TDT4205 Problem Set 3
Spring 2012

PART 1 - Theory due Wed Feb. 22^{th}, 20:00.
PART 2 - Programming due Wed Feb. 29^{th}, 20:00.

Part 1 needs to be passed in order for Part 2 to be evaluated! ALL answers are to be submitted to *itslearning*

ALL OF THIS ASSIGNMENT IS TO BE DONE INDIVIDUALLY. Cheating ("koking"), including using partial solutions from other students, will cause a failing grade for the whole course. We will be checking for this using a plagerism detecting as well as looking for suspiciously similar submissions.

All submitted source code MUST be able to compile and run on asti. **This assignment counts towards your final grade.** Please read the assignment guidelines on itslearning before starting to work on the assignment. Requests for clarifications can be posted on the itslearning forum.

What to turn in

When turning in assignments, please turn in two files:

- (your_username)_answers.pdf : Answers to non-programming questions (Part 1)

PART 1 - Theory (40%)

Task 1.1: Parsing (20%)

1. LL(k) parsing can be extended to an unbounded amount of lookahead by allowing the parser to decide the choice of production based on testing the remaining token stream against a finite set of regular languages. Does this resolve the problem with left-recursion? Explain.

2. Consider the grammar

   F → f I v A w S x
   A → P
   P → P I | ϵ
   S → S s | s
   I → i

   Write an equivalent grammar which is not left-recursive.

3. Tabulate FIRST and FOLLOW for each nonterminal, and construct the LL(1) parsing table for the resulting grammar. Show your work!

4. Using your constructed LL(1) parsing table, construct the top-down parse tree of the program `f i v i i i i w s s s x` (Show every step)

5. Show the steps in bottom-up parsing of `f i v i i i i w s s s x` using the original grammar in task 1.1.2. Show the contents of the parser stack for each step.

Task 1.2: Symbol Tables (10%)

1. What kind of data structure is typically used for symbol tables, and why?

2. Suppose that our VSL language supports pointers to functions. E.g. consider the following program:

```
FUNC main()
{
    VAR f, b
    b := 42

    f := dostuff
    f(a)
}

FUNC dostuff(a)
{
    PRINT "The value of a is ", a
}
```

   In your own words, describe why this may pose a problem with regards to detecting errors at compile time.

3. Suggest a solution to how you can solve this by including additional information in the symbol table.

4. Suggest a symbol table entry for the symbol `f`.

Task 1.3: Syntax-Directed Translations (10%)

1. What is a syntax-directed definition (SDD)? Can you give an example of when SDDs are useful?

2. What is the difference between L-attributed and S-attributed syntax-directed definitions? What does Bison support? Please explain.

3. Given the grammar

   $$\begin{aligned} &E \rightarrow E + T \mid T \\ &T \rightarrow \textbf{num}\ ,\ \textbf{num} \mid \textbf{num} \end{aligned}$$

   Assume there is two types: float (decimal numbers) and int (integers). Give an SDD to determine the type of each term T and expression E.