

CMSC330 Spring 2010 Quiz #2

Name _____

Discussion Time (circle one): 9am 10am 11am 12pm 1pm 2pm

Instructions

- You have 20 minutes for this quiz.
- This is a closed book exam. No notes or other aids are allowed.
- For partial credit, show all of your work and clearly indicate your answers.
- Write neatly. Credit cannot be given for illegible answers.

1. (6 pts) OCaml Types and Type Inference
 - a. (2 pts) Give the type of the following OCaml expression
`fun x -> (x,2)`

Type =

- b. (2 pts) Write an OCaml expression with the following type
`int -> (float * int list)`

Code =

- c. (2 pts) Give the value of the following OCaml expression. If an error exists, describe the error.
`(fun z -> fun y -> z - y) 5 3`

Value =

let rec map f l = match l with [] -> [] l (h::t) -> (f h)::(map f t)	let rec fold f a l = match l with [] -> a l (h::t) -> fold f (f a h) t
--	--

2. (8 pts) OCaml Programming

Using the following code for either map/fold and an anonymous function, write a function `getFirsts` which given a list of pairs, returns a list of the 1st members of each pair as a list (in original or reverse order). Partial credit given for solutions which do not use map/fold.

Example: `getFirsts [(1,2);(3,4);(3,5)] = [1;3;3] OR [3;3;1]`
`getFirsts [("a","x");("b","y");("c","z")] = ["a";"b";"c"] OR ["c";"b";"a"]`

3. (6 pts) Context free grammars

Consider the following grammar: $S \rightarrow aSaS \mid \epsilon$ (* epsilon *)

a. (2 pts) Describe the set of strings generated by the grammar.

b. (4 pts) Is the grammar ambiguous? Show proof if possible.