apl. Prof. Dr. Thomas Noll
Christoph Matheja, Matthias Volk

Compiler Construction 2016

— Series 7 —

Hand in until July 5th before the exercise class.

General Remarks

- It is allowed to hand in your solutions for the theoretical part via email **as a separately attached PDF file**. If you hand in solutions via email please make sure that the size is reasonable ($<$ 1 MB). Solutions that our printer rejects to print within 30min due to large scanned images will not be corrected.
- Please hand in your solutions in groups of 3 or 4.

Exercise 1 (3 Points)

Write an unambiguous grammar for arithmetic expressions (containing addition, multiplication and parenthesis). Define an attribute *Val* to find the value of an expression. Evaluate $2 * 5 + 3$. For this, give the parse tree of this expression, set up the corresponding equation system and solve it.

Exercise 2 (4 Points)

Give a context-free grammar for the language $\{a\}^+$. Extend that grammar with attributes so that the language is the following set:

1. $L = \{a^{2^n} | n \in \mathbb{N}\}$.
2. $L = \{a^{n^2} | n > 0\}$.

You may use any (finite) number of attributes, conditional updates, simple arithmetic and comparison between numbers. However, you may not use a predicate that directly checks whether a given number (in decimal or binary encoding) is a power of two or not.

Exercise 3 (4 Points)

Consider the following grammar $G = (N, \Sigma, P, S)$ with inherited attributes $i1, i2$ and synthesised attributes $s1, s2$.

$$
\begin{array}{lcll}
S' & \to & S & i1.0 = 1 \\
 & & & i2.0 = 2 \\
 & & & i1.1 = s1.1 \\
 & & & i2.1 = i1.0 \\
 & & & s2.0 = s2.1 \\
S & \to & AA & i1.1 = s1.1 \\
 & & & i2.1 = i1.0 \\
 & & & i1.2 = 0 \\
 & & & i2.2 = i2.0 \\
 & & & s1.0 = s2.1 \\
 & & & s2.0 = s2.2 \\
S & \to & A & i1.1 = 0 \\
 & & & i2.1 = i2.0 \\
 & & & s2.0 = s2.1 \\
A & \to & a & s1.0 = 0 \\
 & & & s2.0 = i1.0 \\
A & \to & b & s2.0 = 0 \\
 & & & s1.0 = i2.0
\end{array}
$$

(a) Provide the dependency graph for each production in G.

(b) Apply the circularity test from the lecture to G.

1. Calculate the set $IS(A)$ for all $A \in N$.
2. Is G circular? Justify your answer.

Exercise 4 (* 3 Bonus Points)

In this task we implement a semantic check. In our language we require that every variable identifier is *declared* before the variable is used (read or set). Additionally, a variable defined inside a scope like an `if` statement or a `while` loop is not visible outside this scope. We do not care whether a variable has been *initialised* before it is read. Examples:

This is valid:

```
int x; int y;
if (x <= y) {
    write("Hello world.");
}
write(x);
```

This is not valid (`y` is undefined and `z` is undefined outside the `if`-statement):

```
int x;
if (x <= y) {
    int z;
}
write(z);
```

Implement `checker.DeclarationChecker.checkDeclaredBeforeUsed()`. *Hint: for this you do not need to implement any attributed grammars and their evaluation. Instead simply walking the abstract syntax tree once and checking the required property suffices.*