Final Exam
COSE215, Spring 2015

Instructor: Hakjoo Oh

Problem 1. (20pts) Consider the following grammar of arithmetic expression:

$$\begin{array}{lcl} E & \rightarrow & I \mid E+E \mid E*E \mid (E) \\ I & \rightarrow & a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \end{array}$$

This grammar is ambiguous because it does not respect the precedence and associativity of operators. For instance, expression $1+2*3$ can be interpreted as either $(1+2)*3$ or $1+(2*3)$, and expression $1+2+3$ as either $(1+2)+3$ or $1+(2+3)$.

To remove the ambiguity, a common technique is to classify the expressions into *factors*, *terms*, and *expressions*:

1. A *factor* (F) is either an identifier or a parenthesized expression. For instance, $a, b, (a+b), (a*b), \ldots$ are factors. Define factors by a grammar:

$$F \rightarrow$$

2. A *term* (T) is either a product of one or more factors. For instance, $a, b, (a+b), (a*b), a*b, a*(a+b), a*(a*b)$ are all terms. Define terms by a grammar:

$$T \rightarrow$$

3. An *expression* (E) is a sum of one or more terms. For instance, $a*b, a*(a+b), a*b+a*(a+b), \ldots$ are expressions. Define expressions by a grammar:

$$E \rightarrow$$

4. Putting factors, terms, and expressions together, the final unambiguous grammar for the arithmetic expression is defined:

$$\begin{array}{lcl} I & \rightarrow & a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\ F & \rightarrow & \\ T & \rightarrow & \\ E & \rightarrow & \end{array}$$

Draw the parse tree for string $1+2*3$ according to the new grammar.

Problem 2. (15pts) A pushdown automaton (PDA) is an extension of λ-NFA. Answer the following questions:

1. (5pts) A PDA is defined as a seven-tuple:

$$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

Explain each component. (For δ, Z_0, and F, specify their types).

2. (10pts) Design a PDA that accepts the language:

$$L = \{ww^R \mid w \in \{0,1\}^*\}.$$

Problem 3. (10pts) Draw a Venn-diagram to illustrate the relationships between the following classes of languages:

- *RL*: the set of regular languages
- *CFL*: the set of context-free languages
- *UCFL*: the set of context-free languages that have unambiguous grammars
- *PDA*: the set of languages accepted by some pushdown automata
- *DPDA*: the set of languages accepted by some deterministic pushdown automata

Problem 4. (15pts) A pumping lemma for context-free languages can be stated as follows:

> For any context-free language L there exists an integer n, such that for all $z \in L$ with $|z| \geq n$, there exist $u, v, w, x, y \in \Sigma^*$, such that
>
> 1. $z = uvwxy$
> 2. $|vwx| \leq n$
> 3. $|vx| \geq 1$
> 4. for all $i \geq 0$, $uv^iwx^iy \in L$.

1. (5pts) According to the pumping lemma, what is the essential property of CFLs? Compare the property with that of regular languages.
2. (10pts) Use the pumping lemma to show that the following language is not context-free:

$$L = \{0^n 1^n 2^n \mid n \geq 1\}.$$

Problem 5. (10pts) Context-free languages are closed under union but not under intersection. Use this fact to prove that CFLs are not closed under complementation.

Problem 6. (10pts) Design a Turing machine that, given integers x and y, computes $x + y$. Explain how it works.

Problem 7. (10pts) True/False questions:

1. There exists a general algorithm to remove ambiguity from a context-free grammar.
2. There is an algorithm that can tell whether a CFG is ambiguous or not.
3. For every ambiguous CFG, there exists an equivalent yet unambiguous CFG.
4. The language $\{ww^R\}$ can be accepted by some DPDA.
5. Context-free languages are closed under union, concatenation, and closure.
6. All computer programs written in modern languages can be implemented by some Turing machines.
7. It is possible to automatically translating programs written in C into an equivalent Turing machine.
8. The number of undecidable problems is countably infinite.
9. There is a problem solvable by Turing machines with two tapes but unsolvable by Turing machines with a single tape.
10. Every Turing machine can be represented by an expression in lambda calculus.

Problem 8. (10pts) Let us go back to the motivating questions of this class:

- What is computing?
- What can a digital computer do? or cannot do?

Answer the questions as best as you can.