

Comp 412, Fall 2007 Practice Midterm Exam

This practice exam is intended to provide you with example questions of the kind that might appear on the actual midterm exam. In fact, I have used most of these questions on previous midterm exams in Comp 412.

The actual exam is a three-hour, closed-book, closed-notes exam. It has five questions, drawn from Chapters 1 through 4 of the text, plus the lectures through Wednesday, October 10. All of the lecture notes can be found online.

1. **Short Answers** (5 points each) Give a concise answer for each of the following questions.

- (a) Which is smaller: a parse tree or an abstract syntax tree? Why?
- (b) In a finite automaton, what is an ϵ -transition? What role do they play in Thompson's construction?
- (c) Give an example of a left-recursive grammar and a right-recursive grammar. Which is better for a top-down parser?
- (d) Describe a language that *cannot* be described with a regular expression. Explain why the language is not regular.
- (e) State the LL(1) property for a context-free grammar.
- (f) Describe the steps required to convert a regular expression into code for a scanner.

2. **Intermediate Representations** (20 points)

Show how the statement " $x \leftarrow y * (z + 1)$ " might be represented in each of the following intermediate representations.

- (a) abstract syntax tree
- (b) one-address code
- (c) quadruples
- (d) two-address code

3. **Parsing** (25 points)

The following grammar describes a set of expressions defined over the *terminal symbols* name, foo, bar, baz, and quux.

$$\begin{aligned} \langle \text{expr} \rangle & ::= \underline{\text{name}} \langle \text{op} \rangle \langle \text{expr} \rangle \\ & \quad | \underline{\text{name}} \\ \langle \text{op} \rangle & ::= \underline{\text{foo}} \\ & \quad | \underline{\text{bar}} \\ & \quad | \underline{\text{baz}} \\ & \quad | \underline{\text{quux}} \end{aligned}$$

- (a) Rewrite the grammar to encode the correct precedence between operators, given that baz and quux have highest priority, followed by foo, followed by bar.

(b) Assume that you have an LL(1) parser generator that has successfully processed your rewritten expression grammar. How large is the parse table that it produced? Explain how the size of the LL(1) table would compare with an LR(1) table for the same grammar.

(c) Show the parse tree derived from your grammar for the string
name bar name baz name foo name quux name

(d) If you had an LR(1) parser for your grammar, how many *shifts* and how many *reduces* would it perform while parsing the string

name bar name baz name foo name quux name

4. **Attribute Grammars** (25 points) Consider the following attributed grammar. (Aho, Sethi, & Ullman would call it a syntax-directed translation scheme.)

	<i>Production</i>	<i>Evaluation Rules</i>
1	NUM ::= SIGN LIST	LIST.pos \leftarrow 0 NUM.val \leftarrow if SIGN.neg then -LIST.val else LIST.val
2	SIGN ::= +	SIGN.neg \leftarrow false
3	SIGN ::= -	SIGN.neg \leftarrow true
4	LIST ::= TRI	TRI.pos \leftarrow LIST.pos LIST.val \leftarrow TRI.val
5	LIST ₀ ::= LIST ₁ TRI	LIST ₁ .pos \leftarrow LIST ₀ .pos + 1 TRI.pos \leftarrow LIST ₀ .pos LIST ₀ .val \leftarrow LIST ₁ .val + TRI.val
6	TRI ::= 0	TRI.val \leftarrow 0
7	TRI ::= 1	TRI.val \leftarrow $1 \times 3^{\text{TRI.pos}}$
8	TRI ::= 2	TRI.val \leftarrow $2 \times 3^{\text{TRI.pos}}$

(a) Show the parse tree that would be built for the sentence: -2012

(b) Draw the attribute dependency graph for the sentence.

(c) Identify the *synthesized attributes* and the *inherited attributes*. Justify your answer.

(d) Sketch an evaluation method for the attributes.

(e) Show the values your method would assign to each attribute.