CS 232, Jens Palsberg Jun 13, 2006

Final Exam

Each of the four questions is worth five points, for a total of 20 points. Write your name and id number at the top of the first page you submit. Write the solutions for different questions on different sheets of paper. Don't submit this handout.

1. [**Register allocation**] Suppose we have a machine with K 32-bit registers where K is even. The registers are called $r_1, \ldots, r_K$. Suppose also that any two registers r_i, r_{i+1}, where i is odd and $1 \leq i < K$, can be referenced as a pair using the name p_i. We can use a pair p_i to hold data of size 64 bits. Consider now the following register allocation problem. We have a program in **static single assignment** form with n variables of type int (32 bits each) and with m variables of type long (64 bits each). The core register allocation problem is whether each int-variable can be mapped to one of the r_i registers and each long-variable can be mapped to one of the p_i registers such that variables with interfering live ranges are assigned nonoverlapping registers. *Is there a polynomial-time algorithm for solving the core register allocation problem?* Explain your answer in detail. *Give a heuristic for allocating as many variables to registers as possible.*

2. [**Flow analysis**] In Java we can load a class at run time using `Class.forName(...)`, and we can create an object of a dynamically loaded class using `c.newInstance()`, where `c` is of type `Class`. Design a whole-program flow analysis to determine conservatively which variables and fields can hold an object of a dynamically loaded class.

3. [**Heintze-McAllester style flow analysis**] Here are the rules for Heintze-McAllester style flow analysis of a λ-calculus program P:

$$\frac{}{x \rightarrow dom(\lambda^\ell x.e)} \text{ (if } \lambda^\ell x.e \in P) \qquad \frac{}{ran(\lambda^\ell x.e) \rightarrow e} \text{ (if } \lambda^\ell x.e \in P)$$

$$\frac{}{dom(e_1) \rightarrow e_2} \text{ (if } (e_1\ e_2) \in P) \qquad \frac{}{(e_1\ e_2) \rightarrow ran(e_1)} \text{ (if } (e_1\ e_2) \in P)$$

$$\frac{n_1 \rightarrow n_2 \qquad n \rightarrow dom(n_2)}{dom(n_2) \rightarrow dom(n_1)} \qquad \frac{n_1 \rightarrow n_2 \qquad n \rightarrow ran(n_1)}{ran(n_1) \rightarrow ran(n_2)}$$

Phrase a Heintze-McAllester style flow analysis for a subset of Java with just classes, fields, methods with one argument and nonvoid return type, this, method call, field access, new, and variable reference. Give an example of a Java program for which the Heintze-McAllester style flow analysis does not terminate.

4. [**Access rights analysis**] Present a flow analysis of sequential Java programs which, for every program point, determines the access rights, represented as Permission objects, that are required for the code to execute without AccessControlExceptions.