als $\{(m, 2*m) | m \in \mathbb{N}\}$.

(c) Das Modell mit den natürlichen Zahlen $\mathbb{N}$ als Universum und P als $\{(m,n) | m < n+1\}$.

8. Shakespeare's Welt

Welt A

Das Universum ist $\{Romeo, Julia, Benedikt, Beatrice\}$
Als Konstanten seien definiert: $m := Romeo, n := Julia$
Als Prädikate seien definiert:

$$
\begin{aligned}
F &:= \{Romeo, Benedikt\} \\
G &:= \{Julia, Beatrice\} \\
L &:= \{(Romeo, Julia), (Julia, Romeo), \\
&\quad (Benedikt, Beatrice), (Beatrice, Benedikt), (Benedikt, Benedikt)\}
\end{aligned}
$$

Welt B

Das Universum ist $\{4, 7, 8, 11, 12\}$
Als Konstanten seien definiert: $m := 7, n := 12$
Als Prädikate seien definiert:

F :=Die geraden Zahlen des Universums
G :=Die ungeraden Zahlen des Universums
L :=Die Menge der Paare (x, y) mit $x < y$

Welche der folgenden Aussagen treffen in der jeweiligen Welt zu, welche nicht? Überprüfen Sie ihre Lösung in lwb.

- $\exists x L(m,x)$
- $\forall x L(m,x)$
- $\exists x L(m,x) \to L(m,n)$
- $\forall x (F(x) \leftrightarrow \neg G(x))$
- $\forall x (G(x) \to (L(x,m) \lor \neg L(m,x))$
- $\forall x (G(x) \to \exists y L(x,y)$
- $\exists x (F(x) \land \forall y (G(y) \to L(x,y))$

9. Modell finden

Gegeben sei die Formel ϕ als $\forall x \forall y Q(g(x,y), g(y,y), z)$, wobei Q ein Prädikatensymbol mit 3 Argumenten und g ein Funktionssymbol mit 2 Argumenten ist.

(a) Finden Sie ein Modell $\mathcal{M}$ und eine Variablenbelegung l, so dass gilt: $\mathcal{M} \vDash_l \phi$.

(b) Finden Sie ein Modell $\mathcal{M}$ und eine Variablenbelegung l, so dass gilt: $\mathcal{M} \nvDash_l \phi$.

(c) Können Sie die erste Teilaufgabe auch mit der Logic Workbench lösen?

10. Modelle erfüllen die Formel?

Gegeben sei die Formel ϕ als $\forall x \forall y \exists z (R(x,y) \to R(y,z))$ mit einem Prädikat R in zwei Argumenten.

(a) Das Universum $\mathcal{A}$ sei die Menge $\{a, b, c, d\}$ und gegeben sei die Relation R als $\{(b,c), (b,b), (b,a)\}$. Gilt $\mathcal{M} \vDash \phi$? Begründung.

(b) Das Universum $\mathcal{A}$ sei die Menge $\{a, b, c\}$ und gegeben sei die Relation R als $\{(b,c), (a,b), (c,b)\}$. Gilt $\mathcal{M} \vDash \phi$? Begründung.

11. Relationen und Logik

Gegeben sei eine Menge $\mathcal{A}$ und eine binäre Relation R über $\mathcal{A}$.

Drücken Sie die folgenden Eigenschaften der Relation als Formel der Prädikatenlogik aus und konstruieren Sie jeweils ein kleines Modell mit einer (kleinen) endlichen Menge $\mathcal{A}$.

(a) R ist nicht leer.

(b) R ist transitiv.

(c) R ist reflexiv.

(d) R ist irreflexiv.

(e) R ist symmetrisch.

(f) R ist eine Funktion.

(g) R ist injektiv.

(h) R ist surjektiv.

(i) R ist total.

(j) R ist bijektiv.

12. Paradoxon?

Der Kreter Epimenides sagt: „Alle Kreter lügen“. Dieser Satz wird gerne als das Lügner-Paradoxon bezeichnet.

Beispiel aus dem Internet
(http://www.aleatorik.eu/2010/02/14/alle-kreter-lugen/):

„Wenn alle Kreter lügen und dieser Mann aus Kreta ist, sagt er nicht die Wahrheit. Die Aussage – alle Kreter lügen – ist also gelogen. Daraus folgt, dass die Kreter die Wahrheit sagen. Dann sagt auch unser Kreter die Wahrheit, wenn er behauptet, dass alle Kreter lügen. Wenn aber die Aussage – alle Kreter lügen – die Wahrheit ist, dann ist sie gelogen. Und daraus folgt, dass der Kreter die Wahrheit sagt, wenn er sagt, dass sie lügen. Und das bedeutet etc.pp.“

Überdenken Sie und diskutieren Sie den Sachverhalt.

D Natürliches Schliessen

13. Gesetze der Prädikatenlogik

Zeigen Sie folgende Gesetze der Prädikatenlogik mittels natürlichen Schließens:

(a) $\neg\forall x P(x) \vdash \exists x \neg P(x)$

(b) $\exists x \neg P(x) \vdash \neg\forall x P(x)$

(c) $\neg\exists x P(x) \vdash \forall x \neg P(x)$

(d) $\forall x \neg P(x) \vdash \neg\exists x P(x)$

(e) $\forall x P(x) \wedge \forall x Q(x) \vdash \forall x (P(x) \wedge Q(x))$

(f) $\forall x (P(x) \wedge Q(x)) \vdash \forall x P(x) \wedge \forall x Q(x)$

(g) $\exists x P(x) \vee \exists x Q(x) \vdash \exists x (P(x) \vee Q(x))$

(h) $\exists x (P(x) \vee Q(x)) \vdash \exists x P(x) \vee \exists x Q(x)$

(i) $\forall x P(x) \vee \forall x Q(x) \vdash \forall x (P(x) \vee Q(x))$

(j) $\exists x (P(x) \wedge Q(x)) \vdash \exists x P(x) \wedge \exists x Q(x)$

14. Einfache Herleitungen

Beweisen Sie folgende Aussagen mittels natürlichen Schließens

(a) $\forall x \forall y P(x, y) \vdash \forall u \forall v P(u, v)$

(b) $\exists x \exists y F(x, y) \vdash \exists u \exists v F(u, v)$

(c) $\exists x \forall y P(x, y) \vdash \forall y \exists x P(x, y)$

15. Herleitungen

Beweisen Sie folgende Aussagen mittels natürlichen Schließens

(a) $\exists x (S \rightarrow Q(x)) \vdash S \rightarrow \exists x Q(x)$

(b) $\neg\forall x \neg P(x) \vdash \exists x P(x)$

(c) $\exists x P(x) \vdash \neg\forall x \neg P(x)$

(d) $S \rightarrow \forall x Q(x) \vdash \forall x (S \rightarrow Q(x))$

16. Herleitung finden und beweisen

Formalisieren Sie folgende Aussagen und beweisen Sie mittels des natürlichen Schließens:

Alle Linguisten kennen Chomsky
Alle Computerlinguisten sind Linguisten
Also: Alle Computerlinguisten kennen Chomsky

E Relationale Logik mit Alloy

17. Modelle für Assoziationen zwischen Klassen

Gegeben seien zwei Klassen `A` und `B` mit einer Assoziation zwischen den beiden Klassen (wie im Klassendiagramm der UML üblich). Es sollen folgende Multiplizitäten der Assoziation vorgegeben sein:

(a) `1 - 1`

(b) `0..1 - 1`

(c) `0..1 - 0..1`
(d) `1 - *`
(e) `0..1 - *`
(f) `0..1 - 1..*`
(g) `1 - 1..*`
(h) `* - *`
(i) `1..* - *`
(j) `1..* - 1..*`

Erstellen Sie Alloy-Spezifikationen und verwenden Sie den Alloy Analyzer, um Objektmodelle zu simulieren, die die Constraints erfüllen.

18. Assoziationsklassen simulieren

Im Metamodell der UML ist eine Assoziationsklasse sowohl eine Assoziation als auch eine Klasse. Erstelllen Sie Alloy-Spezifikationen für die Klassen `A` und `B` mit verbindender Assoziationsklasse `AC` mit folgenden Multiplizitäten.

(a) `* - *`
(b) `1..* - 1..*`

19. Graphen

Erstellen Sie Alloy-Modelle für folgende Typen von Graphen und „überprüfen" Sie das jeweilige Modell durch Simulation im Alloy Analyzer.

(a) (Ungerichteter) *Graph*, bestehend aus Ecken und Kanten. Zwei Ecken sind durch höchstens eine Kante verbunden. Es gibt keine Schleifen, d.h. eine Ecke ist nie durch eine Kante mit sich selbst verbunden.
(b) *Gerichteter Graph* ohne Schleifen
(c) *Gerichteter Graph* mit Schleifen
(d) *Gerichteter Multigraph*, bei dem zwei Ecken durch mehrere Kanten verbunden sein können.
(e) Ungerichteter *Multigraph*

20. Entwurfsmuster Composite

Das Entwurfsmuster Composite der GoF hat folgende Bestimmung

> Compose objects into tree structures to represent part-whole hierarchies. Composite lets clients treat individual objects and compositions of objects uniformly.

Erstellen Sie eine Spezifikation des Musters in Alloy und drücken Sie *alle* benötigten Eigenschaften der Struktur des Musters in Alloys relationaler Logik aus.

Zusatzaufgabe: Betrachten Sie auch dynamische Aspekte: Veränderungen der Struktur durch Hinzufügen oder Wegnehmen von Elementen. Was lernt man daraus über das Muster? Was muss man bei einer Implementierung des Musters infolgedessen beachten?

21. Entwurfsmuster Decorator

Das Entwurfsmuster Decorator der GoF hat folgende Bestimmung

> Attach additional responsibilities to an object dynamically. Decorators provide a flexible alternative to subclassing for extending functionality.

Erstellen Sie eine Spezifikation des Musters in Alloy und drücken Sie *alle* benötigten Eigenschaften der Struktur des Musters in Alloys relationaler Logik aus.

Zusatzaufgabe: Betrachten Sie auch dynamische Aspekte: Veränderungen der Struktur durch Hinzufügen oder Wegnehmen von Elementen. Was lernt man daraus über das Muster? Was muss man bei einer Implementierung des Musters infolgedessen beachten?

22. E-Mail-Anwendung

Gegeben sei das Diagramm 1 für die Struktur der Empfänger in einer E-Mail-Anwendung.

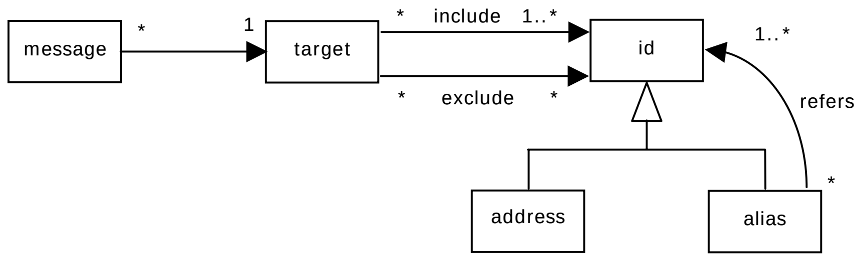

Abbildung 1: Struktur E-Mail-Empfänger

Das Diagramm zeigt, dass jeder potentielle Empfänger einer E-Mail eine direkte Adresse hat oder einen Alias, der selbst wieder auf andere Adressen oder Aliase verweisen kann – verallgemeinert zu einer `id`.

Jede Nachricht hat nun eine Menge von Zielen (`target`), die bestimmte Empfänger explizit einschließen soll, andere explizit ausschließen soll.

Ein solches Schema kann zum Beispiel praktisch sein, wenn man manchmal Nachrichten an bestimmte Gruppen von Personen wie Kollegen verschickt, dabei aber andere Personen, wie z.B. persönliche Bekannte ausschließen möchte – oder umgekehrt.

Die Frage nun: wie soll in der E-Mail-Anwendung mit den Attributen `include` und `exclude` verfahren werden? Sollen erst die Aliase dereferenziert werden und dann die Differenz zwischen den *include*-Adressen und den *exclude*-Adressen gebildet werden? Oder muss die Reihenfolge umgekehrt sein? Oder spielt sie gar keine Rolle?

[Beispiel stammt von Michael Jackson]

23. Kripke-Strukturen

Eine Kripke-Struktur ist ein endlicher Zustandsautomat (*state machine*) mit einer nichtleeren Menge `init` von Initalzuständen, einer Abbildung `prop` der Zustände in eine Menge von atomaren Aussagen, einer Relation von Zustandübergängen `next` und einer Menge von Endzuständen `final`. Die Abbildung `prop` gibt an, welche der atomaren Aussagen in dem jeweiligen Zustand gegeben (wahr) ist.

(a) Erstellen Sie eine Alloy-Spezifikation für Kripke-Strukturen.

(b) Schreiben Sie eine `fun`-Anweisung `Reaches`, die zu einer Kripke-Struktur alle Zustände ermittelt, die von einem Initalzustand aus erreichbar sind.

(c) Schreiben Sie ein Prädikat, mit dem man überprüfen kann, ob eine Kripke-Struktur keinen Deadlock hat (d.h. nur finale Zustände können keinen Folgezustand haben).

(d) Schreiben Sie ein Prädikat, mit dem man überprüfen kann, ob eine Kripke-Struktur deterministisch ist (d.h. ein Zustand hat höchstens eine weiterführende Transition).

(e) Schreiben Sie ein Prädikat, mit dem man überprüfen kann, ob ein Zustand mit einer bestimmten Eigenschaft in `prop` erreicht werden kann.

(f) Schreiben Sie ein Prädikat, mit dem man überprüfen kann, ob man einen Zustand mit einer bestimmten Eigenschaft ausgehend von einem beliebigen Zustand erreichen kann.

24. Gruppen

Eine Gruppe G ist ein Tupel $(G, *, 1)$ bestehend aus einer Menge von Elementen G, einer Funktion $* : G \times G \to G$ und einem ausgezeichneten Element $1 \in G$, so dass gilt:

(1) Inverse Elemente: zu jedem $x \in G$ gibt es ein $y \in G$ so dass gilt: $x * y = y * x = 1$

(2) Assoziativgesetz: für alle $x, y, z \in G$ gilt: $x * (y * z) = (x * y) * z$

(3) Neutrales Element: für alle $x \in G$ gilt: $1 * x = x * 1 = x$.

Aufgaben:

(a) Spezifizieren Sie die Struktur einer Gruppe in Alloy.

(b) Lassen Sie den Alloy Analyzer eine Gruppe mit drei Elementen generieren.

(c) Überprüfen Sie die Aussage, dass jedes Element einer Gruppe ein *eindeutiges* inverses Element hat mittels des Alloy Analyzers. Diskutieren Sie den Befund.

(d) Schreiben Sie eine `assert`-Anweisung zur Überprüfung, ob eine Gruppe *kommutativ* ist, d.h. ob für alle $x, y \in G$ gilt: $x * y = y * x$. Überprüfen Sie die Anweisung mit steigendem *Scope* beginnend bei 1.

25. Gruppenaxiome

In einem einführenden Aufsatz über das Suchen endlicher Modelle erwähnt Jian Zhang folgendes Beispiel:

> It had been thought that the follwing equation alone can axiomatize the group theory
> $i(f(f(x, f((f(y,z), i(f(y, f(i(u), i(z)))))), x)) = u$
> Here i means inverse, f means multiplication, and all the variables are universally quantified.
> [Jian Zhang: Finite Model Searching: An Introduction and Some Personal Notes, Beijing 2002, Aug.]

Der Autor nennt ein Beispiel einer Struktur mit 2 Elementen, die die Gleichung erfüllt, aber keine Gruppe ist. McCune hat diese mit einem „Modelfinder“ gefunden.

Wie geht das mit Alloy?

26. Projektive Ebenen

Eine projektive Ebene ist eine Menge von Punkten und Geraden mit folgenden Eigenschaften:

(1) Zu je zwei verschiedenen Punkten gibt es genau eine Gerade, auf der die Punkte liegen.

(2) Zu je zwei verschiedenen Geraden gibt es genau einen Punkt, in dem sich die Geraden schneiden.

(3) Auf jeder Geraden liegen wenigstens 3 Punkte.

(4) Durch jeden Punkt gehen mindestens 3 Geraden.

Aufgaben:

- Spezifizieren Sie die Struktur einer (endlichen) projektiven Ebene in Alloy.
- Zeigen Sie mit Alloy, dass die kleinste endliche projektive Ebene 7 Punkte und 7 Geraden hat.

27. Sudoku

In einem gelösten Sudoku kommen die Ziffern 1 bis 9 genau einmal in jeder Zeile, einmal in jeder Spalte und einmal in jedem Sektor (einem der 9 3x3-Subquadrate vor).

Gegeben sei folgendes Rätsel (Abb. 2):

				9	7		1	
9	8			4		6		7
5	6							4
1			8					
		6		3	1			
4		2				3	8	
7		5		2			6	
	9							5
					6	4		

Abbildung 2: Sudoku

(a) Schreiben Sie eine Alloy-Spezifikation für die Regeln von Sudoku. Hinweis: man kann Sudoku auffassen als ternäre Relation, bei der jedes Tupel (r, c, v) folgendes darstellt: r ist die Zeile (*row*), c ist die Spalte (*column*) und v ist der Wert (*value*) der entsprechenden Zelle.

(b) Geben Sie ein partielles Modell für die Relation vor – entsprechend der vorbelegten Ziffern im Rätsel und lassen Sie den Alloy Analyzer die Lösung finden.

(c) Wie „übersetzt“ Alloy wohl die Aufgabe in eine Formel der Aussagenlogik? (Denken Sie an die Aufgabe mit der Kartenfärbung)

28. Erdkunde-Klassenarbeit

Logelei aus dem ZEIT-Magazin vom 24.05.2007

> Claudia musste für den Erdkundeunterricht vier Berge auswendig lernen. Am nächsten Tag in der Klassenarbeit kann sie sich noch an Folgendes erinnern:

Auf dem Felderer findet sich kein Steinhaufen.
Der Berg in Barunien heißt weder Schneehorn, noch ist er 2317 Meter hoch.
Auf einem der Berge war ein gigantischer Felsen.
Weder der Berg in Gorabien noch der Felderer ist 2581 Meter hoch.
Der Berg in Seborien ist 2128 Meter hoch.
Das Schneehorn ist nicht 2222 Meter hoch.
Der Berg mit der Hütte darauf ist weder 2222 Meter hoch noch der Borken, noch in Seborien.
Der Berg in Gorabien ist nicht 2317 Meter hoch.
Weder auf dem 2222 Meter hohen Berg noch auf dem Borken gibt es einen See. Es gab einen Weldberg.

Claudia grübelt und grübelt, aber mehr will ihr partout nicht einfallen. Da fällt ihr auf, dass die Tafel nicht geputzt ist. Dort steht noch, wohl von einem Erdkundeunterricht der Parallelklasse: „In Lusanien steht der Borken“. Claudia ist erleichtert.

Wie heißen die vier Berge, wie hoch sind sie, in welchem Land sind sie, und welche Besonderheit findet man auf den vier Gipfeln?

Man kann die Frage noch erweitern: Gibt es nur eine Lösung? Gibt es mehrere Lösungen, wenn die Tafel geputzt gewesen wäre?

29. Klassendiagramm als Alloy Modell

Modellieren Sie das in Abb. 3 dargestellte Klassendiagramm als Alloy-Modell.

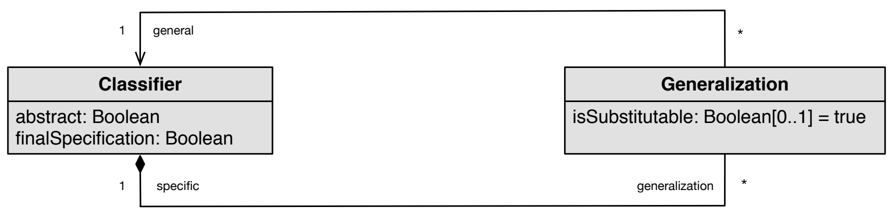

Abbildung 3: Ausschnitt Metamodell der UML

30. ER-Modell Person-Firma in Alloy

Grundlage für diese Aufgabe ist das ER-Modell (in UML-Notation) Person-Firma in Abb. 4.

(a) Wandeln Sie das ER-Modell in ein Alloy-Modell um.

(b) Schränken Sie das Alloy-Modell ein:

> „Der Chef einer Firma arbeitet auch in dieser Firma.“

(c) Schränken Sie das Alloy-Modell ein:

> „Der Chef einer Firma hat keinen seiner Angestellten in einer anderen Firma als Chef.“

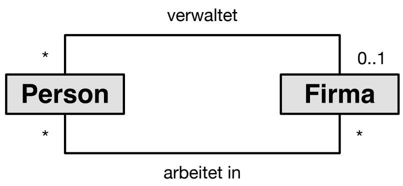

Abbildung 4: Er-Diagramm Firma

31. Alloy-Modell analysieren und anpassen

Betrachten Sie folgendes Alloy-Modell

```
abstract sig Person {
vater: lone Mann,
mutter: lone Frau
}{}

sig Mann extends Person {}{}
sig Frau extends Person {}{}

fact azyklisch {
  no (^(vater + mutter) & iden)
}

pred show {}
run show for 6
```

(a) Zeichnen Sie ein passendes ER-Diagramm (in UML-Notation)

(b) Überprüfen Sie mit Hilfe geeigneter Assertions, ob das Modell gemäß Listing alle unerwünschten Spezialfälle (z.B. eine Person ist ihr eigener Onkel/ihre eigene Tante, eine Person ist ihr eigener Vater/ihre eigene Mutter) aussschließt?

(c) Passen Sie das Alloy-Modell so an, dass alle in (b) als unerwünscht identifizierten Spezialfälle ausgeschlossen sind.

32. Behauptungen mit Alloy prüfen

Betrachten das Alloy-Modell aus der vorherigen Aufgabe. Prüfen Sie die im Folgenden aufgestellten Behauptungen:

(a) Die Formulierung von `fact azyklisch` im obigen Listing ist äquivalent zu folgender Formulierung:

```
fact azyklisch{
  all p:Person | not p in p.^(vater + mutter)
}
```

(b) Für jede binäre Relation `r` gilt, dass die beiden folgende Aussagen äquivalent sind:

```
no ^r & iden
```

und

```
no x:univ | x in x.^r
```

33. Join-Operator

Überprüfen Sie mit Hilfe des Alloy-Analyzers,

(a) ob der Operator $\boxed{.}$ (Dot-Join) assoziativ ist.

(b) ob die Reihenfolge bei der gemeinsamen Anwendung der Operatoren $\boxed{\hat{}}$ und $\boxed{\tilde{}}$ egal ist. Mit anderen Worten gilt stets ^(~r) =~(^r)?

34. Queue

Modellieren Sie eine Queue vom Typ FiFo in Alloy.

(a) Beginnen Sie mit der statischen Struktur.

(b) Erweitern Sie das Modell, so dass man die Operation `add` (Hinzufügen eines Elements) und `poll` (Herausnehmen des Kopf-Elements der Queue) analysieren kann.

35. Aufspannender Baum eines Graphen

(a) Modellieren Sie einen ungericheteten, zusammenhängenden Graphen in Alloy.

(b) Ein aufspannender Baum S (`spanning tree`) eines Graphen G ist ein Baum, der alle Knoten von G überdeckt. Modellieren Sie einen Algorithmus zur Ermittlung eines aufspannenden Baums in Alloy.

36. Drei gewinnt

Betrachten das Spiel „Drei gewinnt“. Ein anderer bekannter Name für dieses Spiel ist „Tic Tac Toe“.

Modellieren Sie den Verlauf von Partien des Spiels Drei gewinnt in Alloy.

Rev 10.0 – 8. Oktober 2020