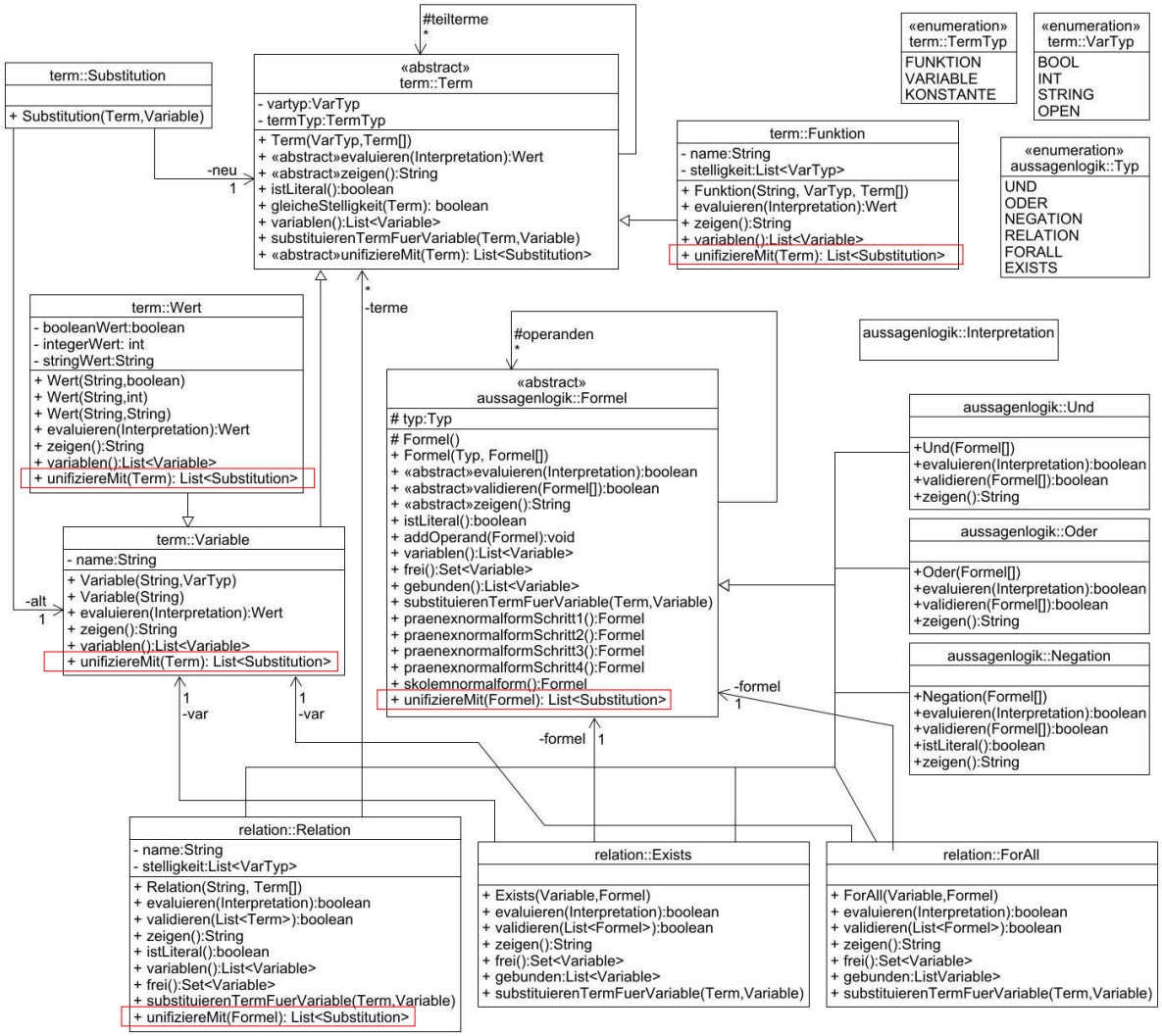
Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Logik

Sommersemester 2022
7. Aufgabenblatt
Variante 2 (spannender)



Das Klassendiagramm zeigt den vollständigen Aufbau nach drei Aufgabenblättern, aktuell zu erstellende Teile sind mit einem roten Kasten markiert. Für Freunde präziser Klassendiagramme: <<nutzt>>-Beziehungen sind nicht eingezeichnet, weiterhin werden existierende und nutzbare get- und set-Methoden weggelassen.

Aufgabe 10 (5 Punkte) [Unifikation, Theorie und Umsetzung]

a) Überlegen Sie sich drei Terme t1, t2 und t3, dabei soll t1 mit t2, t2 mit t3, aber t3 nicht mit t1 unifizierbar sein.

b) Geben Sie für jeden Term und jeweils die anderen Terme an, ob sie miteinander unifizierbar sind und falls ja, wie der allgemeinste Unifikator aussieht.
1) f(x,y)
2) f(z,z)
3) f(g(f(x,y)),g(v))
4) f(g(f(y,y)),g(x))

c) Realisieren Sie die im Code angedeuteten Methoden mit Namen *unifiziereMit(.)* zur Unifikation von *Formel*n, *Relation*en und *Term*en. Das Ergebnis wird jeweils in einer Liste von *Substitution*sobjekten festgehalten. Ein *Substitution*sobjekt beinhaltet jeweils eine Variable *alt* und den für sie substituierten Term *neu*. Ist eine Unifikation nicht

möglich, ist das Ergebnis null. Am Ende müssen zumindest die Tests aus *test.UnifikationTest* laufen.
Das folgende Beispielprogramm

```
public static void main(String[] args) {
  Formel f1 = new Relation("r", new Variable("a", VarTyp.INT)
      , new Funktion("f1", VarTyp.INT
                    , new Variable("a", VarTyp.INT)));
  Formel f1alt = f1.deepClone();
  Formel f2 = new Relation("r"
      , new Funktion("f1", VarTyp.INT, new Wert("__ano", 42))
      , new Variable("b", VarTyp.INT));
  Formel f2alt = f2.deepClone();
  List<Substitution> subs = f1.unifiziereMit(f2);
  System.out.println(f1alt.zeigen() +"\n" + f2alt.zeigen()
      + "\nUnifikator: " + subs
      + "\nunifiziert: " + f1.zeigen()
      + "\nunifiziert: " + f1.zeigen());
}
```

liefert folgendes Ergebnis.

```
r(a, f1(a))
r(f1(42), b)
Unifikator: [[a:=f1(42)], [b:=f1(f1(42))]]
unifiziert: r(f1(42), f1(f1(42)))
unifiziert: r(f1(42), f1(f1(42)))
```

Anmerkung: Mit dem Unifikationsalgorithmus und einfachen Überprüfungen, ob es sich bei einer prädikatenlogischen Formel um ein Logik-Programm der Prädikatenlogik handelt, ist es recht zügig möglich, den Kern eines Prolog-Systems selbst zu schreiben.