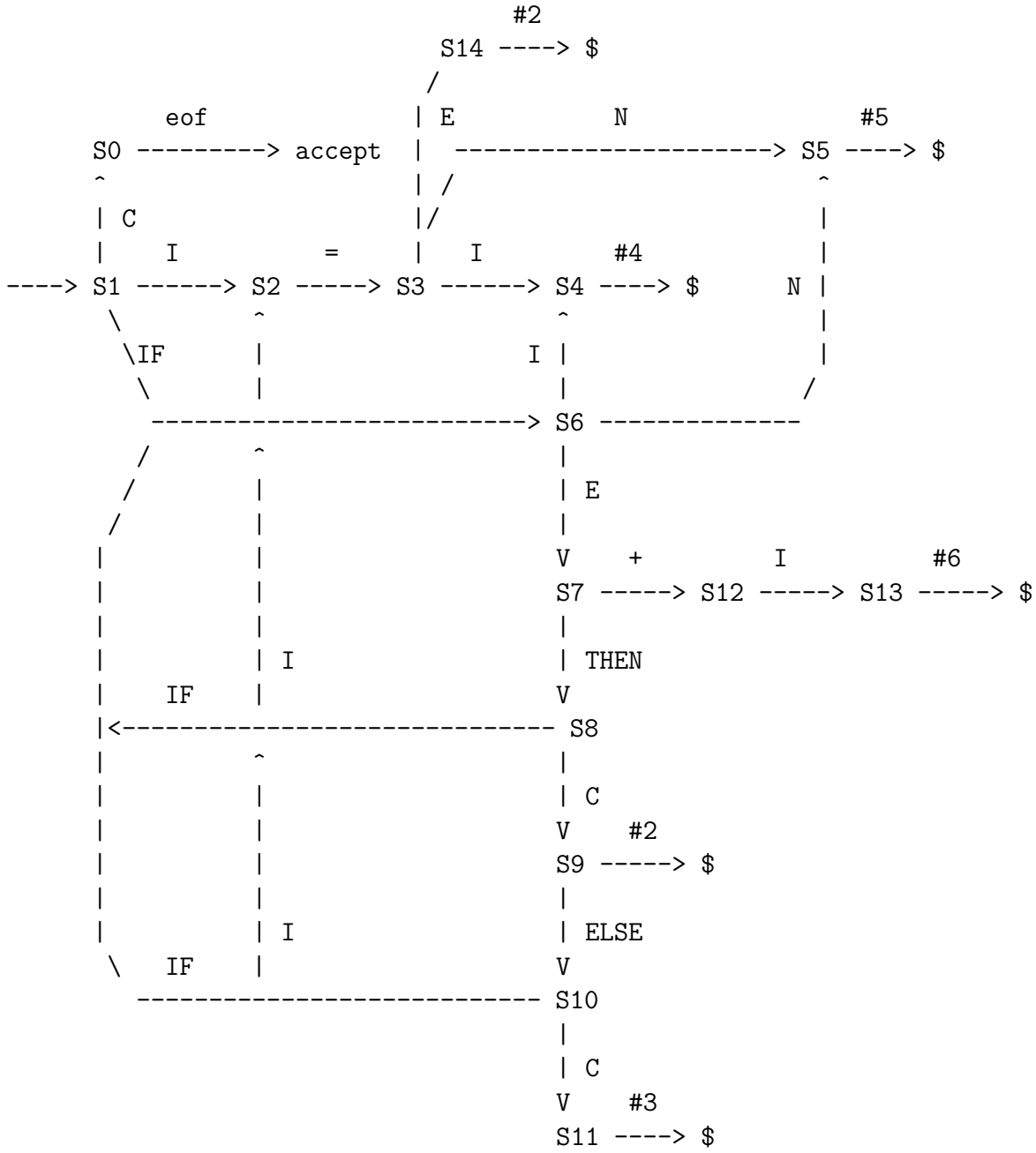
SOLUTION NOTES

Compiler Construction 2003 Paper 4 Question 1 (AM)

[This is a question on parsing.]

- Firstly the grammar is ambiguous, we will assume the normal resolution to "dangling else", i.e. that an `else` matches with the innermost of a choice of `if`s.
- For a recursive descent parser we must adapt E to remove left-recursion, and also left-factor the C cases.
- ```
  void rdS()
    {  lex(); // assume lex() sets global 'token'
       rdC(); if (token != EOF) error();
    }
  void rdE()
    { switch (token)
      {
  default: error();
  case I:
  case N: lex();
          if (token == '+') { lex(); rdE(); }
      }
    }
  void rdC()
    { switch (token)
  default: error();
  case I: lex(); if (token != '=') error();
          lex(); rdE(); return;
  case IF: lex(); rdE(); if (token != THEN) error();
           lex(); rdC(); if (token != ELSE) return;
           lex(); rdC(); return;
    }
  ```
  ```

- The CSFM for SLR (number productions #0...#6)



Note the shift-reduce conflict for state S9 which, if resolved in favour of shift (i.e. SLR(1)), gives the normal ambiguity resolution above.