Lisp Machines and the Analysis of Their High-Level Language Computer Architecture

Dominic Dabish and Evan Bradley

Oakland University

{*dadabish, edbradley*}*@oakland.edu*

April 2016

Overview

1. History of Lisp machines
2. How Lisp works
3. Problems in Execution
4. Example processor
5. Legacy of Lisp machines

Early history of Lisp

- Lambda Calculus introduced in 1930s by Alonzo Church
- Fortran in 1957
- No programming languages optimized for artificial intelligence
- Lisp designed in 1958 by John McCarthy
- Lisp code implemented on IBM 170 months after

Lisp machines

- Lisp machines released in mid-1970s, became popular in 1980s
- Manufactured by Symbolics, Lisp Machines, Inc., Xerox, TI
- Offered GUIs, advanced programmability, flexibility
- Noncompetitive hardware
- Eventually became outperformed by general-purpose computers
- Vendors went bankrupt in 1990s

Functions in Lisp

Consider a simple function `g`, which takes one argument `x`.

Example (Function in mathematics)

```
g(x)
```

In Lisp, this is written in the following manner:

Example (Function in Lisp)

```
(g x)
```

Lists

Consider a simple array of characters in Java:

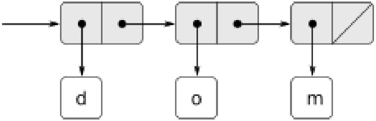
Example (Array in Java)

```
char[] myArray = ['d', 'o', 'm'];
```

In Lisp, we can write the elements as symbols, and our data structure is a `list`.

Example (List in Lisp)

```
'(d o m)
```



List-manipulating functions

car and cdr are primitive operations for lists.

- car extracts the first element from the list
- cdr extracts the rest of the list

Example (car)

```
(car '(d o m)) => 'd
```

Example (cdr)

```
(cdr '(d o m)) => '(o m)
```

Defining a recursive function

Let us define a factorial function in Lisp using recursion:

Example (Factorial function in Lisp)

```
(defun factorial(n)
(if (= n 0)
1
(* n (factorial (- n 1))
)
)
```

Evaluating a recursive function

Consider the following function call:

Example (Factorial function call in Lisp)

```
(factorial 5)
```

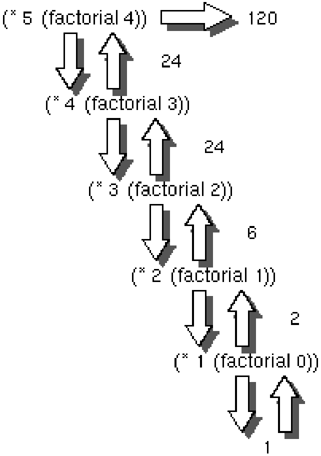
Here is its evaluation path:

Function call	Evaluation 1	Eval 2	Eval 3
`(factorial 5)`	`(* 5 (factorial 4))`	`(* 5 24)`	120
`(factorial 4)`	`(* 4 (factorial 3))`	`(* 4 6)`	24
`(factorial 3)`	`(* 3 (factorial 2))`	`(* 3 2)`	6
`(factorial 2)`	`(* 2 (factorial 1))`	`(* 2 1)`	2
`(factorial 1)`	`(* 1 (factorial 0))`	`(* 1 1)`	1

Table: Complete recursive evaluation

Recursive evaluation example explained

The program checks if the argument `n` is equal to 0 (the base case). If so, the function evaluates to 1 and terminates; otherwise, the function will multiply the argument by the factorial with `n-1` as the argument instead. This creates a chain, and evaluates to the factorial of `n`.



Paragraphs of Text

Sed iaculis dapibus gravida. Morbi sed tortor erat, nec interdum arcu. Sed id lorem lectus. Quisque viverra augue id sem ornare non aliquam nibh tristique. Aenean in ligula nisl. Nulla sed tellus ipsum. Donec vestibulum ligula non lorem vulputate fermentum accumsan neque mollis.

Sed diam enim, sagittis nec condimentum sit amet, ullamcorper sit amet libero. Aliquam vel dui orci, a porta odio. Nullam id suscipit ipsum. Aenean lobortis commodo sem, ut commodo leo gravida vitae. Pellentesque vehicula ante iaculis arcu pretium rutrum eget sit amet purus. Integer ornare nulla quis neque ultrices lobortis. Vestibulum ultrices tincidunt libero, quis commodo erat ullamcorper id.

Blocks of Highlighted Text

Block 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

Block 2

Pellentesque sed tellus purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vestibulum quis magna at risus dictum tempor eu vitae velit.

Block 3

Suspendisse tincidunt sagittis gravida. Curabitur condimentum, enim sed venenatis rutrum, ipsum neque consectetur orci, sed blandit justo nisi ac lacus.

Multiple Columns

Heading

1. Statement
2. Explanation
3. Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

Theorem

Theorem (Mass–energy equivalence)

$E = mc^2$

Figure

Uncomment the code on this slide to include your own image from the same directory as the template .TeX file.

Citation

An example of the `\cite` command to cite within the presentation:

This statement requires citation [Smith, 2012].

References

John Smith (2012)
Title of the publication
Journal Name 12(3), 45 – 678.

The End