CS153 Mid-Term Examination

Greg Morrisett

November 2, 2006

1. You are perusing the manual for a new language called Dish. Dish is a very dummed down version of Scheme. The manual gives the following BNF grammar to define the syntax of the language:

$$\begin{array}{lll} E & ::= & \mathsf{id} \mid \mathsf{num} \mid (\, L\,) \\ L & ::= & \epsilon \mid L\ E \end{array}$$

   The Dish designer claims that the language is LL1 and can thus be parsed with a recursive-descent parser. However, the grammar as given is not LL1. Rewrite the grammar so that it is LL1. (Note that ϵ denotes an empty string of terminals and non-terminals.)

2. Based on your new grammar, write by hand a recursive descent parser for Dish. Your parser should have the type:

```
parse : token list -> exp
```

   where the types `token` and `exp` are defined below:

```
datatype token = ID of string | NUM of int | LPAREN | RPAREN

datatype exp = Id of string | Num of int | App of exp list
```

3. Translate the following Fish code into MIPS assembly language. You may assume that the variables such as `x`, `y`, `z` and `i` have already been declared in the global data segment and thus their values can be accessed via symbolic labels, but do not assume that you know the initial values of these variables. You may also assume that the temporary registers are available for use.

   Try hard to minimize the execution time of your MIPS code but make sure that your assembly code is correct. In particular, you should consider loads and stores as taking 10 cycles, multiplies and divides as taking 5 cycles, and other operations as taking 1 cycle. You may ignore other issues (e.g., branch miss penalties.) All things being equal, a correct but slow implementation will receive more points than a fast but incorrect solution.

```
i = 0;
for (x = y; x != 0; x = x - z) {
   i = i + 2*x;
}
```

4. Briefly, and at a relatively high-level, explain the tasks that must be done to support procedure calls. In particular, explain what the caller must do to prepare for the call, and what the callee must do upon entry and exit of the procedure.