„Generic Interpreter“

Christian Gierds

Seminar Compilerbau

12.06.03

Übersicht

- Allgemeines zum GI
- Fähigkeiten des GI
- Grammatik einbinden (allgemein)
- Lexikalische Ausdrücke definieren
- Syntax definieren
- Semantik
- Komplexes Beispiel (Minako)

Allgemeines

- Home: http://www.csupomona.edu/~carich/gi/
- Autor: Craig A. Rich <carich@acm.org>
- Version 1.0
- Status: wird fortgesetzt (Konfliktbehandlung?)

Fähigkeiten

- Grammatiken: LL(1), LR(0), SLR(1), LR(1)
- keine Warnung bei Konflikten
- schlechte Unterscheidung von Terminals und Non-Terminals (für „faule“ Programmierer)

Beispiel

- CopCow.java

Grammatik einbinden (allgemein)

- neue Grammatikklasse erbt von Grammatiktypklasse
- Konstruktor leer oder „statische“ Grammatikregeln
- Regeln können dynamisch hinzugefügt werden
- danach wird DataStream/String „interpretiert“

Lexik einbinden

- `Lexicon.put("token" , Lexicon.Expression expr)`
  wobei `expr =`
  - `new Lexicon.Singleton(String literal)`
  - `new Lexicon.Match( ... )`
  - `new Lexicon.Union(Lexicon.Expression expr1, expr2)`
  - `new Lexicon.Repetition( ... )`
  - `...`
  - `Lexicon.expression(String ere)`

Syntax

- „GI“ kennt kein EBNF
- rule ::= token1 token2 ... | token3 token 4 ... | ...

```
Grammar.put(“rule“, new Object [][] {
    { “token1“, “token2“, ... },
    { “token3“, “token4“, ... },
    ...
});
```

Semantik

- Definition als Variable vom Typ `Semantics`

```
Semantics sem_rule = new Semantics() {
    public void evaluate(ParseTree tree) { ... }
};
```

  - Wurzel eines Teilbaumes ist `Object tree.attribute`
  - Kindknoten ist `Object tree.phrase[i].attribute`

-

```
Grammar.put("rule", new Object [][] {
    { "token1", "token2", ... , sem_rule },
    ...
});
```

Beispiel

- Rechner2.java
- Minako.java

Fragen