

CS 603: Programming Language Organization

Lecture 7

Spring 2005

Department of Computer Science

University of Alabama

Joel Jones

©2005 Joel Jones

Outline

- Questions
- Impcore Interpreter
- Reading for next time

Impcore Interpreter: Environments

- Environments map names to values (or types), therefore interpreter needs to represent names, values, and environments
- Names are an abstract data type, `Name`, which support comparison and conversion to and from `char*`
 - `nametostr` converts a `Name` to `char*`
 - `strtoname` converts a `char*` to a `Name` and acts as a constructor

Impcore Interpreter: Environments (cont.)

- Uses one C type for `Value` and another for `Fun`
- Constructors
 - `mkValenv`
 - `mkFunenv`
- Accessors
 - `fetchval`, `isvalfound`, `bindval`
 - `fetchfun`, `isfunbound`, `bindfun`

Impcore Interpreter: Evaluation

- Heart of the interpreter—`topeval` and `eval`
- Arguments follow form of operational semantics

Pair Up: • What was the form of the $\{\downarrow\}$ relation?

- What did it signify?

```
• Value eval(Exp *e, Valenv
*globals, Funenv *functions,
Valenv *formals)
```

Impcore Interpreter: Evaluation (cont.)

- Evaluation of tree structure
 - Dispatch on type of node at root
 - Recursively execute children
 - Execute tree

Pair Up:

- What field of the AST representation do we dispatch on?
- How do you dispatch on type in a non-OO language like C?

Impcore Interpreter: Evaluation (cont.)

- For each AST type, there may be multiple applicable judgments in the operational semantics
 - VAR has `FORMALVAR` and `GLOBALVAR`

Pair Up:

- Write the mappings from tags to judgment names for the rest of the `Exp` productions

Impcore Interpreter: Evaluation (cont.)

- The form of eval is then:

```
switch (e->typ) {
```

```
case TAG:
```

```
    evaluate rhs if shared premise exists
```

```
    choose and execute appropriate judgment
```

```
    ...
```

```
}
```


Impcore Interpreter: Evaluation (cont.)

- What is done in implementing each of the productions?