1999

p6q6 AM p13q7

# Compiler Construction

**Compiling Techniques cmptech4.tex**

(a) It is desired to obtain an unambiguous context-free grammar $G'$ which generates the same strings as the grammar $G$ with start symbol S.

```
S -> E
E -> ( E ) | [ E ] | E * E | a | b | c
( E ) -> ( + E )
[ E ] -> [ - E ]
```

Define a suitable $G'$ or explain why $G$ already satisfies the criterion. [5 marks]

(b) Write a context-free (Type 2) grammar which describes floating point numbers of the form $[\pm]dd^*[.d^*][e[\pm]dd^*]$ where $d$ stands for decimal digit and $d^*$ stands for zero or more decimal digits. $[\cdots]$ means that the enclosed item is optionally present in the floating point number. [7 marks]

(c) Explain, giving brief reasons, whether it would be possible to construct a regular (Type 3) grammar which recognised the same syntax of numbers as your answer to part (b) [4 marks]

(d) Sketch a recursive descent parser for the grammar $H$ with start symbol S:

```
P -> 1 | 2 | (E)
E -> P | E - P
S -> E eof
```

assuming a routine `lex()` which sets variable `token` to one of '1', '2', '(', ')', '-' or `eof`. [5 marks]

~~Is this too long?~~

## Model Answer

(a)

```
S -> E
E -> F | E * F
F -> ( + E ) | [ - E ] |  a | b | c
```

(b)

```
S -> + F | - F | F
F -> DS | DS . | DS . DS |
     DS E | DS . E | DS . DS E
E -> e + DS | e - DS | e DS
DS -> D | D DS
D -> 0|1|...|9
```

(c) Yes, the grammar is regular, it is only the expression of it above in (b) which is Type 2, hence there exists a Type 3 form

```
A -> a
A -> a B
```

(d)

```
void RdP()
{ switch (token)
  { case '(': lex(); RdE();
              if (token != ')') error("expected ')');
              lex(); return;
    case '1': lex(); return;
    case '2': lex(); return;
    default:  error("unexpected token");
  }
}

void RdE()
{ RdP();
  for (;;) switch (token)
  { case '-': lex(); RdP(); continue;
    default:  return;
  }
}

void RdS()
{ RdE();
  switch (token)
  { case 'eof': return;
    default:  error("just (not eof) after E");
  }
}
```

Lose lots of marks for calling `RdE()` as first call within `RdE()`. Gain a mark for dealing with `eof` correctly.