

Exercises Syntax-Semantics

Problem 1

Mitchell exercises 4.1 and 4.2:

4.1:

Draw the parse tree for the derivation of the expression 25 according to the grammar given in subsection 4.1.2. (This is the grammar defined in slide no 8 from the lecture.) Is there another derivation for 25? Is there another parse tree?

4.2:

Draw parse trees for the following expressions, assuming the grammar and precedence as described in Example 4.2, that is

$e ::= 0 \mid 1 \mid e+e \mid e-e \mid e*e,$

(a) $1 - 1 * 1$

(b) $1 - 1 + 1$

(c) $1 - 1 + 1 - 1 + 1$, if we give $+$ higher precedence than $-$

The grammar is the one defined in slide no 8 from the lecture. $*$ has a higher precedence than $+$ and $-$, which means that an expression $5*4-3$ will be parsed as if it were written $(5*4)-3$.

Problem 2

Make a syntax diagram for the following Extended BNF:

```

program ::= { statement* }
statement ::= assignment | if-then-else | while
assignment ::= identifier = exp
if-then-else ::= if exp { statement+ } |
                if exp { statement+ } else {statement+ }
while ::= while exp { statement+ }
exp ::= identifier | number | (exp) | exp operator exp

```

Problem 3

a) Make the parse tree (according to the grammar in Problem 2) for

```

{ i=1
  while i<10 {
    i=i+1
    j=j+i
  }
}

```

Assume that operator may be one of the ordinary operators and that 'i' and 'j' are legal identifiers.

b) Propose a corresponding abstract syntax tree (AST). What is the difference between an AST and a parse tree/concrete syntax tree?

Problem 4

a) Make a meta-model corresponding to the grammar in Problem 2.

Hint: Think of how an abstract syntax tree would be represented by means of a structure of objects, and make the class model that reflects this.

Note that there is not one and only one correct meta-model, so just make one that you think may represent programs in this language.

b) Assume that the grammar is extended, so that program starts with a number of declarations of variables, each with a name and a type. Assume further that the meta-model therefore will have a class 'type' that represents the common properties of all types in the language. Modify the meta-model so that programs may now have declarations, and so that an identifier is modeled by a link to the object representing the declaration of the corresponding variable.