

UNIVERSITY OF NEVADA, RENO



CS 326 — PROGRAMMING LANGUAGES

---

# Assignment #4

---

Joshua GLEASON

*Instructor:*  
Dr. Mircea NICOLESCU

October 7, 2010

1. The advantage of explicit terminators such as `end if`, is that the end of the statement is clear. Not having an explicit terminator may lead to confusion. By making only one expression allowed following each part of the statement reduces this confusion in implicitly terminating languages like C, where code is placed into a block beginning and ending with brackets. By introducing explicit terminators, multiple statements can be placed into the an if-then-else, making the use of code blocks or parenthesis unnecessary.

2.

$$\begin{aligned} \text{expr} &\rightarrow \text{nr} \mid \text{id} \mid \text{op expr expr} \\ \text{op} &\rightarrow * \mid + \end{aligned}$$

This grammar is not ambiguous because for every expression there is exactly one parse tree possible for every expression in the grammar.

3. The code does not produce short-circuit behavior because the arguments of the function had to be calculated before the function body is entered. If normal-order evaluation were used, then this function would properly compute the short-circuit and operator. This is because in normal-order evaluation the arguments aren't evaluated until they are needed. In this function `b` would not be evaluated if `a` were true because the function would never need the value of `b`.

4. (a) `do`

```
s;  
while(c);
```

```
// goes to
```

```
s;  
while(c)  
s;
```

- (b) `while(c)`

```
s;
```

```
// can be simulated by
```

```
do  
  if (c)  
    s;  
while(c);
```

```

(c) for(s1;c;s2)
    s;

// can be simulated using a do

s1;
do {
    if (c)
    {
        s;
        s2;
    }
} while(c);

```

```

5. (define (sum L)
    (define (sum-helper L a)
        (if (null? L)
            a
            (sum-helper (cdr L) (+ a (car L)))))
    (sum-helper L 0))

```