TDT4165 PROGRAMMING LANGUAGES

Fall 2012

Exercise 05
Grammars

Before you begin

Read section 2.1.1 in V&H and study the lecture notes from the lecture about grammars. Make sure you are familiar with the following terms: *syntax*, *grammar*, *BNF*, *EBNF*, *parsing*, *ambiguity*, *associativity*, *precedence*, *left-recursive*, *right-recursive*.

What you must hand in

- `answers.txt`: Answers to the tasks below.
- `questions.txt` (optional): Any questions you want answered. Answers will come by email or on the website.
- `feedback.txt` (optional): Your thoughts about the course so far. What are the main difficulties? How can we improve the course? Anything else you would like to say.

Task 1 Grammar for arithmetic expressions

a) Write a grammar in EBNF-notation for arithmetic expressions with the operators $+$, $-$, $*$ and $/$. The operators are binary (taking two operands) and infix (written between the operands), and each application of an operator to its operands must be enclosed within parentheses. The only type of value is integer. Here is an example expression:

```
((1 - 2) + (3 * (4 / 5)))
```

b) Suppose we remove the requirement that each application of an operator to its operands must be enclosed within parentheses. (An application still can be enclosed, but it doesn't have to. Explain two problems that will arise from this change and how they can be solved. Write a new grammar that takes these problems into account.

Task 2 Evaluating arithmetic expressions

Suppose we represent our arithmetic expressions as nested Oz records. The application of an operator is represented by a record with the name of the operator as its label and the records for the operands as its fields. Integers are represented as records with the label `int` and the Oz integer value as its field. Here is the record that would correspond to the example above.

```
plus(minus(int(1) int(2)) times(int(3) divide(int(4) int(5))))
```

a) Write a function `Evaluate` that takes such a record as input and returns the result of evaluating the expression it represents.

b) Write a function `DivByZeroCheck` that takes such a record as input and returns `true` if the evaluation of the expression it represents will involve a division by zero and `false` if it will not.

Task 3 Grammars and parsing

Here is an EBNF grammar for binary numbers:

```
<number> ::= <digit> { <digit> }
<digit>  ::= 0|1
```

a) Write a grammar for binary numbers in BNF notation. BNF notation is a subset of EBNF notation that does not contain the symbols { and].

b) There are two similar but distinct ways to solve the preceding subtask. (Hint: One is called left-recursive and the other is called right-recursive.) The choice between these two can be significant if the grammar will be used as a basis for automatic analysis of strings. Can you guess why?

Task 4 Validating strings

Given the grammar below, determine which of the strings are legal in the language:

```
<S> ::= <A> | <B>
<A> ::= a <A> b | a <C> b | e
<B> ::= a <B> b b | e
<C> ::= c <C> c | e
```

(Where 'e' is the greek letter ϵ and means the empty token)

a) abc

b) acb

c) cab

d) abba

e) aabb

f) babb

g) aabbb

(Optional): If you are feeling up to it, describe all legal strings of the language, using set notation.

Task 5 Understanding a grammar

You are given an (obviously obfuscated) grammar. Which language does it describe?

```
<A> ::= <B> <B> <C>
<B> ::= <A> | <D>
<D> ::= any number
<C> ::= + | - | * | /
```

Task 6 Grammar for subset of SQL

Create a formal grammar in EBNF notation that can construct a subset of sentences in the Server Query Language (SQL). The grammar shall be able to produce sentences like these ones:

select number from telephonebook where city = 'Trondheim'
select firstname, lastname from customers where customerid = 10
select employeename, vacationdays from employees, vacation where empolyeeid = 8453;

(Hint: SQL-specific tokens are 'select', 'from' and 'where')

Task 7 Grammar for roman numerals (Optional if you have done the next task)

Write a grammar for the roman numerals from `I` through `MMMCMXCIX`. You grammar should allow only one roman numeral for each number in the range. If you need to refresh your knowledge of roman numerals, see `http://en.wikipedia.org/wiki/Roman_numerals`. This task may be a bit difficult. There is a hint in the file `hint.txt`, but don't look at it if you don't need it.

(The rules for roman numerals are exempted from the syllabus. You don't need to remember them at the exam.)

Task 8 Grammar for natural language (Optional if you have done the previous task)

Marvin is equipped with a simple natural language processor. He was constructed against his will by the Sirius Cybernetics Corporation, and thus he does not trust the grammar for this piece of hardware. You are going to help him to write a new grammar for understanding English.

Assume that english consists of the following components:

- Nouns
- Verbs
- Adjectives
- Prepositions

Write a grammar which validates the following sentences:

- "Beeblebrox is not in the room"
- "Zaphod is in Oslo"
- "Beeblebrox sits in room Z"
- "Marvin is depressed"