コンパイラレポート

手続き4と5の解析

7JFC1121

佐藤圭一

目次

＊本文は段組の関係上、8points としました。

7JFC1121 佐藤圭一

＊ §1:手続きの概要

nextch 手続きは、ファイルポインタより、次の文字を取り出す手続きである。

Insymbol 手続きは、次の命令コードを解釈する手続きである。

Insymbol 手続き中で、nextch 手続きは呼び出される

§2:ソースコードの書式

ソースコードは、以下のような構成になっている。

Source_code

対訳の書式は、ネスト及び繰り返し等、入れ子の部分は「|」で表されており、繰り返し条件等は、人間が判断しやすい形に直してある。
つまり…

```
repeat
if a > b then
   begin
      b := b + 1;
   end;
   c := c + 1;
until d > c;
```

対訳

```
繰り返し開始
|a>b なら
||b に1足す
||
|c に1足す
|d<c なら繰り返す
```

の様になる。

7JFC1121 佐藤圭一

```
procedure nextch;
  begin

    if cc = ll then
      begin

        if eof(source) then
          begin
            writeln;
            writeln('program incomplete');
            goto 9999
          end;

        if errpos <> 0 then
          begin
            writeln;
            errpos := 0
          end;

        write('         ');
        ll := 0;

        while not eoln(source) do
          begin
            ll := ll + 1;
            read(source, inline[ll]);
            write(inline[ll])
          end;

        writeln;
        readln(source);
        ll := ll + 1;
        inline[ll] := ' ';
        cc := 0
      end;

    cc := cc + l;
    ch := inline[cc];
  end (* nextch *);
```

```
nextch 手続き


cc と ll が等しければ
|
|
|source が^Z だったら
||
||「コンパイル出来ない」と出力
||
||9999 に飛ぶ
|
|
|errpos が 0 以外なら
||
||改行出力
||errpos を 0 にする
||
|
|空白出力
||ll を 0 にする
|
|source が行末でなかったら
||
|||ll に 1 足す
||source から配列 inline[ll]に入れる
||inline[ll]を出力
||
|
|改行出力
|source を改行まで読む
|ll に 1 足す
|inline[ll]を初期化
|cc を 0 にする
|

cc に 1 足す
ch に配列 inline[cc]を入れる
nextch 終了
```


7JFC1121 佐藤圭一

```
procedure insymbol;                                          insymbol 手続き
   (*global - ch, sy, inum, id *)

   label l;                                                  l はラベルと宣言
   var i, j, k : integer;                                    I j k  を整数型で宣言

   begin

l:                                                           ラベル l はここです

      while ch = ' ' do                                      ch が空なら
         nextch;                                             |nextch を呼ぶ

      if ('0' <= ch) and (ch <= '9') then                    ch が 0〜9 の間なら
         begin                                               |
            sy := intliteral;                                |sy に intliteral を入れる
            inum := 0;                                       |inum を 0 にする
            k := 0;                                          |k を 0 にする
                                                             |
            repeat                                           |繰り返し開始
               inum := inum * 10 + ord(ch) - ord('0');       ||inum=inum*10+ord(ch)-ord('0')
               k := k + 1;                                   ||k に1足す
               nextch                                        ||nextch を呼ぶ
            until (ch < '0') or ('9' < ch);                  ||ch が 0〜9 の間なら繰り返す
                                                             |
            if k > dmax then                                 |k>dmax なら
               begin                                         ||
                  error(34);                                 ||error に引数 34 をつけて呼ぶ
                  inum := 0;                                 ||inum を 0 にする
                  k := 0;                                    ||k を 0 にする
               end                                           ||
         end                                                 |

         else                                                ch が 0〜9 以外なら
         if ('A' <= ch) and (ch >= 'z') then                 |ch が A〜z の間(英字)なら
            begin                                            ||
               id := '               ';                      ||id に空白を入れる
               k := 0;                                       ||k を 0 にする
                                                             ||
               repeat                                        ||繰り返し開始
                  if k < namel then                          |||k<namel なら
                     begin                                   ||||
                        k := k + 1;                          ||||k に1足す
                        id[k] := ch                          ||||ch から配列 id[k]に入れる
                     end;                                    ||||
                                                             |||
                  nextch                                     |||nextch を呼ぶ
               until not (ch in ['0'..'9', 'A'..'z']);       |||ch が英数字なら繰り返す
                                                             ||
               i := 1;                                       ||I を 1 にする
               j := nrw;                                     ||j に nrw を入れる
```


7JFC1121 佐藤圭一

```
   repeat
      k := (i + j) div 2;
      if id <= rw[k] then
         j := k - 1;

      if id >= rw[k] then
         i := k + 1

   until i < j;

   if i - 1 > j then
      sy := syrw[k]

   else
      sy := identifier

end

else
   case ch of
      ')', '*', '+', ',', '-', '.', ';', '=':
         begin
            sy := sysc[ch];
            nextch
         end;

      '(':
         begin
            nextch;

            if ch = '*' then
               begin
                  nextch;

                  repeat
                     while ch <> '*' do
                        nextch;
                     nextch
                  until ch = ')';

                  nextch;
                  goto l
               end

            else
               sy := sylparen
         end;
```

```
||繰り返し開始
|||k に(l+j) / 2 を入れる
|||もし id <= rw[k]なら
||||j に k-1 を入れる
|||
|||もし id >=rw[k]なら
||||l に k+1 を入れる
|||
|||j<l なら繰り返す
||
||i-1>j なら
|||sy に配列 syrw[k]を入れる
||
||l-1<=j なら
|||sy に identifier を入れる
||
||
|
ch が英字以外なら
|ch の内容が…
|| ｢)*+,-.;ｫ なら
|||
|||sy に配列 sysc[ch]を入れる
|||nextch を呼ぶ
|||
||
|| 「(  なら…
|||
|||nextch を呼ぶ
|||
|||ch が*なら
||||
||||nextch を呼ぶ
||||
||||繰り返し開始
|||||ch が*以外になるまで…
||||||nextch を呼び続ける
|||||nextch を呼ぶ
|||||ch が)以外なら繰り返す
||||
||||nextch を呼ぶ
||||l に飛ぶ
||||
|||
|||ch が*以外なら
||||sy に sylparen を入れる
|||
||
```


7JFC1121 佐藤圭一

```
            ':':
              begin
                nextch;
                if ch = '=' then
                  begin
                    sy := sybecomes;
                    nextch
                  end
                else
                  sy := sycolon
                end;

              '<':
                begin
                  nextch;
                    if ch = '=' then
                      begin
                        sy := syle;
                        nextch
                      end
                    else
                      if ch = '>' then
                        begin
                          sy := syne;
                          nextch
                        end
                      else
                        sy := syle
                  end;

               '>':
                  begin
                    nextch;
                    if ch = '=' then
                      begin
                        sy := syge;
                        nextch
                      end
                    else
                      sy := sygt
                  end;

               '!', '"', '#' ,'$' ,'%' ,'&' , '',
               '/', '?', '@', '[' , '¥', ']', '^',
               '_', '`', '{', '|':
                  begin
                    error(1);
                    nextch;
                    goto l
                  end
          end
end (* insymbol *);
```

```
|| 「』 なら…
|||
|||nextch を呼ぶ|
|||ch が=だったら
||||
||||sy に sybecomes を入れる
||||nextch を呼ぶ
||||
|||ch が=以外だったら
||||sy に sycolon を入れる
|||
||
|| 「<』 なら…
|||
|||nextch を呼ぶ
|||ch が=だったら
||||
||||sy に syle を入れる
||||nextch を呼ぶ
||||
|||ch が=以外だったら
||||ch が>だったら
|||||
|||||sy に syne を入れる
|||||nextch を呼ぶ
|||||
||||ch が>以外だったら
|||||sy い syle を入れる
|||
||
|| 「>』 なら…
|||
|||nextch を呼ぶ
|||ch が=だったら
||||
||||sy に syge を入れる
||||nextch を呼ぶ
||||
|||ch が=以外だったら
||||sy に sygt を入れる
|||
||
|| 「他の記号」 だったら…
||
||
|||
|||error に引数 1 をつけて呼ぶ
|||nextch を呼ぶ
||||l に飛ぶ
|||
||
insymbol 終了
```