CSE443
Compilers

Dr. Carl Alphonce
alphonce@buffalo.edu
343 Davis Hall
Phases of a compiler

Intermediate Representation (IR): specification and generation

Figure 1.6, page 5 of text
Project notes
Helpful Links

- **MIDRULE ACTIONS**
  - Bison manual: Using mid-rule actions
  - Bison manual: How mid-rule actions are translated

- **ERROR HANDLING**
  - Bison manual: error reporting
  - Bison manual: error recovery
  - Article and sample code from IBM showing error handling
Helpful Links

- Type look-up for primitive types?
- Marker non-terminal rules
- %union for type checking
Intermediate Representations
Our language
(use name equivalence)

- **pre-defined types:**
  - **primitive types:** integer, real, Boolean, character
  - **composite type:** string

- **user-defined types:**
  - **record types have names**
    - type rec : [ real : x , y ]
  - **array types have names**
    - type arr : 2 -> string
  - **function types have names**
    - type fun : ( real : x ) -> rec
Recursive records
Recursive functions

A record type must allow a component to be of the same type as the type itself:

```plaintext
type Node: [ integer datum:=0 ; Node rest:=null ]
```
type information

- type indicates size
- type indicates storage location
  - primitives: either stack or heap
  - records: on heap (via pointer)
  - arrays: on heap (via pointer)
  - functions: code in static, locals on stack
- need to determine how to lay out records, arrays, invocation records in memory
Sizes of types

- int: 32 bits (2's complement)
- real: 64 bits (IEEE 754)
- Boolean: 8 bits (TBD)
- character: 8 bit (ASCII)
Sizes of types

- type string: 1 -> character
- 4 bytes + length of string * size of character (= 1 byte)
- # of dimensions is part of type

<table>
<thead>
<tr>
<th>size of dimension 1 (integer)</th>
<th>(0)</th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>V</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S</td>
</tr>
</tbody>
</table>

https://en.wikipedia.org/wiki/VAX
What is the size of a multi-dimensional array of type T?

**sizes of dimensions \((S_i)\): X*4 bytes**

**data: \((T \times S_i) \times \text{sizeof}(T)\)**

- First dimension
  - 0
  - 0
  - 2
- Second dimension
  - 0
  - 0
  - 3

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>size of first dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>a(0,0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a(0,1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a(0,2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a(1,0)</td>
<td></td>
<td>first row</td>
</tr>
<tr>
<td>a(1,1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a(1,2)</td>
<td></td>
<td>second row</td>
</tr>
</tbody>
</table>