Computational Linguistics Seminar
LING-696G

Week 1
Syllabus

• Instructor:
  – Sandiway Fong (Douglass 311), sandiway@email.arizona.edu

• Time:
  – 3:30pm-6pm Wednesdays (except March 3rd)

• Place:
  – Douglass 206 (or my lab upstairs)

• Office hours:
  – by appt. or drop by...

• Course objectives:
  – Implementation of (reasonably) current syntactic theory.
  – Read carefully papers like Derivation by Phase, On Phases, Pesetsky & Torrego (2001) and others, e.g. Gallego (2005)
  – Work through derivations in class (active participation expected) and on the computer.

• Deliverable:
  – Choose a suitable paper. Implement it and write a report.
Computer Software

• Prototype of a Minimalist Machine
  – still under development, i.e. expect changes
  – input:
    • stream (and substreams) of heads
  – output:
    • assembled syntactic object(s), i.e. multiple derivations may be output
  – grammar:
    • selection of heads (and features)
    • spell out: language-particular morphological realization of heads (and features) – strictly speaking, not part of narrow syntax
    • currently: for fragments of English, Arabic and Japanese
Homework

• **Part 1:** read (re-read?) *Derivation by Phase* (Chomsky 2001)
  – I've emailed the PDF to you already
  – review the basic model
  – familiarize yourself with the syntactic operations performed by the heads, e.g. T, T-defective, v*, PRT, etc.
  – be prepared to work through derivations in class
  – write down questions you have (*collectively, we'll try to answer them*)

• **Part 2:** install the prototype software
  – I'll email you the code (*not available on my webpage*)
  – should not be re-distributed outside of the class

• **Part 3:** install and re-familiarize yourself with Prolog
Minimalist Machine

• Software Architecture:
  – client/server using sockets
  – general background LING 408/508, lectures 16 and 17
    http://dingo.sbs.arizona.edu/~sandiway/ling508-14/lecture16.pdf
    – http://dingo.sbs.arizona.edu/~sandiway/ling508-14/lecture17.pdf
  – uses the WebSocket interface
    – a fairly new standard (RFC dated Dec 2011)
    – client:
      • html5 program
      • runs on a web browser (I don't care which one...)
      • **must** support the WebSocket interface
    – server:
      • machine that computes derivations
      • communicates with the client
      • written in SWI-Prolog
      • has a WebSocket API (currently **only** in development releases of SWI Prolog)
SWI-Prolog with Websocket interface

• [www.swi-prolog.org](http://www.swi-prolog.org)

<table>
<thead>
<tr>
<th>Binaries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SWI-Prolog 7.1.28 for Windows XP/Vista/7/8</strong></td>
</tr>
<tr>
<td>Self-installing executable for MS-Windows. Installs <code>swipl-win.exe</code> and <code>swipl.exe</code> for XP/Vista/7/8. This binary is linked against GMP 5.0.5, which implies that it is covered by the LGPL-V3 license. See below.</td>
</tr>
<tr>
<td>11,635,651 bytes</td>
</tr>
<tr>
<td><strong>SWI-Prolog 7.1.28 for Windows XP/Vista/7/8 64-bit edition</strong></td>
</tr>
<tr>
<td>Self-installing executable for Microsoft's XP/Vista/7/8 64-bit editions. See the reference to whether to use the 32- or 64-bits version. This binary is linked against GMP 5.0.5, which implies that it is covered by the LGPL-V3 license. See below.</td>
</tr>
<tr>
<td>12,228,077 bytes</td>
</tr>
<tr>
<td><strong>SWI-Prolog 7.1.28 for MacOSX 10.6 (Snow Leopard) and later on intel</strong></td>
</tr>
<tr>
<td>Mac OS X disk image with relocatable application bundle. Needs <code>xquartz</code> (X11) development tools. Currently, version 2.7.7 is required. You can check the version of <code>xquartz</code> application and then checking 'about' in the X11 menu. The graphical application bundle also provides the commandline tools in <code>Contents/MacOS</code>. The command <code>swipl</code> runs the graphical application bundle as well.</td>
</tr>
<tr>
<td>29,442,615 bytes</td>
</tr>
</tbody>
</table>

• **Linux (Ubuntu)**

```
% sudo apt-add-repository ppa:swi-prolog/devel
% sudo apt-get update
% sudo apt-get install swi-prolog
```
Websocket

• part of the HTML5 standard
• once a Websocket connection is established
  – messages can be passed between a client and a server
• Protocol:
  – establish a normal TCP/IP socket first
  – then the client asks for an upgrade to Websocket using a HTTP GET request
  – has some built-in authentication
    • Client sends
      – Sec-WebSocket-Key: dGhlIHNhbXBsZSBzdWJzb2VudGljYWxpZz==
    • Server computes
      – concatenate with (fixed): 258EAFA5-E914-47DA-95CA-C5AB0DC85B11
      – SHA-1 hash it and base-64 encode it
      – send it back to the client in the Sec-WebSocket-Accept field
Minimalist Machine

- File: dist.zip for OSX

1. In Terminal, run the file `mmachine`
   ```bash
   ./mmachine
   ```
2. Double-click on `index.html`. It should fire up in Safari.
   Alternatively:
   ```bash
   open index.html
   ```
Minimalist Machine

- Javascript (on client):
  ```javascript
  var port = 8010;
  var wsUri = "ws://localhost:" + port + "/ws"; // application lives here
  var websocket;

  function testWebSocket() {
    writeToScreen(" "+ port);
    websocket = new WebSocket(wsUri);
    websocket.onopen = function(evt) { onOpen(evt) };
    websocket.onclose = function(evt) { onClose(evt) };
    websocket.onmessage = function(evt) { onMessage(evt) };
    websocket.onerror = function(evt) { onError(evt) };
  }
  ```
Minimalist Machine

- Javascript (on client):

```javascript
function onOpen(evt) {
    writeToScreen(" CONNECTED");
}
function onClose(evt) {
    writeToScreen(" DISCONNECTED");
}
function onMessage(evt) {
    // writeToScreen('<span style="color: blue;">IN: ' + evt.data+'</span>');
    var x = eval(evt.data);
    doSend(x);
}
function onError(evt) {
    writeToScreen('<span style="color: red;">ERROR:</span> ' + evt.data);
}
function doSend(message) {
    // writeToScreen("OUT: " + message);
    websocket.send(message);
}
function writeToScreen(message) {
    var pre = document.createElement("span");
    pre.innerHTML = message;
    o.appendChild(pre);
}
```
Minimalist Machine

On server (SWI-Prolog side):

```prolog
:- use_module(library(http/websocket)).
:- use_module(library(http/thread_httpd)).
:- use_module(library(http/http_dispatch)).

start(Port) :- http_server(http_dispatch, [port(Port),workers(1)]).
stop(Port) :- http_stop_server(Port,[]).

:- http_handler(root(ws),
               http_upgrade_to_websocket(echo, []),
               [spawn([])]).

echo(WebSocket) :-
  retractall(ws(_)),
  asserta(ws(WebSocket)),
  ws_receive(WebSocket, M),
  (M.opcode == close
      |   -> format('Received close~n')
      ;
  format('Received:~w~n', [M.data]),
  term_string(Term,M.data),
  call(Term)
).```
Minimalist Machine

• On server (SWI-Prolog side):

```prolog
process_in(I) :-
    M = websocket<data:I,format:string,opcode:text>,
    ws(WebSocket),
    ws_send(WebSocket, M),
    ws_receive(WebSocket, Reply),
    ( Reply.opcode == close
    -> format('Received close~n')
    ; true).

process_in(I,Ans) :-
    M = websocket<data:I,format:string,opcode:text>,
    ws(WebSocket),
    ws_send(WebSocket, M),
    ws_receive(WebSocket, Reply),
    ( Reply.opcode == close, Ans = false
    -> format('Received close~n')
    ; (number_string(Ans,Reply.data)
    -> true
    ; atom_string(Ans,Reply.data)
    )
    ).
```
Minimalist Machine

- Website:
  - http://dingo.sbs.arizona.edu/~sandiway/mpp/mm.html

**Derivations**

For full step-by-step derivations, visit the following links.

<table>
<thead>
<tr>
<th>Section</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic sentence patterns</td>
<td>link</td>
</tr>
<tr>
<td><em>Derivation by Phase</em>. Chomsky (2001).</td>
<td>link</td>
</tr>
<tr>
<td><em>T-to-C Movement: Causes and Consequences</em>. Pesetsky &amp; Torrego (2001).*</td>
<td>link</td>
</tr>
<tr>
<td><em>Explaining TH/EX</em> (Sobin, ms.) and <em>TH/EX, Agreement, and Case in Expletive Sentences</em> (Sobin, Syntax: to appear).</td>
<td>link</td>
</tr>
<tr>
<td>Computation with doubling constituents: Pronouns and antecedents in Phase Theory. (Fong &amp; Ginsberg, 2002).</td>
<td>link</td>
</tr>
<tr>
<td>Test cases from <em>A Course in GB Syntax</em> (Lasnik &amp; Uriagereka, 1988).</td>
<td>link</td>
</tr>
<tr>
<td>Some Arabic examples (Monica Larcom &amp; Sandiway Fong)</td>
<td>link</td>
</tr>
<tr>
<td>Some Japanese examples (Gustave Hahn-Powell &amp; Sandiway Fong)</td>
<td>link</td>
</tr>
</tbody>
</table>
Minimalist Machine

• Demo time ...