Meeting 14: Data Types and Polymorphism

Announcements

- HW4 due Friday 10/13
- Exercise 4 and review session Tue
- Midterm next Thu

Questions

1. And, or, not explicit
   - What's a P1?
   - Explode?
   - Optimizing with constants?

2. Type checking

3. When to create calls to runtime?

4. Lists/Dicts/arrays
3. \(a_1, a_1\) - registers for compare
4. specifically flattening compare

and, or, not in Python 2 conditional (if-then-else)

\(e_1\) and \(e_2\)
- evaluate \(e_1\), if \(e_1\) is "true-y" then evaluate \(e_2\) and return the value of \(e_2\)
- otherwise return the value of \(e_1\)

\(e_1\) or \(e_2\)
- not \(e_1\)
- evaluate \(e_1\), if \(e_1\) is "true-y" then return False, otherwise return True

\(e_1\) and \(e_2\)
- if \(e_1\) evaluates to True, execute \(e_2\)
- else return False

not \(e_1\)
- if \(e_1\) evaluates to True, return False
- otherwise return True

\(e_1\) and \(e_2\) 
- duplication! 
- let
if e2 else e3

If e2 is true then replace that and "plug back".

If e2 is false, then generate code to check if untagged is true.

"False" = 0 or 40 unh.

Name('False') Python/Pl

Tagged value (x86)

Untagged value (x86)
\begin{align*}
\text{Constants} & \quad \text{Constant Folding (Partial Evaluation)} \\
0 \text{ and } e_2 \Rightarrow 0 \\
\text{And}(\text{Const}(n_1, e_2)) \\
\text{And}(q_1, e_2)
\end{align*}