Compilerbau für die Common Language Run-Time

Statische Analyse und Synthese

© 2006 Martin v. Löwis

Ansätze für Analyse und Synthese

- Abstrakte Syntax
- Symboltabellen
- Modelle, Modelltransformation
- ANTLR: tree parsing
- kimwitu: unparsing
- ...

© 2006 Martin v. Löwis

Synthese mit ANTLR

- imperativ: Algorithmus auf AST

```
public interface AST{
  //...
  AST getFirstChild();
  AST getNextSibling();
  String getText();
  int getType();
  int getNumberOfChildren();
}
```

© 2006 Martin v. Löwis

Synthese mit ANTLR

- deklarativ: tree parser
  - EBNF-ähnliche Grammatik
  - rule : alternative | alternative | ... ;
- Alternativen: Mustervergleich
  - #(root-token child1 child2 ...)
  - #(PLUS INT INT)
  - #(IF expr stmt (stmt)?)

© 2006 Martin v. Löwis

Tree Parser

```
class CalcTreeWalker extends TreeParser;
expr returns [int r]
    {  int a,b; r = 0; }
    :      #(PLUS a=expr b=expr) { r = a+b; }
    |      #(STAR a=expr b=expr) { r = a*b; }
    |      i:INT                 { r = Integer.parseInt(i.getText()); }
```

© 2006 Martin v. Löwis