CSCI 4220 Quiz

1 Instructions

The following quiz contains multiple answer true/false questions. Recall that a multiple answer question may have one or more correct answers.

On the blackboard there will be an image of this quiz. That is, a quiz having the same questions, same answers, and same order of questions and answers. In order to get credit for taking this quiz, you will need to enter your answers for this quiz on the blackboard. The blackboard will then record your grade for the quiz.

This quiz is over the material covered on slides: 03 Equational Reasoning.ppt, 04 Semantics part 1, and in the discussion below. These slides are available on the blackboard. Please make sure you understand the material on the slides before taking this quiz.

2 Full Parse-Expressions: A Notation for Parse Trees

A parse tree can be unambiguously denoted using a symbolic form that I refer to as a *full parse-expression.* Let G denote an arbitrary context-free grammar G. Let A denote a nonterminal symbol in G and let α denote a string of symbols (terminals and/or nonterminals) such that:

$$A \overset{+}{\Rightarrow} \alpha$$

Let α' denote a string that is derived from α by subscripting all nonterminals appearing in α. Under these assumptions we say that the term $A[\![\alpha']\!]$ is a parse-expression with respect to the grammar G.

2.1 An Example

Consider the following grammar containing the terminal symbols $\{c, d\}$ and the nonterminal symbols $\{A\_list, B, D\}$:

$$\boxed{\begin{array}{lcl} A\_list & ::= & A\_list\ B \mid c \\ B & ::= & B\ D \mid D \\ D & ::= & d \end{array}}$$

Among others, this grammar contains the following full parse-expressions:

$A\_list[\![c]\!]$
$A\_list[\![A\_list_1\ B_1]\!]$
$A\_list[\![A\_list_1\ B_1\ B_1]\!]$
$A\_list[\![A\_list_1\ B_1\ B_2]\!]$
$A\_list[\![cB_1\ B_2]\!]$
$A\_list[\![cB_1\ d]\!]$

$B[\![D_1]\!]$
$B[\![B_1\ D_1]\!]$
$B[\![B_1\ D_1\ D_2]\!]$
$B[\![B_1\ D_1\ D_1]\!]$
$B[\![d]\!]$

The following expressions are **NOT** full parse-expressions with respect to G:

Expression	Comment
$A\_list[\![A\_list_1]\!]$	The length of the derivation must be at least 1.
$A\_list[\![A\_list_1\ B]\!]$	The nonterminal B must be given a subscript.
$A\_list[\![D_1]\!]$	The derivation $A \overset{+}{\Rightarrow} D$ is not possible in G.

3 (Abbreviated) Parse-Expressions: A Notation for Parse Trees

In this class, we will symbolically denote parse trees using a notation that we refer to as *abbreviated parse-expressions* or simply a *parse-expression* for short. For example, instead of $A[\![\alpha']\!]$ we will write $[\![\alpha']\!]$ and leave it to the reader to infer A.

Consider the following derivation:

$$A \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_n$$

We say that the abbreviated parse-expression, $[\![\alpha'_n]\!]$ **derived from** A is well-formed iff:

$$|\alpha_1| > 1 \lor n = 1 \text{ where } |\alpha_1| \text{ denotes the total number of symbols in } \alpha_1$$

The constraint $|\alpha_1| > 1$ formally states that the first step in a multi-step derivation may not be linear (i.e., A must have at least 2 children). The condition $n = 1$ states that any α, regardless of its size, that can be derived in a single derivation step is well-formed.

3.1 An Example

Consider the following grammar containing the terminal symbols $\{c, d\}$ and the nonterminal symbols $\{A\_list, B, D\}$:

A_list	$::=$	$A\_list\ B \mid B$
B	$::=$	$c \mid D$
D	$::=$	d

Among others, this grammar contains the following parse-expressions:

Parse-expression	Derived from
$[\![B_1]\!]$	A
$[\![A\_list_1\ B_1]\!]$	A
$[\![A\_list_1\ B_1\ B_1]\!]$	A
$[\![A\_list_1\ B_1\ B_2]\!]$	A
$[\![c\ B_1\ B_2]\!]$	A
$[\![c\ B_1\ d]\!]$	A
$[\![D_1]\!]$	B
$[\![c]\!]$	B
$[\![d]\!]$	D

4 Matching

Parse-expressions are useful because we will use them in equations to define the semantics of programming language constructs. In this framework, a program can be executed by using these equations to simplify a program until it can be simplified no further. This simplification is based on equational reasoning and requires matching as one of its computational steps. Therefore, to understand how a program can be simplified we must first understand matching in this context.

In the context of parse-expressions, a *match expression* is of the form:

$$[\![\alpha']\!] \ll [\![\beta]\!]$$

In this case, α denotes a string that may contain terminals and subscripted nonterminals and β denotes a string that may only contain terminal symbols. It is also assumed that $[\![\alpha']\!]$ and $[\![\beta]\!]$ are well-formed parse-expressions.

In the context of matching, subscripted nonterminals occurring in parse-expressions are treated as **variables** quantified over the set of all strings they can derive with respect to the given grammar. A match is said to **succeed** if the variables on the left-hand side of the match expression (i.e., the variables to the left of $\ll$) can be instantiated in such a manner so that the left and right sides of the match expression become syntactically equal.

The algorithm for solving a match expression of the form $[\![\alpha']\!] \ll [\![\beta]\!]$ can be summarize as follows:

1. Check that $[\![\alpha']\!]$ and $[\![\beta]\!]$ are well-formed.
2. Find values for variables (if possible) so that when the variables in α' are replaced with their values, the resulting parse-expression is $[\![\beta]\!]$.
3. Make sure that a value assigned to a variable can actually be derived from that variable. That is if A_1 is a variable and v_1 is a value, then $A \overset{+}{\Rightarrow} v_1$ must be possible within the grammar.

4.1 Example

In this example, we use a standard BNF grammar fragment describing a subset of mathematical expressions. We assume that *num* and *ident* are terminal symbols and have the usual meaning.

```
E  ::=  E + T | T
T  ::=  T * F | F
F  ::=  num | ident | ( E )
```

Match Expression	Result of Match
$[\![E_1 + T_1]\!] \ll [\![1 + 2]\!]$	true – $E_1 \overset{*}{\Rightarrow} 1$ and $T_1 \overset{*}{\Rightarrow} 2$
$[\![E_1 + T_1]\!] \ll [\![1 + 1]\!]$	true – $E_1 \overset{*}{\Rightarrow} 1$ and $T_1 \overset{*}{\Rightarrow} 1$
$[\![T_1 * F_1]\!] \ll [\![1 * x]\!]$	true – $T_1 \overset{*}{\Rightarrow} 1$ and $F_1 \overset{*}{\Rightarrow} x$
$[\![T_1 * F_1]\!] \ll [\![x * x]\!]$	true – $T_1 \overset{*}{\Rightarrow} x$ and $F_1 \overset{*}{\Rightarrow} x$
$[\![E_1 + E_2]\!] \ll [\![x + 2]\!]$	false – $[\![E_1 + E_2]\!]$ is not well-formed
$[\![T_1 + T_2]\!] \ll [\![1 + y]\!]$	true – $T_1 \overset{*}{\Rightarrow} 1$ and $T_2 \overset{*}{\Rightarrow} y$
$[\![T_1 + T_1]\!] \ll [\![1 + 2]\!]$	false – T_1 cannot be instantiated to be both 1 and 2 at the same time.
$[\![T_1 * T_2]\!] \ll [\![x * 2]\!]$	false – $[\![T_1 * T_2]\!]$ is not well-formed
$[\![F_1 * F_2]\!] \ll [\![1 * y]\!]$	true – $F_1 \overset{*}{\Rightarrow} 1$ and $F_2 \overset{*}{\Rightarrow} y$
$[\![F_1 * F_1]\!] \ll [\![1 * 2]\!]$	false – F_1 cannot be instantiated to be both 1 and 2 at the same time.
$[\![E_1 + T_1 * F_1]\!] \ll [\![1 + 2 * 3]\!]$	true – $E_1 \overset{*}{\Rightarrow} 1$ and $T_1 \overset{*}{\Rightarrow} 2$ and $F_1 \overset{*}{\Rightarrow} 3$
$[\![T_1 + T_1 * F_1]\!] \ll [\![2 + 2 * 3]\!]$	true – $T_1 \overset{*}{\Rightarrow} 2$ and $F_1 \overset{*}{\Rightarrow} 3$
$[\![T_1 + T_1 * F_1]\!] \ll [\![1 + 2 * 3]\!]$	false – T_1 cannot be instantiated to be both 1 and 2 at the same time.
$[\![E_1 + F_1 * F_2]\!] \ll [\![1 + 2 * 3]\!]$	true – $E_1 \overset{*}{\Rightarrow} 1$ and $F_1 \overset{*}{\Rightarrow} 2$ and $F_2 \overset{*}{\Rightarrow} 3$
$[\![F_1 + F_2 * F_3]\!] \ll [\![1 + 2 * 3]\!]$	true – $F_1 \overset{*}{\Rightarrow} 1$ and $F_2 \overset{*}{\Rightarrow} 2$ and $F_3 \overset{*}{\Rightarrow} 3$
$[\![F_1 + F_2 * T_3]\!] \ll [\![1 + 2 * 3]\!]$	false – $[\![F_1 + F_2 * T_3]\!]$ is not well-formed.

5 Quiz Problems

In the following multiple answer questions you are asked whether a given match expression succeeds (true) or fails (false). When answering these questions, you may assume the following grammar. In this grammar, the symbols *num* and *ident* should be treated as terminal symbols having the usual meaning. Similarly, the symbols {, }, *int*, *bool*, ;, +, $*$, (, and) are also to be treated as terminal symbols. **Note** that the nonterminal stmt_list can derive the empty symbol ϵ.

prog	::=	stmt
stmt	::=	block | assign | dec
block	::=	{ stmt_list }
assign	::=	id = E
dec	::=	type id
type	::=	int | bool
stmt_list	::=	stmt ; stmt_list | ϵ
E	::=	E + T | T
T	::=	T * F | F
F	::=	num | id | (E)
id	::=	ident

Advice: Be careful, some of these questions are tricky.

Problem 1. (10 points)

True or False

1. $[\![\ stmt_1;\ ]\!] \ll [\![\ int\ x;\ ]\!]$
2. $[\![\ stmt_1\ ]\!] \ll [\![\ x = 5;\ ]\!]$

Problem 2. (10 points)

True or False

1. $[\![\ stmt;\ ]\!] \ll [\![\ int\ x;\ ]\!]$
2. $[\![\ id_1 = E;\ ]\!] \ll [\![\ x = 5;\ ]\!]$

Problem 3. (10 points)

True or False

1. $[\![\ dec_1; stmt\_list_1\ ]\!] \ll [\![\ int\ x;\ ]\!]$
2. $[\![\ stmt_1; stmt_2\ ]\!] \ll [\![\ int\ x;\ x = 5;\ ]\!]$

Problem 4. (10 points)

True or False

1. $[\![\ stmt_1;\ stmt_2;\ ]\!] \ll [\![\ int\ x;\ y = 5;\ ]\!]$
2. $[\![\ \{stmt\_list_1\}\ ]\!] \ll [\![\ \{int\ x;\ y = 5;\}\ ]\!]$

Problem 5. (10 points)

True or False

1. $[\![\ \{stmt_1;\ stmt_2;\}\ ]\!] \ll [\![\ \{int\ x;\ y = 5;\}\ ]\!]$
2. $[\![\ \{stmt_1;\ stmt\_list_1;\}\ ]\!] \ll [\![\ \{int\ x;\ y = 5;\}\ ]\!]$

Problem 6. (10 points)

True or False

1. $[\![\ id_1 = id_1\ ]\!] \ll [\![\ x = x\ ]\!]$
2. $[\![\ id_1 = id_1;\ ]\!] \ll [\![\ y = y;\ ]\!]$

Problem 7. (10 points)

True or False

1. $[\![\ \{int\ id_1;\ id_1 = E_1;\}\ ]\!] \ll [\![\ \{int\ x;\ x = 4 + 5;\}\ ]\!]$
2. $[\![\ \{type_1\ id_1;\ id_2 = E_1;\}\ ]\!] \ll [\![\ \{int\ x;\ x = 4 + 5;\}\ ]\!]$

Problem 8. (10 points)

True or False

1. $[\![\ \{int\ id_1;\ id_2 = E_1 + E_2;\}\ ]\!] \ll [\![\ \{int\ x;\ x = 4 + 5;\}\ ]\!]$
2. $[\![\ \{int\ id_1;\ id_2 = y * T_1;\}\ ]\!] \ll [\![\ \{int\ x;\ x = y + 5;\}\ ]\!]$

Problem 9. (10 points)

True or False

1. $[\![\ \{int\ id_1;\ id_2 = id_1 + 5;\}\ ]\!] \ll [\![\ \{int\ x;\ x = y + 5;\}\ ]\!]$
2. $[\![\ \{type_1\ id_1;\ id_2 = id_1 + 5;\}\ ]\!] \ll [\![\ \{int\ x;\ y = x + 5;\}\ ]\!]$

Problem 10. (10 points)

True or False

1. $[\![\ stmt_1;\ \{stmt\_list_1\};\ id_2 = id_1 + 5;\ ]\!] \ll [\![\ int\ x;\ \{x = 2;\};\ x = x + 5;\ ]\!]$
2. $[\![\ int\ id_1;\ \{stmt\_list\};\ stmt_1;\ ]\!] \ll [\![\ int\ x;\ \{x = 2;\};\ x = x + 5;\ ]\!]$