Interpreter & Compiler: exercises

Lectures 14-15-16

Formal Languages and Compilers 2011

Nataliia Bielova

Exercises for compiler

- for each in vectors (test_foreach.cre)

```
var 1: array[5] of int;
...
for each i in v do
     begin
       i := n;
       n := n - 1;
     end;
```

Intermediate code for FOR

- Node in syntax tree `For (i, 1, n, cmd)`
- Generated code (gencommand in commands.ml)

```
startfor:   CG      increg      n    tmp     // tmp = (increg > n)
            JNE     tmp         0    endfor  // if (increg > n) then jump
            ...                              // code generated for cmd
            ADD     increg      1       tmp
            CPY     tmp         NULL increg
            GOTO    startfor    NULL    NULL
endfor:     ...
```

Intermediate code for FOR EACH

- Node in syntax tree `Foreach (e, v, cmd)`
- Lower bound start=0
- Upper bound finish= dim-1
- Take a new increg = start
- start loop (increg <= finish)
- e ← v[increg]
- generate commands for cmd
- v[increg] ← e
- increg ← increg +1
- end loop

Exercises for compiler

- interactive input from console (test_read.cre)

```
var i: int;
var f: float;
...
readInt(i);
readFloat(f);
```

- change: lexer, parser, syntax tree
- type checking (semantic.ml)
- intermediate code generation (commands.ml)
- + declaration of new intermediate code command (intermediate.ml)
- + target code generation (target.ml)

Exercises for interpreter

- Lecture 9: code of the interpreter
- Implement **for each** in vectors (test_foreach.cre)

```
var 1: array[5] of int;
...
for each i in v do
    begin
      i := n;
      n := n - 1
    end;
```

For: semantics

$$\mathcal{C}\,\|\ \texttt{for i := min to max do c}\|_{\texttt{rs}} =$$

$$= \mathcal{C}'\,\|\ \texttt{for i := min to max do c}\|_{\texttt{rs0}}$$

where :

$$l_{\texttt{i}} = \Lambda\|\ \texttt{i}\ \|_{\texttt{rs}}$$

$$\texttt{s0} = \texttt{updatemem}(\texttt{s}, l_{\texttt{i}}, \texttt{min})$$

$$\mathcal{C}'\,\|\ \texttt{for i := min to max do c}\|_{\texttt{rs0}} =$$

$$= \begin{cases} \mathcal{C}'\,\|\ \texttt{for each e in v do c}\|_{\texttt{rs}''} & \texttt{if}\ \ \texttt{B}\|\texttt{i < max}\| = \mathit{true} \\ \texttt{s}' & \texttt{otherwise} \end{cases}$$

where :

$$\texttt{s}' = \mathcal{C}\,\|\ \texttt{c}\ \|_{\texttt{rs}}$$

$$\texttt{v} = \mathcal{E}\|\ \texttt{i}\ \|_{\texttt{rs}}$$

$$\texttt{s}'' = \texttt{updatemem}(\texttt{s}', l_{\texttt{i}}, \texttt{v + 1})$$

For each : semantics

$\mathcal{C} \| \texttt{for each e in v do c} \|_{rs} =$

$$= \begin{cases} \mathcal{C} \| \texttt{for each e in v do c} \|_{rs''} & \texttt{if } i \leq \texttt{max} \\ \texttt{s} & \texttt{otherwise} \end{cases}$$

local variables : min = lb, max = ub, i = min

where :

$s0 = \texttt{updatemem}(s, l_e, v_{vi})$

$l_e = \Lambda \| \texttt{ e } \|_{rs}$

$v_{vi} = E \| \texttt{ v[i] } \|_{rs}$

$s' = \mathcal{C} \| \texttt{ c } \|_{rs0}$

$s'' = \texttt{updatemem}(s', l_{vi}, v_e)$

$l_{vi} = \Lambda \| \texttt{ v[i] } \|_{rs'}$

$v_e = E \| \texttt{ e } \|_{rs'}$