

Software Tools Exercises

Part II - Lecture/Tutorial 10

Today's Outline

- The Exam
- Grammar Exercise
- Type Derivation Exercise

2009
YEAR

COMPSCI 732

The University of Auckland | New Zealand



The Exam

The Exam

- Two hours time
- 50% about first half, **50%** about second half
- Second part will include:
 - 1. Compilers (15%):**
define regular expressions and a context-free grammar with actions according to a specification
 - 2. Type Systems (15%):**
derive a given Java program part using an environment and type rules that are given
 - 3. Text Questions (20%):**
short answer questions (3 sentences each) about DALs, version control, compilers, static checking, type systems

Grammar Exercise



Regular Expressions

Define regular expressions for the following tokens:

1. The `package` keyword
2. Boolean literals that are either `true` or `false`
3. Hexadecimal numbers with digits 0 to 9 and A to F
4. Identifiers that start with an alphabetic lower-case character followed by an arbitrary sequence of lower-case alphanumeric characters

```
PACKAGE: 'package';
```

```
BOOLEAN: 'true' | 'false';
```

```
HEXNUM: ('0'..'9'|'A'..'F')+;
```

```
IDENTIFIER: ('a'..'z') ('a'..'z'|'0'..'9')*;
```

Context-Free Grammars

Define grammar rules for the following syntax elements:

1. Expressions that are variable accesses or use the binary operators + and *
2. The while statement
3. An interface definition with an optional extends clause

You can use the following tokens and subrules:

**PLUS, STAR, IDENTIFIER, WHILE, LPAR ("("), RPAR (")"),
INTERFACE, EXTENDS, COMMA, LCURLY ("{"), RCURLY ("}"),
statement, interfaceBody**

expr: expr (PLUS|STAR) expr | IDENTIFIER ;

while: WHILE LPAR expr RPAR statement ;

**interface: INTERFACE IDENTIFIER
(EXTENDS IDENTIFIER (COMMA IDENTIFIER)*)?
LCURLY interfaceBody RCURLY ;**

Actions

Given the following grammar rule:

```
expr: expr PLUS expr
    | expr STAR expr
    | INT
    | LPAREN expr RPAREN ;
```

Rewrite the rule using ANTLR syntax so that it returns the `int` value that is the arithmetic result of the parsed expression.

```
expr returns [int value]:
    a=expr PLUS b=expr
    { $value = $a.value + $b.value; }
    | a=expr STAR b=expr
    { $value = $a.value * $b.value; }
    | i=INT { $value = Integer.parse($i.text); }
    | LPAREN a=expr RPAREN { $value = $a.value; };
```


Type Derivation



Type Derivation



Given the environment

$\Gamma = \{\text{int } a; \text{String } b; \text{int } m(\text{String } s, \text{int } t); \}$

derive the following code:

```
if (a==1) a = m(b, 1);
```

$$[\text{int lit}] \frac{\Gamma \vdash \diamond \quad x \in \text{int}}{\Gamma \vdash x:\text{int}} \quad [\text{int ==}] \frac{\Gamma \vdash \text{expr}_1:\text{int} \quad \Gamma \vdash \text{expr}_2:\text{int}}{\Gamma \vdash \text{expr}_1==\text{expr}_2:\text{boolean}}$$

$$[\text{var}] \frac{\Gamma \vdash \diamond \quad \{\text{type } id; \} \subseteq \Gamma}{\Gamma \vdash id:\text{type}} \quad [\text{if}] \frac{\Gamma \vdash \text{expr}:\text{boolean} \quad \Gamma \vdash \text{stat}}{\Gamma \vdash \text{if}(\text{expr}) \text{stat}}$$

$$[\text{call}] \frac{\Gamma \vdash \text{expr}_1:\text{type}_1 \quad \dots \quad \Gamma \vdash \text{expr}_n:\text{type}_n \quad \{\text{type}_{ret} \text{ id}(\text{type}_1 \text{ id}_1, \dots, \text{type}_n \text{ id}_n); \} \subseteq \Gamma}{\Gamma \vdash \text{id}(\text{expr}_1, \dots, \text{expr}_n):\text{type}_{ret}}$$

$$[\text{assign}] \frac{\Gamma \vdash \text{expr}:\text{type} \quad \{\text{type } id; \} \subseteq \Gamma}{\Gamma \vdash id=\text{expr};}$$

Type Derivation Solution

Given the environment

$\Gamma = \{\text{int } a; \text{String } b; \text{int } m(\text{String } s, \text{int } t); \}$

derive the following code:

```
if (a==1) a = m(b, 1);
```

$$[var] \frac{\Gamma \vdash \diamond \quad \{\text{int } a;\} \subseteq \Gamma}{\Gamma \vdash a:\text{int}}$$

$$[int \textit{lit}] \frac{\Gamma \vdash \diamond \quad 1 \in \text{int}}{\Gamma \vdash 1:\text{int}}$$

$$[var] \frac{\Gamma \vdash \diamond \quad \{\text{String } b;\} \subseteq \Gamma}{\Gamma \vdash b:\text{String}}$$

$$[int ==] \frac{\Gamma \vdash a:\text{int} \quad \Gamma \vdash 1:\text{int}}{\Gamma \vdash a==1:\text{boolean}}$$

$$[call] \frac{\Gamma \vdash b:\text{String} \quad \Gamma \vdash 1:\text{int} \quad \{\text{int } m(\text{String } s, \text{int } t); \} \subseteq \Gamma}{\Gamma \vdash m(b, 1):\text{int}}$$

$$[assign] \frac{\Gamma \vdash m(b, 1):\text{int} \quad \{\text{int } a;\} \subseteq \Gamma}{\Gamma \vdash a=m(b, 1);}$$

$$[if] \frac{\Gamma \vdash a==1:\text{boolean} \quad \Gamma \vdash a=m(b, 1);}{\Gamma \vdash \text{if}(a==1) a=m(b, 1);}$$