Names, Binding, and Lifetime

Prof. Evan Chang
Meeting 5, CSCI 3155, Fall 2009

Announcements

- Assignment 2 due Thu at 11:55pm
  - Submit individually
  - PL-Detective User ID sent out by e-mail

Review

Syntax: One-Slide Summary

- **Concrete syntax** is the surface level of a language (think, strings)
- **Abstract syntax** is the deep structure of a language (think, trees/terms)
- Parsers convert concrete syntax into abstract syntax and have to deal with ambiguity
- Precedence and associativity are some ways to deal with ambiguity

What is an ambiguous grammar?

One w/ multiple parse trees for a given string

What is abstract syntax (vs. concrete syntax)?

<table>
<thead>
<tr>
<th>Concrete</th>
<th>Abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>strings</td>
<td>terms</td>
</tr>
</tbody>
</table>
| IMPUTATION C's expressions vs. FUNCTIONAL MATH: 
  MATH: I, transformations of terms |
| (result of parsing) easy to read for washing |
End of Review
On to Names, Binding, and Lifetime

Names in PLs

- Used for?
  - Variables, functions, keywords, classes, modules
  - Declaters, in statements, expressions
  - Operators (and)

Variables

- Attributes (Sebesta)
  - Name
  - Type
  - Value
  - Address
  - Scope
  - Lifetime (how long it exists)

Variables: Names

- Write a program to figure out properties of names in MYSTERY
  - What properties?
    - Capitalization / case sensitivity
    - Characters (numbers, -, ?)
    - Length

Attributes (Sebesta)

- Name
- Address
- Type
- Value
- Lifetime
- Scope

Keywords vs. reserved words:
- Can reuse names
  - names depend on context

int if; x
int float; x
Variables: Address

- May change at run-time ("run-time name")
  - Example?
  ```c
  void f (int x) { int y; ... f(y); ... }
  ```

Aliasing

- What is aliasing?
  - Same memory location (address) having multiple names (variables)
- Where do they arise (language features)?
  - References, pointers

Aliasing

- Example in Java?
  ```java
  public class Int { public int val; }
  ```

Variables: Type, Value, Lifetime

- Type
  - `int, float` "some of contents" "possible values of contents"
- Value
  - `data/contents`
- Lifetime
  - `when it exists` affected by `storage binding`
Binding

- A mapping from a name to a value
- Name?
  - Variables
- Value?
  - contents/data/bits
  - address
  - type

Binding Time

- What is binding time?
  - Compile-time (type of var for lang)
  - Link-time (addresses for globals)
  - Design-time (static = before run-time and unchanging)

Example: Static and Dynamic Typing

- static: type bound to variable at compile-time
- dynamic: type bound to variable at run-time

Example: Static and Dynamic Typing

```
x = 3;
y = x + 1;
x = [7, 9, 2];
y = x + 1
```

Ok with static or dynamic typing?
Example: Static and Dynamic Typing

- Can we have
  \[ x = 3; x = [10.2, 5, "bob"]; \]

Example: Static and Dynamic Typing

- Static typing includes dynamic typing

- With this view, what is wrong with dynamic typing?

- Dynamic typing forces the overhead of tag checking always

For Next Time

- Reading
- Online discussion forum
  - \( \geq 1 \) substantive question, comment, or answer each week
- Homework assignment 2
  - Start early!