

Übungsblatt 3

Revision: 1.0

Ein Meta-Modell für “MINI-C”

In der Vorlesung wurde argumentiert, dass die Ausdrucksmächtigkeit des GOPPRR-Meta-Meta-Modells nicht geringer als die der EBNF ist, da sich eine in EBNF gegebene Syntaxdefinition als Meta-Modell in GOPPRR realisieren lässt. Das Ziel dieses Übungsblattes ist es, das in der Vorlesung skizzierte Beispiel in *MetaEdit+* umzusetzen.

Gegeben sei folgende Syntaxdefinition in EBNF zur Definition von Funktionen in einer C-artigen (im Umfang jedoch sehr stark eingeschränkten) Sprache:

```
IDENT = C-like identifier matching [a-zA-Z_][a-zA-Z0-9_]*
EXP = C-like expression
QUALIFIER = 'static' | 'inline'
VAR_TYPE = 'int' | 'bool' | 'float' | 'char' | IDENT
RET_TYPE = 'void' | VAR_TYPE
PARAM_DEF = 'void' | VAR_TYPE IDENT { ',' VAR_TYPE IDENT } [ ',' '...' ]
FUNC_BODY = { VAR_TYPE IDENT ';' } { STMT }
STMT = IDENT '=' EXP ';'
      | 'if' '(' EXP ')' '{' { STMT } '}' [ 'else' '{' { STMT } '}' ]
      | 'while' '(' EXP ')' '{' { STMT } '}'
FUNC_DEF = { QUALIFIER } RET_TYPE IDENT '(' [ PARAM_DEF ] ')' '{' FUNC_BODY '}'
```

Neben den Anforderungen, die sich bereits aus den EBNF-Regeln ergeben, muss eine Funktionsdefinition auch folgende Bedingungen erfüllen, die sich in EBNF nur umständlich ausdrücken lassen bzw. eher der statischen Semantik zuzuordnen sind:

1. In `FUNC_DEF` dürfen vor `RET_TYPE` genau 0, 1 oder 2 `QUALIFIER` stehen.
2. Wenn in `FUNC_DEF` 2 `QUALIFIER` angegeben sind, müssen diese unterschiedlich sein.
3. Der Funktionsname (`IDENT` in `FUNC_DEF`) sowie alle innerhalb einer Funktion definierten Variablen (`IDENT` in `FUNC_BODY`) müssen eindeutig sein.
4. In `STMT` muss `IDENT` der Identifier einer innerhalb der Funktion deklarierten Variable sein.

Aufgabe: Erstellen Sie ein Meta-Modell in *MetaEdit+*, das dazu verwendet werden kann, Funktionen gemäß den obigen Vorgaben zu definieren. Verwenden Sie *Bindings* und *Constraints* und schreiben Sie *Generatoren* mit denen die Korrektheit (hinsichtlich dieser Vorgaben) der in Ihrem Meta-Modell definierbaren Modelle forciert wird bzw. nachträglich überprüft werden kann.

Abgabe: 21.12.2010 bis 16:00 Uhr