CMSC 330: Organization of Programming Languages

Data Languages
Logic Languages
Language Types

• Procedural
• Functional
• Data
• Logic
Data Languages

• Markup
  – HTML, XML, LaTex, MediaWiki, etc.

• Query
  – SQL (Structured Query Language)

• Rule-based
  – R++
Markup Languages

- HTML (HyperText Markup Language)
  - World-Wide Web (WWW) document encoding
- XML (eXtensible Markup Language)
  - General data encoding
- LaTeX
  - Document formatting and typesetting
- MediaWiki / BBcode
  - General-purpose text encoding
- reStructuredText
  - Simplified general-purpose text encoding
Markup Languages

• A markup language’s general purpose is to impose structure on raw textual data

• Reasons:
  – Formatting and rendering
  – Standardization
  – Serialization
  – Data sharing (export/import)
  – Self-documentation
GML and SGML

• General Markup Language (GML)
  – Developed in the 1960s at IBM
  – Originally a set of macros for a text formatting program
  – Defined document structure: headers, paragraphs, lists, tables, lines, etc.

• Standard Generalized Markup Language (SGML)
  – Standardization of GML in 1986 by ISO
  – Several variations
  – Provided inspiration for HTML and XML
SML Example

Chapter 1: Introduction

GML supported hierarchical containers, such as:
- Ordered lists (like this one),
- Unordered lists, and
- Definition lists.

 Markup minimization (later generalized and formalized in SGML), allowed the end-tags to be omitted for some elements (e.g., "h1" and "p").
HTML

• Basic concept: tag

  <p>This is some <b>bold</b> text.</p>

  This is some **bold** text.

• **Elements** begin and end with tags
  – Data in between (optional)

• Some elements have **attributes**
  – Name/value pairs in opening tag

  <p>This is <font color="green">green</font> text.</p>

  This is **green** text.
HTML History

• 1991 – “HTML Tags” (original proposal)
  – By Tim Burners-Lee, founder of WWW project
• 1993 – First release of NCSA Mosiac
  – First popular WWW browser
• 1995 – HTML 2.0
  – First committee standard
• 1997 – HTML 3.2
  – First WWW Consortium (W3C) standard
HTML History

• December 1997 – HTML 4.0
  – Strict (no deprecated elements)
  – Transitional
  – Frameset (only frame-based elements)

• 1999 – HTML 4.01
  – Minor differences
  – Current full standard

• 2008 – First draft of HTML5
  – New multimedia constructs
XML

- Can encode nearly any type of data
- Two standards
  - XML 1.0
    - Published in 1998
    - Latest version in 2008
  - XML 1.1
    - Published in 2004
    - Latest version in 2006
    - Supports more scripts and characters than XML 1.0
    - Not widely implemented yet
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <description>An in-depth look at creating applications with XML.</description>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <description>A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.</description>
  </book>
</catalog>
XML

• Provides a template for data representation
• Must provide grammars for specific domains
  – Document Type Definitions (DTDs)
  – XML Schema Definitions (XSDs)

• Sample DTD:

```xml
<!DOCTYPE catalog [ 
<!ELEMENT catalog (book)*> 
<!ELEMENT book (author, title, genre?, description?)> 
<!ATTLIST book id ID > 
<!ELEMENT author (#PCDATA)> 
<!ELEMENT title (#PCDATA)> 
<!ELEMENT genre (#PCDATA)> 
<!ELEMENT description (#PCDATA)> ]>
```
XML

• Many wide-spread XML-based languages
  – GraphML / MathML
  – OpenDocument / MS Office Open
  – RSS / Atom
  – SOAP
  – XAML / XUL
  – XHTML
  – XPath
XML Tradeoffs vs. Binary Formats

• Advantages
  – Self-describing
  – Portability
  – Interchangeability

• Disadvantages
  – More storage space required
  – Expensive to encode/decode
LaTeX

• Document formatting and typesetting
• Language for TeX typesetting program
  – Designed by Don Knuth and first released in 1978
• Widely used by mathematicians and other academic/technical writers
• Usually used in conjunction with BibTex, an XML-like bibliography description language
LaTeX Example

\documentclass[12pt]{article}
\usepackage{amsmath}
\title{LaTeX}
\date{}
\begin{document}
\maketitle
LaTeX is a document preparation system for the \TeX{} typesetting program. It offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout, bibliographies, and much more. \LaTeX{} was originally written in 1984 by Leslie Lamport and has become the dominant method for using \TeX{}; few people write in plain \TeX{} anymore. The current version is \LaTeXe.  

% This is a comment; it will not be shown in the final output.
% The following shows a little of the typesetting power of LaTeX:
\begin{align}
E &= mc^2
\\%

m &= \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}
\end{align}

\end{document}
\textsc{LaTeX} Example

\textsc{LaTeX} is a document preparation system for the \TeX{} typesetting program. It offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout, bibliographies, and much more. \textsc{LaTeX} was originally written in 1984 by Leslie Lamport and has become the dominant method for using \TeX{}; few people write in plain \TeX{} anymore. The current version is \textsc{LaTeX} 2\varepsilon.

\begin{align}
E &= mc^2 \\
\frac{m}{m_0} &= \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} 
\end{align}
Wiki/Forum Markup

• BBCode (1998)
  – Prevented malformed HTML from crashing forums
  – Can be customized by forum administrators

• MediaWiki (2002)
  – Designed for Wikipedia editors
  – Easy to learn
  – Standardized formatting

• reStructuredText (2002)
  – Developed for Python code documentation
  – Very minimalist philosophy
Introduction
==============

Here is a shopping list:

- milk
- sandwich ingredients
  - bread
  - peanut butter
  - jelly
- orange juice
- cookies

<table>
<thead>
<tr>
<th>Sample Table</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Final</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>

For more info, see their website.

.. _website: http://docutils.sourceforge.net/rst.html
SQL (Structured Query Language)

- Originally SEQUEL
  - Structured English QUEry Language
  - Developed in 1970s at IBM
  - Standardized in 1986 (ANSI) and 1992 (ISO)

- Designed for querying and updating databases
  - Natural English syntax

```
[UPDATE clause
  [SET population = population + 1
    [WHERE name = 'USA';]
  [WHERE clause]]
```

CMSC 330
SQL

• Statements
  – SELECT
  – INSERT / UPDATE / DELETE
  – CREATE / DROP

• Clauses
  – FROM
  – WHERE / HAVING
  – GROUP BY
  – ORDER BY
SQL

• Advantages
  – Allows non-programmers to manipulate data
  – Can archive common queries

• Disadvantages
  – Every database engine has slightly different implementations and language extensions
R++

• Extension to C++ that allows rule-based data event triggers

• The sample rule shown below expresses a company's policy that employees must retire upon reaching age 65.
  – Member of class Employee
  – Triggered whenever an employee's age or status changes.

```cpp
rule Employee::retirement_policy {
  age >= 65 &&
  status != retired
=>
  cout << name << " must retire."
}
```
Logic Languages

• Planner (1969)
  – Hybrid procedural and logic language

• Prolog (1972)
  – “PROgrammation en LOGique”
  – Many extensions and variants
    • Visual Prolog, λProlog, etc.

• Mercury (1995)
  – “Real-world” logic language

• Curry (1999)
  – Extension to Haskell
  – Hybrid functional and logic language
Logic Programming

- **Facts**
  - Pieces of information provided by the programmer
  - Can be individual entities (existence assertions)
  - Can be assertions about entities

- **Relationships**
  - Define connections and associations between facts

- **Rules**
  - Allows the deduction of new facts and relationships from those already defined
Logic Programming

• Forward chaining
  – Start with assertions
  – Prove new statements

• Backward chaining
  – Start with goals and assertions
  – Prove goals
  – Match rules against goals and then recursively prove antecedents
  – This is the default Prolog behavior
A Song of Facts and Logic

- Tywin
  - Robert
    - Cersei
      - Joffrey
      - Myrcella
      - Tommen
    - Tyrion
Game of Rules

- Facts:
  male(tywin).
  female(cersei).
  male(joffrey).
  male(tommen).

- Relations:
  parent(cersei, tywin).
  parent(joffrey, cersei).
  parent(tommen, cersei).
Game of Rules

- **Rules with variables:**
  
  \[
  \text{human}(X) :\text{male}(X).
  \]
  
  \[
  \text{human}(X) :\text{female}(X).
  \]
  
  \[
  \text{father}(X,Y) :\text{parent}(X,Y), \text{male}(Y).
  \]
  
  \[
  \text{mother}(X,Y) :\text{parent}(X,Y), \text{female}(Y).
  \]

- **Recursion**
  
  \[
  \text{ancestor}(X,Y) :\text{parent}(X,Y).
  \]
  
  \[
  \text{ancestor}(X,Y) :\text{parent}(X,Z), \text{ancestor}(Z,Y).
  \]
father(cersei,tywin).
  yes
father(joffrey,tywin).
  no
mother(joffrey,X).
  cersei
mother(X,cersei).
  joffrey, tommen
ancestor(joffrey,X).
  cersei, tywin
GNU Prolog

• Free, open-source implementation of Prolog
  – http://www.gprolog.org/

• Run interactive interpreter: gprolog
  – End each statement with a “.”
  – Use “:-” to define a rule
  – Load file: consult('test.pro').
  – Press CTRL-D to exit (CTRL-Z on Windows)

• Compile database: gplc test.pro
  – Compiles file into self-contained executable (test)
  – Includes interactive interpreter
Lists in Prolog

- Lists consist of a **head** and a **tail**
  - Just like OCaml
  - The head is a single element
  - The tail is another list
- Create lists with `[]` and `,`
  - `[]`
  - `[1]`
  - `[1,2,3]`
- Pattern-match lists with `|`
  - `[H|T]`
Lists in Prolog

head(H,[H|_]).
tail(T,[_|T]).
first(F,[F|_]).
last(L,[L]).
last(L,[H|T]) :- last(L,T).

length([],0).
length([_|T],N) :- length(T,M), N is M+1.

mem(X,[X|_]).
mem(X,[_|T]) :- mem(X,T).
nth_mem(1,[M|_],M).
nth_mem(N,[_|T],M) :- N>1, N1 is N-1, nth_mem(N1,T,M).

append([],L,L).
append([H|T],L,[H|LT]) :- append(T,L,LT).
Lists in Prolog

?- head(X,[1,2,3]).
X = 1

?- tail(X,[1,2,3]).
X = [2,3]

?- last(X,[1,2,3]).
X = 3

?- length(X,[1,2,3,4]).
X = 4

?- mem(4,[1,2,4]).
yes

?- mem(4,[1,2,3]).
no

?- nth_mem(3,[6,7,8,9],X).
X = 8

?- append([1,2,3],[4,5,6],X).
X = [1,2,3,4,5,6]
% move(N,X,Y,Z) - move N disks from peg X to peg Y, with peg Z being the auxiliary peg
%
% Base Case: One disc - To transfer a stack consisting of 1 disc from peg X to peg Y, simply move that disc from X to Y
% Recursive Case: To transfer n discs from X to Y, do the following:
% Transfer the first n-1 discs to some other peg X
% Move the last disc on X to Y
% Transfer the n-1 discs from X to peg Y

move(1,X,Y,_):-
  write('Move top disk from '),
  write(X),
  write(' to '),
  write(Y),
  nl.
move(N,X,Y,Z):-
  N>1,
  M is N-1,
  move(M,X,Z,Y),
  move(1,X,Y,_),
  move(M,Z,Y,X).
Towers of Hanoi in Prolog

?- move(3,left,right,center).
  Move top disk from left to right
  Move top disk from left to center
  Move top disk from right to center
  Move top disk from left to right
  Move top disk from center to left
  Move top disk from center to right
  Move top disk from left to right

yes
Prolog

• Many other cool Prolog examples and applications
• Some resources:

http://www.learnprolognow.org/

http://www.csupomona.edu/~jrfisher/www/prolog_tutorial/contents.html

http://www.inf.ed.ac.uk/teaching/courses/aipp/

http://www.freetechbooks.com/artificial-intelligence-through-prolog-t471.html
Q. How many Prolog programmers does it take to change a lightbulb?

A. no.