

Assignment 1

Jonathan Money
348 074

- 1) a) The applications that use PL/I are scientific and business applications.
- b) The advantages of using the keywords begin and end is to improve readability, though the disadvantage is that the complexity of the language increases due to more keywords.
- c) 3 control statements were added to FORTRAN IV to get FORTRAN 77, character string handling, logical loop control statements, and the IF-THEN-ELSE statement
- d) C++ includes features of C that are known to be unsafe because a goal of C++ was to be able to use it for every application for which C could be used. Another goal of C++ was that there shouldn't be a significant performance penalty compared to C. Fixing some of these unsafe features would have lead to a performance penalty
- e) Two reasons why Ada is safer than C are, the use of packages leading to support for abstraction, and exception handling leading to the programmer having more control in the event of an error or exception. Other reasons why Ada is safer than C include generic program units and concurrency.
- f) The applications that use FORTRAN are scientific applications. The primary reason why FORTRAN is used for scientific applications is for the efficiency it provides. No language after FORTRAN was much better and that is why it is still used even today.

g) Type declaration statements on simple variables give a significant increase to the readability of a language. A good example of this is boolean values. Since a boolean is on or off one could define a variable equal to 1 or 0 giving something like power = 1. We know this variable is a boolean value cause we knew that when we created it, but to an outside reader they may assume it to be the number 1. Using something like power=true improves the readability because the keyword true shows that it is definitely a boolean value. Another example of this could be the case of characters. We could define some variable equal to a specific unicode character number, e.g. char=42, and an outside reader may once again mistake this for a number instead of the asterisk character. If we did something like char='*' we see that the character is the asterisk and the single quotes tell us further "this is the character".

h) The main advantage to single line comments is that an ending syntax is not required thus one can not make the mistake of commenting all their code accidentally. The main disadvantage to single line comments is when you have a longer comment which should span many lines. In this case one can put a very long comment on one line, which is very hard to read, or make it span multiple lines, making sure each line begins with the comment syntax, which is very tedious. The biggest advantage to multiline comments is that a longer comment can span multiple lines with less comment syntax. Large areas of the program can also be commented out easier. The biggest disadvantage to multi line comments is the ending syntax is often forgotten and errors occur.

2. a) $a = a + (b/c - b)$ - Right most Derivation

$$\langle \text{assign} \rangle \Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + \langle \text{term} \rangle$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + \langle \text{factor} \rangle$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (\langle \text{expr} \rangle)$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (\langle \text{expr} \rangle - \langle \text{term} \rangle)$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (\langle \text{expr} \rangle - \langle \text{factor} \rangle)$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (\langle \text{expr} \rangle - \langle \text{id} \rangle)$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (\langle \text{expr} \rangle - b)$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (\langle \text{term} \rangle - b)$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (\langle \text{term} \rangle / \langle \text{factor} \rangle - b)$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (\langle \text{term} \rangle / \langle \text{id} \rangle - b)$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (\langle \text{term} \rangle / (-b))$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (\langle \text{factor} \rangle / (-b))$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (\langle \text{id} \rangle / (-b))$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle + (b / (-b))$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{term} \rangle + (b / (-b))$$

$$\Rightarrow \langle \text{id} \rangle = \langle \text{factor} \rangle + (b / (-b))$$

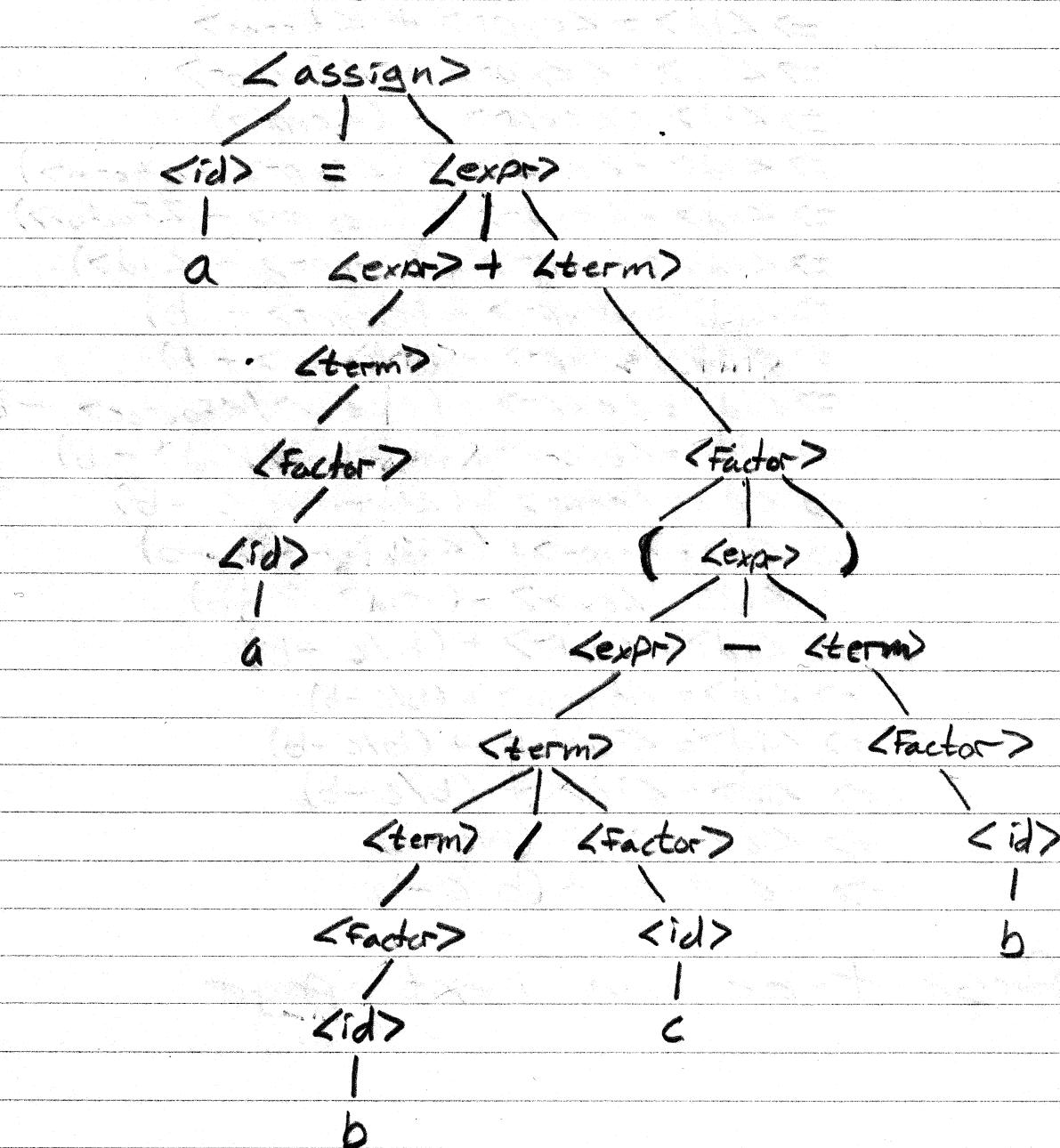
$$\Rightarrow \langle \text{id} \rangle = \langle \text{id} \rangle + (b / (-b))$$

$$\Rightarrow \langle \text{id} \rangle = a + (b / (-b))$$

$$\Rightarrow a = a + (b / (-b))$$

Parse tree on next page

2 a) continued, Parse Tree of Rightmost Derivation
for $a = a + (b/c - b)$



2 b) $\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

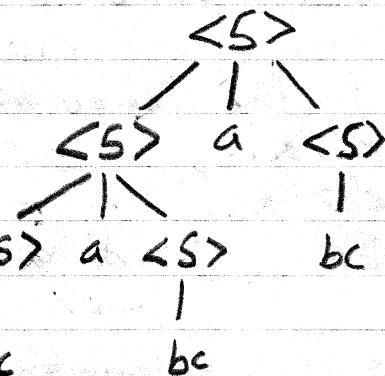
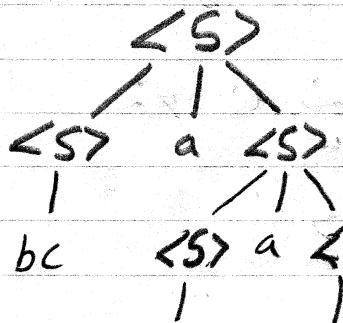
$\langle \text{id} \rangle \rightarrow a \mid b \mid c$

$\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle \mid \langle \text{expr} \rangle - \langle \text{term} \rangle \mid \langle \text{term} \rangle$

$\langle \text{term} \rangle \rightarrow \langle \text{term} \rangle * \langle \text{factor} \rangle \mid \langle \text{term} \rangle / \langle \text{factor} \rangle \mid \langle \text{factor} \rangle$

$\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle) \mid \langle \text{id} \rangle \mid -\langle \text{id} \rangle$

c) $\langle S \rangle \rightarrow \langle S \rangle a \langle S \rangle \mid bc$



Since the grammar has 2 distinct parse trees, we have proven the grammar is ambiguous.

d) Zero or more 0's followed by a_j , followed by zero or more 1's followed by b_j , followed by zero or more 2's followed by c_j .

e) $\langle S \rangle \rightarrow F \langle S \rangle \mid 0 \langle S \rangle \mid r \langle S \rangle \mid F \mid 0 \mid r$

Q F) $\langle \text{Var} \rangle \rightarrow \text{var} \langle \text{declarations} \rangle$

$\langle \text{declarations} \rangle \rightarrow \langle \text{type-decl} \rangle \{ \langle \text{type-decl} \rangle \}$

$\langle \text{type-decl} \rangle \rightarrow \langle \text{name-list} \rangle : \langle \text{type} \rangle ;$

$\langle \text{name-list} \rangle \rightarrow \langle \text{Name} \rangle \{ , \langle \text{Name} \rangle \}$

$\langle \text{Name} \rangle \rightarrow \langle \text{letter} \rangle \{ \langle \text{letter} \rangle | \langle \text{digit} \rangle \}$

$\langle \text{letter} \rangle \rightarrow a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z$

$\langle \text{digit} \rangle \rightarrow 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

$\langle \text{type} \rangle \rightarrow \text{int}128 | \text{uint}32 | \text{int}64 | \text{Float} | \text{long double}$

g) i) abba is in the language as shown by:

$\langle S \rangle \Rightarrow a \langle S \rangle a$

$\Rightarrow a \langle B \rangle ba$

$\Rightarrow abba \quad abba = abba$

ii) accaa is in the language as shown by:

$\langle S \rangle \Rightarrow a \langle S \rangle a$

$\Rightarrow a \langle A \rangle a$

$\Rightarrow ac \langle A \rangle a$

$\Rightarrow acc \langle A \rangle a$

$\Rightarrow acca a$

$accaa = accaa$

iii) accca is not in the language as shown by:

$\langle S \rangle \Rightarrow a \langle S \rangle a$

$\Rightarrow a \langle A \rangle a$

$\Rightarrow ac \langle A \rangle a$

$\Rightarrow acc \langle A \rangle a$

$\Rightarrow accc \langle A \rangle a$

$\Rightarrow accca a$

$acccaa \neq accca$

2 g) iv) $abbcccaba$ is not in the language as shown by:

$$\langle S \rangle \Rightarrow a \langle S \rangle a$$

$$\Rightarrow a \langle B \rangle ba$$

$$\Rightarrow ab \langle A \rangle ba$$

$$\Rightarrow abc \langle A \rangle ba$$

$$\Rightarrow abcc \langle A \rangle ba$$

$$\Rightarrow abccll \langle A \rangle ba$$

$$\Rightarrow abcccaba$$

NOTE: $\langle B \rangle b$ must be chosen

first otherwise b is not
accessible, which we need 3 of

$$abcccaba \neq abbcccaba$$

v) $acbccaa$ is not in the language as shown by:

$$\langle S \rangle \Rightarrow a \langle S \rangle a$$

$$\Rightarrow a \langle B \rangle ba$$

$$\Rightarrow ab \langle A \rangle ba$$

$$\Rightarrow abc \langle A \rangle ba$$

$$\Rightarrow abcc \langle A \rangle ba$$

$$\Rightarrow abccaba$$

NOTE: $\langle B \rangle b$ must be chosen

First otherwise b is not
accessible, which we need 1 of

$$abccaba \neq acbccaa$$

h) $\langle \text{case_stmt} \rangle \rightarrow \text{case } (\langle \text{expr} \rangle) \text{ of } \langle \text{case_list} \rangle \text{ default } \langle \text{statement} \rangle ;$

$$\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle \{ (+|-) \langle \text{term} \rangle \}$$

$$\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \{ (*|/) \langle \text{factor} \rangle \}$$

$$\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle) | \langle \text{id} \rangle$$

$$\langle \text{id} \rangle \rightarrow \langle \text{letter} \rangle \{ \langle \text{letter} \rangle | \langle \text{digit} \rangle \}$$

$$\langle \text{letter} \rangle \rightarrow a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z$$

$$\langle \text{digit} \rangle \rightarrow 0|1|2|3|4|5|6|7|8|9$$

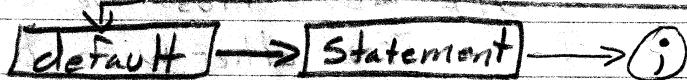
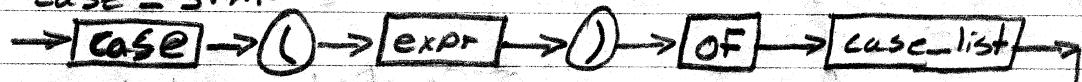
$$\langle \text{case_list} \rangle \rightarrow \langle \text{case} \rangle : \langle \text{statement} \rangle ; \{ \langle \text{case} \rangle : \langle \text{statement} \rangle ; \}$$

$$\langle \text{case} \rangle \rightarrow \langle \text{id} \rangle | \langle \text{number} \rangle | \langle \text{range} \rangle \{ , \langle \text{id} \rangle | , \langle \text{number} \rangle | , \langle \text{range} \rangle \}$$

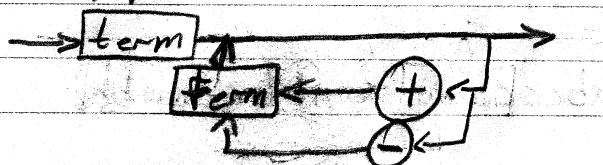
$$\langle \text{number} \rangle \rightarrow \langle \text{digit} \rangle \{ \langle \text{digit} \rangle \}$$

$$\langle \text{range} \rangle \rightarrow \langle \text{number} \rangle .. \langle \text{number} \rangle$$

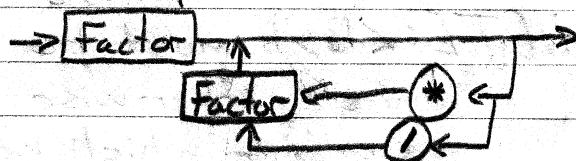
2. h) ii) case-stmt



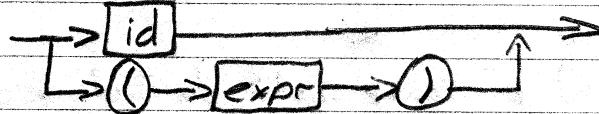
expr



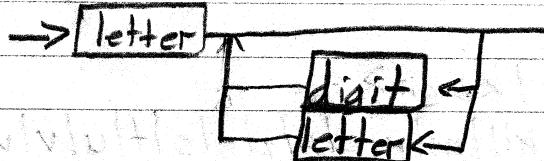
term



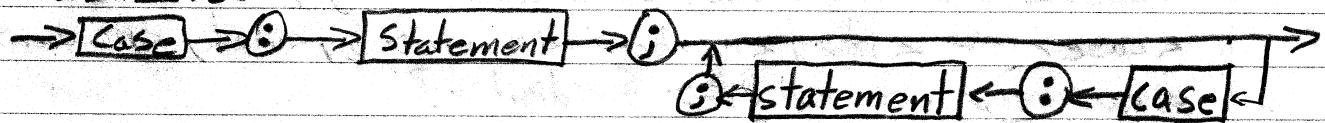
factor



id

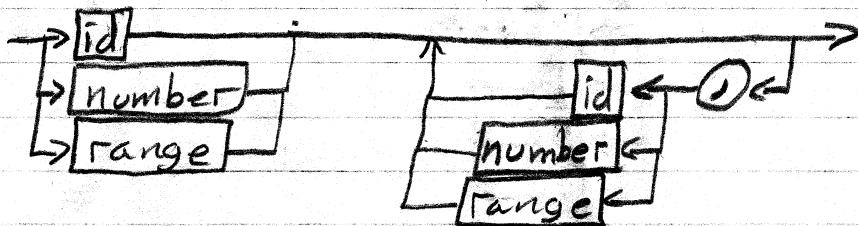


Case_list



2 h) ii) cont

Case



Number



Range



2 h) ii) cont.

