CMPU101
Problem Solving and Abstraction

Lecture 5

String and Non-Empty List Functions
CONS, FIRST, REST functions
Non-empty Lists

Lists are a self-referential data structure for data sets of arbitrary size.

Smallest possible list is ?

A list has many parallels with a string. There are many similar functions for each data type (because they are similar in structure).
list->string & string->list

The list->string function consumes a list of characters and returns a string.

The string->list function consumes a string and produces a list of characters.

These functions allow us to use a string with list functions if necessary. Just convert the string to a list, apply the list function, and convert the list back to a string.
String functions

Many string functions correspond to list functions:

<table>
<thead>
<tr>
<th>STRING</th>
<th>LIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>list</td>
</tr>
<tr>
<td>string-ref</td>
<td>list-ref</td>
</tr>
<tr>
<td>string-append</td>
<td>append</td>
</tr>
<tr>
<td>string-&gt;list</td>
<td>list-&gt;string</td>
</tr>
<tr>
<td>none</td>
<td>reverse</td>
</tr>
</tbody>
</table>
Building and accessing lists

There exist functions that build a list one item at a time (all actually in lowercase):

- **CONS**  Add item to left end of list
- **FIRST** Returns item at left end of list
- **REST**  Returns the whole list except 1\(^{st}\) item

(first corresponds to the old lisp term car)
(rest corresponds to the old lisp term cdr)
SELF-REFLEXIVE DATA TYPE

SELF-REFERENTIAL DATA DEFINITION FOR A LIST OF STRINGS:
A list of strings (LOS) is either:
1. empty, or
2. it's a (cons s los), where s is a string and los is an LOS
Self-referential data type

SELF-REFERENTIAL DATA DEFINITION FOR A LIST OF NUMBERS:
A list of numbers (LON) is either:
1. empty, or
2. it's a (cons n lon), n is a number and lon is an LON.

You'll see that functions used to process lists have a similar structure to this definition.