

CS-257L

Nonimperative Programming: Scheme!

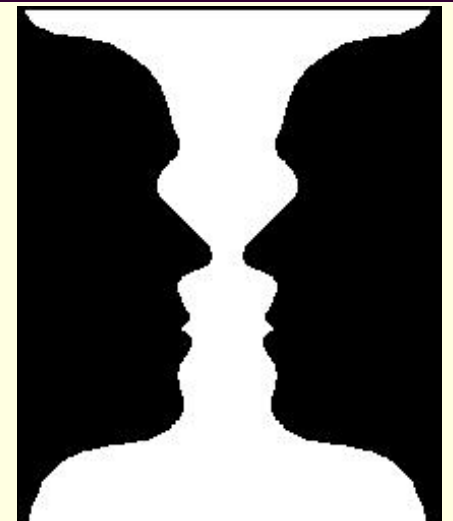
Instructor:

Joel Castellanos

e-mail: joel@unm.edu

Web: <http://cs.unm.edu/~joel/>

Office: Farris Engineering
Center (FEC) room 321



Major Issues from Roulette Assignment

1. Test Your Code!!!

```
((< (random) (/ 18 38)) ;I win!!!
```

verses

```
((< (/ 18 38) (random)) ;I win!!!
```

2. Efficiency:

- My version, on my laptop runs 50 million bets in about 90 seconds.
- Most student versions take more than 17 minutes to run 50 million bets on the same machine.

Quiz 5-Part A – What is the output?

```
(let ((a 7) (b 13))
  (let ((b 3) (c (+ a b)))
    (let ((b 5))
      (cons a (cons b (cons c ' ())))))
  )
)
```

Quiz 5-Part B

Exercises 5.9 and 5.10 involve functions of polynomials such as `p+` which adds to polynomials and has the form:

`(p+ poly1 poly2)`

What is the format for calling `p+` with the following two polynomials?

$$5x^7 + 9x + 4$$

$$x^4 + 3x^2$$

Class Friday 3/14 is Optional

- There will not be a quiz.
- No new material covered.
- Expect it to be sparsely populated.
- Question / Answer / Chat format.

- Next week is Spring Break.

Homework: Due Wednesday March 26

```
(define DCFL-builder
  (lambda
    ( axiom ; a list of atoms
      rules ; a list of lists of atoms.
      n      ; number of generations
    )
```

Assuming you do not want to do homework during spring break, please start this tonight or Thursday night. Then, if you have questions, you can get them answered Friday.

Grading of DCFL-builder

- Ok to copy and/or change, without reference, my code from DCFL-checker, any of my code on the class web site, and any code from either textbook.
- Always code with comments, but the grader will not be looking at your code.
- The assignment is worth 15 points.
- You score one point for each test you pass.
- Efficiency not graded on this project, however, you will want to reuse your code on the next project (when you will apply a graphical interpretation to the return list). On that project, the lists will be millions of elements long and efficacy will become important.

Test format

The test program, "*test.scm*" on the class website will be used for testing.

The tests given on the following slides are very similar to what you will be given, but not exactly what you will be given. For example, (b) or (c) below might be used in place of (a).

a) `(DCFL-builder '(a b c)
 '((a a a) (b b b) (b c c)) '3)`

b) `(DCFL-builder '(x y z)
 '((x x x) (y y y) (y z z)) '3)`

c) `(DCFL-builder '(x y z)
 '((x x x) (y y y y) (y z z)) '3)`

Test Case 1

```
(test msg
  (DCFL-builder
    '(a b c)
    '((a a a) (b b b) (b c c))
    '3
  )
  '(#f "Nondeterministic CFL")
)
```

Test Case 2

```
(test msg
  (DCFL-builder
    '(a)
    '((a b c x) (b a x) (c a y) (b a a))
    '3
  )
  '(#f "Nondeterministic CFL")
)
```

Test Case 3

```
(test msg
  (DCFL-builder
    '(a b (c))
    '((a a a))
    '3
  )
  '(#f "axiom must be a list of atoms.")
)
```

Test Case 4

```
(test msg
  (DCFL-builder
    '(a b c)
    '((a a a) b (c c c))
    '3
  )
  '(#f "rules must be a list of a list
      of atoms."))
)
```

Test Case 5

```
(test msg
  (DCFL-builder
    '(a b c)
    '((a a a) (b b b) (c (c c)))
    '3
  )
  '(#f "rules must be a list of a list
      of atoms."))
)
```

Test Case 6

```
(test msg
  (DCFL-builder
    '(a b c)
    '(() (b b b) (c c c))
    '3
  )
  '(#f "rules must be a list of a list
      of atoms."))
)
```

Test Case 7

```
(test msg
  (DCFL-builder
    '(a b c)
    '(() (b b b) (c c c))
    '-3
  )
  '(#f "n must be a non-negative integer.")
)
```

Test Case 8

```
(test msg
  (DCFL-builder
    '(a b c)
    '(() (b b b) (c c c))
    '2.5
  )
  '(#f "n must be a non-negative integer.")
)
```

Test Case 9

```
(test msg
  (DCFL-builder
    '(f)
    '((f f - h) (h f + h))
    '0
  )
  '(#t (f)))
)
```

Test Case 10

```
(test msg
  (DCFL-builder
    '(f)
    '((f f - h) (h f + h))
    '1
  )
  '(#t (f - h))
)
```

Test Case 11

```
(test msg
  (DCFL-builder
    '(f)
    '((f f - h) (h f + h))
    '2
  )
  '(#t (f - h - f + h))
)
```

Test Case 12

```
(test msg
  (DCFL-builder
    '(f)
    '((f f - h) (h f + h))
    '3
  )
  '(#t (f - h - f + h - f - h + f + h))
)
```

Test Case 13

```
(test msg (DCFL-builder
```

```
'(f)
```

```
'((f k < 5 + f > < 7 - f > - k < 4 + f > < 6 - f >  
- k < 3 + f > < 5 - f > - k f )) '2)
```

```
'(#t (k < 5 + k < 5 + f > < 7 - f > - k < 4 + f >  
< 6 - f > - k < 3 + f > < 5 - f > - k f >  
< 7 - k < 5 + f > < 7 - f > - k < 4 + f >  
< 6 - f > - k < 3 + f > < 5 - f > - k f > - k  
< 4 + k < 5 + f > < 7 - f > - k < 4 + f >  
< 6 - f > - k < 3 + f > < 5 - f > - k f >  
< 6 - k < 5 + f > < 7 - f > - k < 4 + f >  
< 6 - f > - k < 3 + f > < 5 - f > - k f > - k  
< 3 + k < 5 + f > < 7 - f > - k < 4 + f >  
< 6 - f > - k < 3 + f > < 5 - f > - k f >  
< 5 - k < 5 + f > < 7 - f > - k < 4 + f >  
< 6 - f > - k < 3 + f > < 5 - f > - k f > -k k  
< 5 + f > < 7 - f > - k < 4 + f > < 6 - f > -k  
< 3 + f > < 5 - f > - k f)))
```

Test Case 14

```
(test msg
  (DCFL-builder
    '(d x)
    '((x d f - < < x > + x > + f < + a f x > - x)
      (f d f f)
      (a)
      (d)
    ) '1)

  '(#t (d f - < < x > + x >
        + f < + a f x > - x)))

)
```

Test Case 15

```
(test msg
```

```
  (DCFL-builder
```

```
    '(d x)
```

```
    '((x d f - < < x > + x > + f < + a f x > - x)
```

```
      (f d f f)
```

```
      (a)
```

```
      (d)
```

```
    )
```

```
  '2)
```

```
'(#t (d f f - < < d f - < < x > + x > + f < + a f x >
- x > + d f - < < x > + x > + f < + a f x > - x > + d f
f < + d f f d f - < < x > + x > + f < + a f x > - x >
-d f - < < x > + x > + f < + a f x > - x))
```

```
)
```

MU-Puzzle from Gödel, Escher Bach

- Can you derive **MU** in the system below?
- **Alphabet:** {**M U I**}
- **Axiom:** **MI**
- **Rule 1:** If you have a theorem whose last letter is **I**, then you may add **U** to the end. Symbolically, $xI \Rightarrow xIU$, where \Rightarrow stands for implies.
- **Rule 2:** If **Mx** is a theorem, so is **Mxx**: $Mx \Rightarrow Mxx$.
- **Rule 3:** In any theorem, **III** can be replaced with **U**:
 $xIIIIy \Rightarrow xUy$.
- **Rule 4:** **UU** can be dropped from any theorem: $xUUy \Rightarrow xy$.

Inside and Outside of the System

MUI-System:

- Inside:
 - deriving theorems
- Outside
 - All theorems begin with the letter M.

