

Syntax (continued)

Flexible & Ambiguous Grammars

- A grammar is ambiguous if its language contains at least one string with two or more distinct parse trees.
- Sometime we might want to use an ambiguous grammar to simplify the number of rules required.
- Ambiguities in grammars are generally resolved using additional rules.
 - For example, if we have a table of precedence and a default left-to-right ordering of operators of equal precedence, then we can resolve any ambiguities that arise.
- Dangling else
 - Another common ambiguity in language syntax.
 - When an if statement is contained inside an if statement, which if statement does a subsequent else belong to?

$ifStatement \rightarrow if (Expression)Statement \mid if (Expression)Statement \text{ else } Statement$

$Statement \rightarrow Assignment \mid ifStatement \mid Block$

$Block \rightarrow \{Statement\} \{Statement\}$

- Consider the following code snippet.
- The second else could match with either if condition. Only by inserting brackets could the interpretation be unambiguous.
- Solution of C
 - The arbitrary rule, included in the description of the language, that an else clause is associated with the textually nearest if statement in any ambiguous case. The actual output is “there”.

```
#include <stdio.h>

int main (int args, char *argv[]) {
    int a = -1;
    int b = 1;

    if (a < 0)
        if (b < 0)
            printf("here\n");
    else
        printf("there\n");
}
```

- Solution of Java
 - Clearly defined in grammar to address the unambiguous.
 - It is not permitted that an if statement without an else clause as the single statement after an if.
 - The following code snippet, for example, will not do what the tabbing implies.

```
public class Ambiguity {  
    public static void main (String args[]) {  
        int a = -1;  
        int b = 1;  
  
        if (a < 0)  
            if (b < 0)  
                System.out.println("here");  
        else  
            System.out.println("there");  
    }  
}
```

- The actual output is “there”, since Java consider the else branch belongs to the second if statement.
 - Solution of Python
 - Require nested if statements to be indented. The actual output is “”.
-