

Your Name: \_\_\_\_\_ Your EID: \_\_\_\_\_

Circle Your Discussion Section:

- 54075: Ian Wehrman, Friday, 9:00 – 10:00a, ENS 109
- 54080: Ian Wehrman, Friday, 10:00 – 11:00a, GSB 2.122
- 54085: David Rager, Friday, 10:00 – 11:00a, JES A218A
- 54088: Behnam Robatmili, Friday, 12:00 – 1:00p, RLM 5.122
- 54090: Behnam Robatmili, Friday, 1:00 – 2:00p, CBA 4.344
- 54095: Nathan Wetzler, Friday, 1:00 – 2:00p, JES A209A

### Midterm Exam 1

CS313K Logic, Sets, and Functions – Spring, 2009

#### Instructions

Write your name and EID above and circle the unique ID of your discussion section! Write your answers in the space provided. There are 20 questions. Each question is worth 5 points. Partial credit may be given so do your best on each question. You have until 5:00 pm.

You may refer to the course notes (the red book) during the exam. You may refer to your own notes if they are on paper. No computers are allowed. No talking is allowed. No cellphones. Remove hats, baseball caps, etc.

You are free to use any function whose definition is exhibited in the red book. You can use any primitive function used in the red book, e.g., `if`, `equal`, `car`, `cdr`, etc. You can also use any function whose `defun` is shown explicitly in the red book. You can use any function whose `defun` is shown explicitly in this exam, either in the part I handed out or in your previous answers.

The last section of the exam has no questions. It just lists some possibly useful `defuns` we've seen.

But if you want to use any other function – even functions I defined in class or you defined in your homeworks or that I mentioned (but did not define) in the red book – you must write down their `defuns` with your answer. Use as many functions as you want, but make sure we know what their definitions are!

## Some Possibly Useful Definitions

You may not need these functions in your answers, but they may suggest useful patterns for definitions.

```
(defun app (x y)
  (if (endp x)
      y
      (cons (car x)
            (app (cdr x) y))))

(defun how-many (e x)
  (if (endp x)
      0
      (if (equal e (car x))
          (+ 1 (how-many e (cdr x)))
          (how-many e (cdr x)))))

(defun mem (e x)
  (if (endp x)
      nil
      (if (equal e (car x))
          t
          (mem e (cdr x)))))
```