Compiler Construction CSCI-GA.2130-001 Fall 2011 hw7

Assigned We 11/9/2011, due Fr 11/18/2011 at 1pm.

Reading Assignments

- For lecture on 11/16/2011: Dragon-book 8.1-8.4 + 8.6 (34 pages)

Homework Assignments

1. IR generation for jumping code ($8 = 2 + 6$ points).
   Consider the following TACK code:

   ```
   x = 0;
   while x < 5 || x % 4 != 0 {
     x := x + 1;
   }
   y = x;
   ```

   1a. (2 points) What is the value of `y` at the end?
       Recall that logical or (`||`) in TACK has short-circuit semantics.

   1b. (6 points) What are the IR instructions for this code?

2. IR generation for memory accesses (8 points).
   Consider the following TACK code:

   ```
   r = (f = 1);
   a = [ r ];
   a[0].f := 2;
   i = a[0].f;
   ```

   What are the IR instructions for this code?

3. Stack frames and calling conventions ($8 = 2 + 2 + 2 + 2$ points).

   We will refer to activation records as "AR" for short. Consider the drawings of AR layouts from the text-book and from the x64 introduction on the class webpage, reproduced here:

General activation record (see Dragon book)	x64 activation record (see x64-intro.pdf)
(Caller's AR)	(Caller's AR)
Actual parameters	`arg[n-1]`
Returned values	…
Control link	`arg[7]`
Access link	`arg[6]`
Saved machine status	Return address
Local data	Caller `%rbp`
Temporaries	Callee local variables and temporaries

   1a. (2 points) The x64 AR only shows actual parameters indexed 6 or higher.
       How are the remaining actual parameters passed on x64?

   1b. (2 points) The general AR has a slot with a "control link".
       What is the corresponding slot on x64?

   1c. (2 points) The x64 AR has a slot with the "return address".
       What is the corresponding slot in the general AR?

   1d. (2 points) The general AR has a slot for "returned values".
       How is the return value passed on x64?

Total points: 24.