

CMPS 461: Programming Language Concepts

Midterm 1 Practice Questions

Problem 1 λ -calculus.

- What are the free variables in the λ -term $(\lambda x . (\lambda y . y z) y) x$.
- Fully evaluate $(\lambda x . (\lambda y . ((\lambda y . y) x))) z$.
- Define a function `doubleChurch`, which takes a Church Numeral and doubles it. Try to build it from scratch, without using the encoding of $+$ or $*$ as defined in lecture.

Problem 2 Higher-order functions and currying.

- What is the curried form of $\lambda x y . (+ x y)$.
- Explain in one sentence, what's the result of $(\lambda x y . (+ x y)) 1$.

Problem 3 Scheme.

- Define a recursive Scheme function that takes a list of numbers and returns the product of all the list elements. Given the empty list, the product should be 1.
- Define the same functions using "fold".

Problem 4 Scheme.

What do the following functions do?

```
(define (blue p l) (= (length l) (length (filter p l))))  
(define (renmap f x ys) (map (lambda (y) (f x y)) ys))  
(define (binmap f xs ys) (map (lambda (x) (renmap f x xs)) ys))
```

Problem 5 Regular Expressions.

- Define a regular expression for all strings of odd length, over the alphabet of $\{0\}$.
- Define a regular expression for identifiers over the alphabet of $\{A,B,C,a,b,c,0,1,2,3,4,5,6,7,8,9\}$, such that an identifier must begin with an alphabetic character and must contain at least one numeric character.

Problem 6 Context free grammar.

Consider the following context free grammar:

$$P ::= id \mid P \vee P \mid P \wedge P \mid \neg P \mid (P)$$
$$id ::= p \mid q \mid r \mid s$$

where the set of terminals symbols is $\{p, q, r, s, \vee, \wedge, \neg, (,)\}$ and the start symbol is P .

- Give the leftmost derivations for the following expression:

$$\bullet \neg(p \wedge q)$$

- Show that this language is ambiguous.

Problem 7 Context Free Grammar.

For each of the following grammars state whether it is ambiguous or unambiguous. If it is ambiguous give an equivalent unambiguous grammar. If it is unambiguous, state all the precedences and associativities enforced by the grammar. Assume Id generates identifiers.

a) $E ::= E + F \mid E - F \mid F$
 $F ::= -F \mid Id \mid (E)$

b) $E ::= E \vee F \mid F$
 $F ::= G \cap F \mid G \cup F \mid G$
 $G ::= G \wedge H \mid H$
 $H ::= Id \mid (E)$

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder