

Vorlesung Perlen der Informatik II

Aufgabe 1 Unifikation in Java

Programmieren Sie das Verfahren zur syntaktischen Unifikation von Termen aus der letzten Vorlesung in Java. Die Substitutions-Operation soll hierbei durch *Pointer-Updates* implementiert werden. Beachten Sie, daß bei dieser Implementierungstechnik Variablen gleichen Namens durch ein und dasselbe Objekt dargestellt werden müssen (*“shared”*). Als Rahmen für die Implementierung könnte die folgende Klassendefinition nützlich sein:

```
public abstract class Term {
    public abstract boolean isVariable();

    public abstract boolean isApplication();

    public abstract Vector argsOf();

    public abstract String nameOf();

    public abstract String stringOf();

    public abstract boolean occurs(String s);

    public abstract void unify(Term t) throws CannotUnify;
}
```

- Überlegen Sie sich, wie Terme in Java dargestellt werden können
- Finden Sie geeignete Implementierungen für die Methoden in der obigen abstrakten Klasse.
- Schreiben Sie zum Testen eine Funktion zum Parsen von Termen. Hierfür könnte die Klasse `StreamTokenizer` nützlich sein, die einen Strom von Zeichen in Tokens (d.h. Wörter oder Sonderzeichen) umwandelt. Um Variablen von Funktionssymbolen unterscheiden zu können, kann die in Prolog übliche Konvention verwendet werden, Variablennamen mit *Großbuchstaben* und Funktionsnamen mit *Kleinbuchstaben* zu beginnen.
- Testen Sie die Unifikationsfunktion an den folgenden Problemen:
 - $f(X, g(X, X))$ und $f(h(Y), g(Z, h(a)))$
 - $f(X, X)$ und $f(g(Y), g(Z))$
 - $f(X, X)$ und $f(g(Y), Y)$
 - $g(X_1, X_2, X_3, X_4, X_5, X_6)$ und
 $g(f(X_0, X_0), f(X_1, X_1), f(X_2, X_2), f(X_3, X_3), f(X_4, X_4), f(X_5, X_5))$