Project: The small programming language

Syntax:

Program ::= Declarations 'in' Expression

Declarations ::= Declaration [Declarations]

Declaration ::= 'function' Name '(' [Parameters] ')' [':' Type] '=' Expression
| 'var' Name [':' Type] '=' Expression

Parameters ::= Name [':' Type] [', Name [':' Type]]

Type ::= 'Unit' | 'Bool' | 'Int' | 'Char' | 'String' | 'Record' [Name [':' Type] [', Name [':' Type]]] 'end'

Expressions ::= Expression [';' Expressions]

ExpressionList ::= Expression [', ExpressionList]

Expression ::= Name ':=' Expression
| 'if' Expression 'then' Expression 'else' Expression 'fi'
| 'while' Expression 'do' Expression
| 'begin' Expressions 'end'
| 'choice' '(' ExpressionList ')' 'end'
| 'return' Expression
| 'input' Name String
| 'output' Expression
| Value

Value ::= 'not' Value | EqValue

EqValue ::= EqValue ['==' | '!='] RelValue | RelValue

RelValue ::= RelValue ['and' | 'or'] AddValue | AddValue

AddValue ::= AddValue ['+' | '-'] MulValue | MulValue

MulValue ::= MulValue ['*' | '/'] PrimValue | PrimValue

PrimValue ::= 'skip' | Name | 'TRUE' | 'FALSE' | Character | Number | String | Name ['(' [ExpressionList] ')' | 'record' [Name '=' Expression [', Name '=' Expression]] 'end' | PrimValue '.' Name
Problem 1

Study the given parser (modules Scanner.hs, Abstract.hs, and Parser.hs). Do you understand everything? Do you have any questions concerning ”monadic parsing”? Do you understand the reason for the decomposition of the parser for record projection? Can you add new language constructs to the given parser? Do you see that the parser monad is just another instance of the state monad?

Problem 2

1. Define a denotational semantics specification without input, output, and choice,
2. Construct an interpreter for the given programming language.
3. Implement choice,
4. Implement input and output,
5. (Optionally) Implement the check of the static (declaration) semantics.

Submission deadline: Tuesday, December 11, 2001, 11am

Submission procedure: Submissions should be made electronically. Send your solution to pramod@cs.iastate.edu with a CC to lumpe@cs.iastate.edu.

On the department’s computer systems (HP’s), use the command hugs to start the Haskell interpreter.