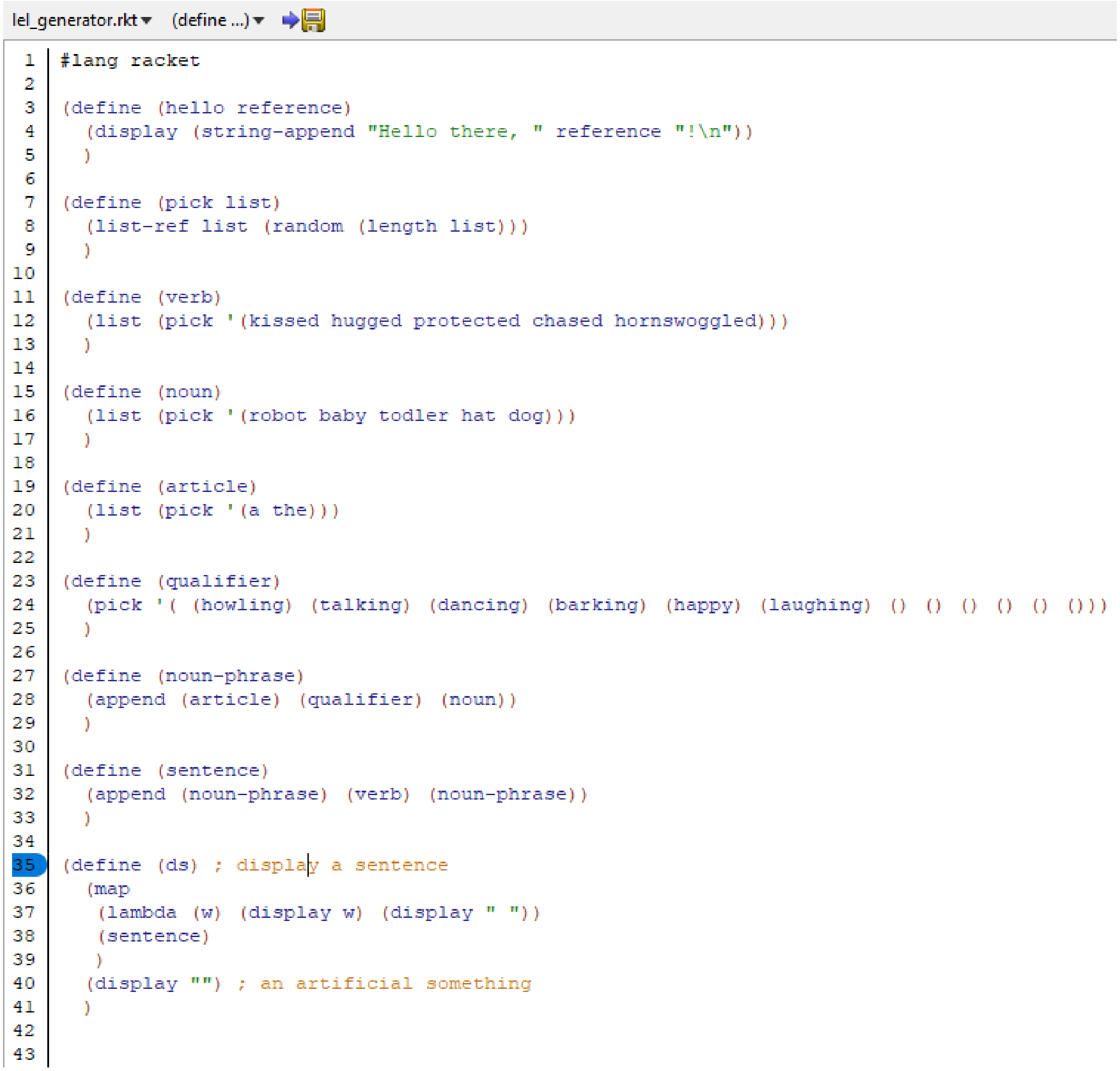
Title: Racket Assignment #1: Getting Acquainted with Racket/DrRacket + LEL Sentence Generation

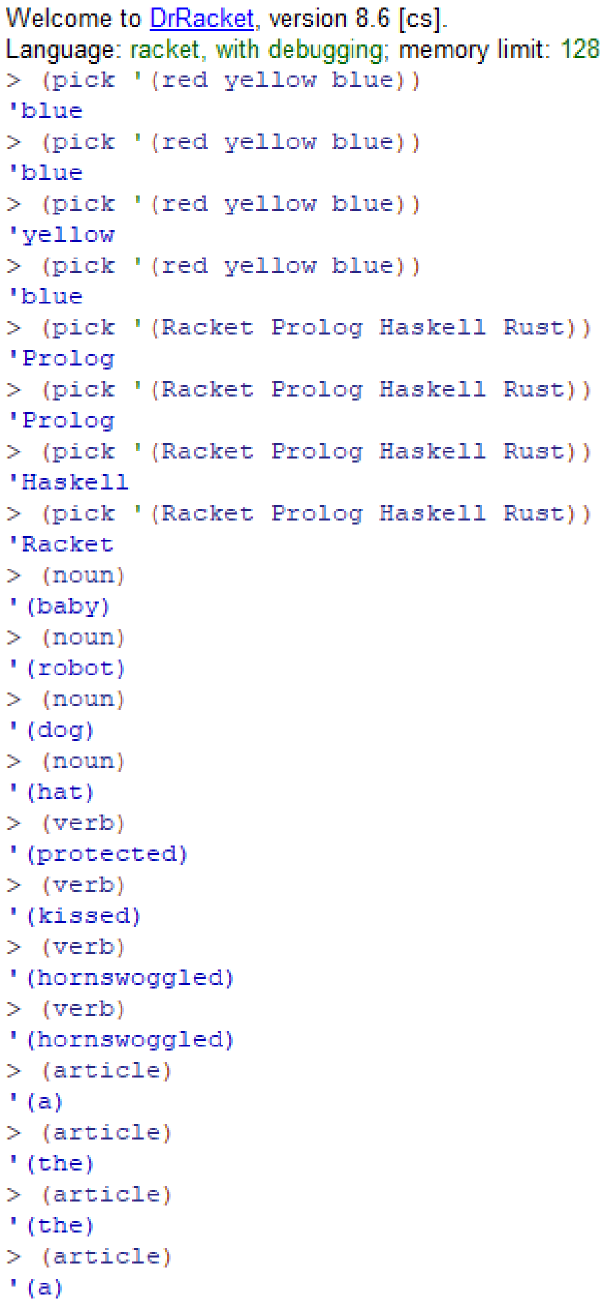
Abstract

In this assignment I familiarized myself with the basics of Racket by copying and running Professor Graci's code.

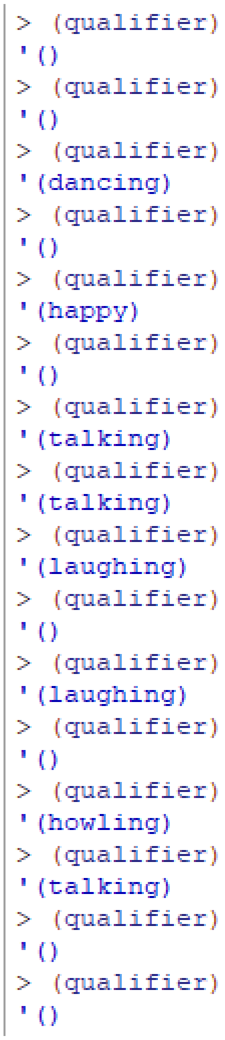
Code, followed by Demo



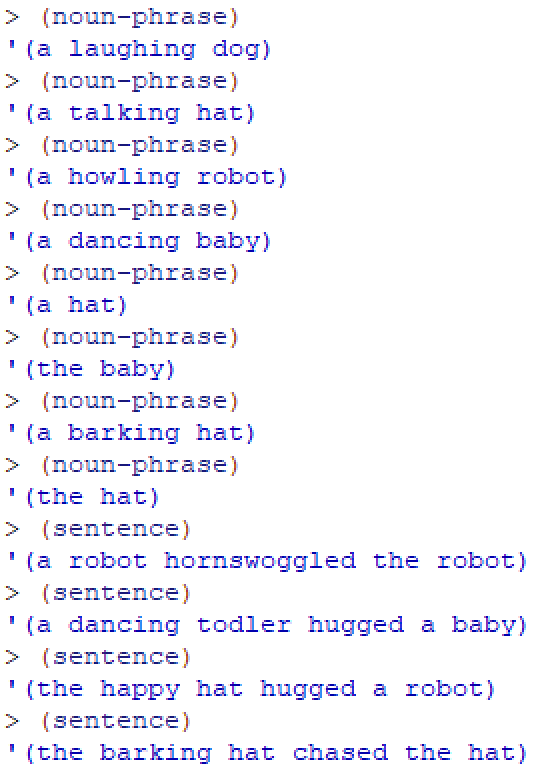
```
#lang racket

(define (hello reference)
  (display (string-append "Hello there, " reference "!\n"))
  )

(define (pick list)
  (list-ref list (random (length list)))
  )

(define (verb)
  (list (pick '(kissed hugged protected chased hornswoggled)))
  )

(define (noun)
  (list (pick '(robot baby todler hat dog)))
  )

(define (article)
  (list (pick '(a the)))
  )

(define (qualifier)
  (pick '( (howling) (talking) (dancing) (barking) (happy) (laughing) () () () () () ()))
  )

(define (noun-phrase)
  (append (article) (qualifier) (noun))
  )

(define (sentence)
  (append (noun-phrase) (verb) (noun-phrase))
  )

(define (ds) ; display a sentence
  (map
   (lambda (w) (display w) (display " "))
   (sentence)
   )
  (display "") ; an artificial something
  )
```



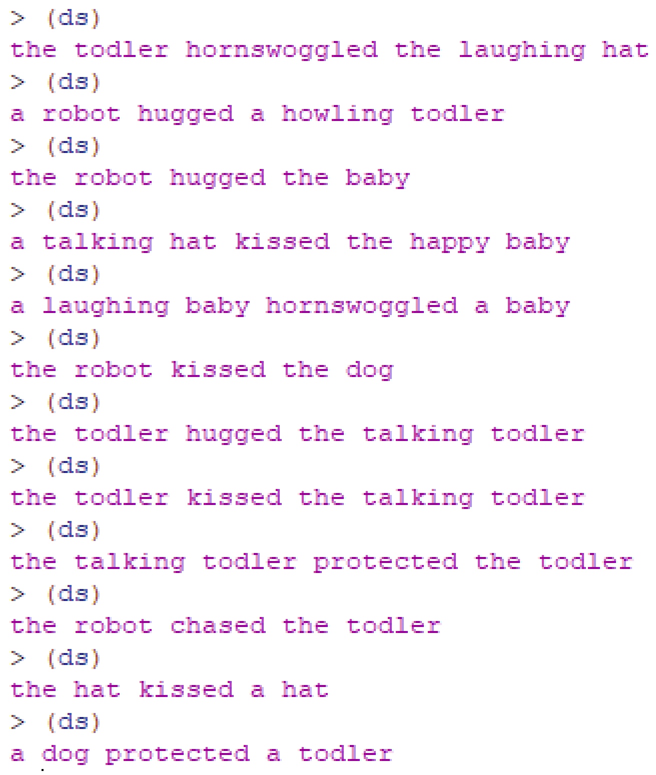
```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (pick '(red yellow blue))
'blue
> (pick '(red yellow blue))
'blue
> (pick '(red yellow blue))
'yellow
> (pick '(red yellow blue))
'blue
> (pick '(Racket Prolog Haskell Rust))
'Prolog
> (pick '(Racket Prolog Haskell Rust))
'Prolog
> (pick '(Racket Prolog Haskell Rust))
'Haskell
> (pick '(Racket Prolog Haskell Rust))
'Racket
> (noun)
'(baby)
> (noun)
'(robot)
> (noun)
'(dog)
> (noun)
'(hat)
> (verb)
'(protected)
> (verb)
'(kissed)
> (verb)
'(hornswoggled)
> (verb)
'(hornswoggled)
> (article)
'(a)
> (article)
'(the)
> (article)
'(the)
> (article)
'(a)
```



```
> (qualifier)
'()
> (qualifier)
'()
> (qualifier)
'(dancing)
> (qualifier)
'()
> (qualifier)
'(happy)
> (qualifier)
'()
> (qualifier)
'(talking)
> (qualifier)
'(talking)
> (qualifier)
'(laughing)
> (qualifier)
'()
> (qualifier)
'(laughing)
> (qualifier)
'()
> (qualifier)
'(howling)
> (qualifier)
'(talking)
> (qualifier)
'()
> (qualifier)
'()
```



```
> (noun-phrase)
'(a laughing dog)
> (noun-phrase)
'(a talking hat)
> (noun-phrase)
'(a howling robot)
> (noun-phrase)
'(a dancing baby)
> (noun-phrase)
'(a hat)
> (noun-phrase)
'(the baby)
> (noun-phrase)
'(a barking hat)
> (noun-phrase)
'(the hat)
> (sentence)
'(a robot hornswoggled the robot)
> (sentence)
'(a dancing todler hugged a baby)
> (sentence)
'(the happy hat hugged a robot)
> (sentence)
'(the barking hat chased the hat)
```



```
> (ds)
the todler hornswoggled the laughing hat
> (ds)
a robot hugged a howling todler
> (ds)
the robot hugged the baby
> (ds)
a talking hat kissed the happy baby
> (ds)
a laughing baby hornswoggled a baby
> (ds)
the robot kissed the dog
> (ds)
the todler hugged the talking todler
> (ds)
the todler kissed the talking todler
> (ds)
the talking todler protected the todler
> (ds)
the robot chased the todler
> (ds)
the hat kissed a hat
> (ds)
a dog protected a todler
```