CS 410/510 Compiling Advanced Languages Homework 2 – due Tuesday, April 18, 2000

Pencil and Paper

1. Consider the following datatype intended to represent terms in the pure λ-calculus.

```
type exp = Abs of int * exp
         | App of exp * exp
         | Var of int
```

This is just the same as in the file `lambda.ml` from assignment 1, except that variables are represented by integers instead of strings.

(a) Assuming that datatypes are represented using Assumption 1 and the CAML constructor convention as described in class, draw a picture of the runtime data structure that would be built for the λ-term $((\lambda f.\lambda x.(f\ x)\ x)(\lambda y.y))(\lambda z.z)$ How much memory does the term occupy, assuming an integer and a pointer each occupy a single word?

(b) Suppose the function `reduce` from `lambda.ml` were applied once to the term. How much *additional* memory (number of words) will be allocated for term storage in the course of execution? State any important assumptions you make and justify your answer.

Programming

2. File `core.ml` defines an interpreter for a subset of the core λ-calculus given in class.

(a) The rules for evaluting `Record` and `Select` expressions are missing; add them. Hint: all the ML features you need are already illustrated in the existing code.

(b) Add an explicit `Let` binding construct to the `lexp` type and a rule for evaluating it.

For extra credit: add support for the `Equal` primop, to operate on any `Labeled` or `Record` value.