

CS603

Programming Languages

Lecture 20

Spring 2004

University of Alabama

Overview

- Polymorphic type systems and Typed μ ML
 - Problems with Typed Impcore
 - Typed μ ML

Problems with Typed Impcore

- Too complicated: multiple type environment
- Lacks power: user defined functions can only operate on values of a single type—
monomorphic
 - Example: can't write general-purpose array reversal
- Every new type constructor requires change to type system

Polymorphism

- polymorphic type system—type system where generic type constructors can be used to create new type without changing the type system -or-
- polymorphic function—a function that has one or more arguments whose type may differ based upon the caller and which is still type sound

Typed μ ML Syntax

```
toplevel ::= exp
           | (use file-name)
           | (val variable-name exp)
           | (val-rec variable-name exp)
           | (define type function-name (formals) exp)
exp ::= literal
        | variable-name
        | (if exp exp exp)
        | (begin {exp})
        | (exp {exp})
        | (let-keyword ({(variable-name exp)}) exp)
        | (lambda ({formals}) exp)
        | primitive
formals ::= {variable-name}
S-exp ::= literal | symbol-name | ({S-exp}) | 'S-exp'
primitive ::= + | - | * | / | = | < | > | print | car | ..
```