

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

Page 3 §1:手続きの概要

Page 3 §2:ソースコードの書式

Page 4 §3:ソースコード

*本文は段組の関係上、8points としました。

コンパイラレポート-手続き 4 と5の解析

* §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

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;
```

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。

コンパイラレポート-手続き 4 と5の解析

```
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 + 1;  
        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
```

コンパイラレポート-手続き 4 と5の解析

procedure insymbol; (*global - ch, sy, inum, id *) label l; var i, j, k : integer; begin l: while ch = ' ' do nextch; if ('0' <= ch) and (ch <= '9') then begin sy := intliteral; inum := 0; k := 0; repeat inum := inum * 10 + ord(ch) - ord('0'); k := k + 1; nextch until (ch < '0') or ('9' < ch); if k > dmax then begin error(34); inum := 0; k := 0; end end else if ('A' <= ch) and (ch <= 'z') then begin id := ' '; k := 0; repeat if k < namel then begin k := k + 1; id[k] := ch end; nextch until not (ch in ['0'..'9', 'A'..'z']); i := 1; j := nrw;	insymbol 手続き l はラベルと宣言 l j k を整数型で宣言 ラベル l はここです ch が空なら nextch を呼ぶ ch が 0 ~ 9 の間なら sy に intliteral を入れる inum を 0 にする k を 0 にする 繰り返し開始 inum=inum*10+ord(ch)-ord('0') k に 1 足す nextch を呼ぶ ch が 0 ~ 9 の間なら繰り返す k>dmax なら error に引数 34 をつけて呼ぶ inum を 0 にする k を 0 にする ch が 0 ~ 9 以外なら ch が A ~ z の間(英字)なら id に空白を入れる k を 0 にする 繰り返し開始 k<namel なら k に 1 足す ch から配列 id[k]に入れる nextch を呼ぶ ch が英数字なら繰り返す i を 1 にする j に nrw を入れる
--	---

コンパイラレポート-手続き 4 と5の解析

```
repeat                                     ||繰り返し開始
  k := (i + j) div 2;                       ||k に(i+j) / 2を入れる
  if id <= rw[k] then                       ||もし id <= rw[k]なら
    j := k - 1;                             ||j にk-1を入れる
                                           ||
  if id >= rw[k] then                       ||もし id >=rw[k]なら
    i := k + 1                              ||i にk+1を入れる
                                           ||
until i < j;                               ||j<i なら繰り返す
                                           ||
if i - 1 > j then                          ||i-1>j なら
  sy := syrw[k]                            ||sy に配列 syrw[k]を入れる
                                           ||
else                                        ||i-1<=j なら
  sy := identifier                         ||sy に identifier を入れる
                                           ||
end                                         ||
                                           |
else                                       ch が英字以外なら
  case ch of                               |ch の内容が...
    ')', '*', '+', ',', '!', '-', '.', ':', '=', ':': ||'|'+, -, ., =, |' なら
      begin                                ||
        sy := sysc[ch];                   ||sy に配列 sysc[ch]を入れる
        nextch                             ||nextch を呼ぶ
      end;                                  ||
                                           ||
    '(':                                  ||'|(' なら...
      begin                                ||
        nextch;                           ||nextch を呼ぶ
                                           ||
        if ch = '*' then                  ||ch が*なら
          begin                            ||
            nextch;                       ||nextch を呼ぶ
                                           ||
            repeat                        ||繰り返し開始
              while ch <> '*' do          ||ch が*以外になるまで...
                nextch;                  ||nextch を呼び続ける
                nextch                    ||nextch を呼ぶ
              until ch = ')';            ||ch が)以外なら繰り返す
                                           ||
            nextch;                       ||nextch を呼ぶ
            goto l                         ||l に飛ぶ
          end                              ||
                                           ||
        else                               ||ch が*以外なら
          sy := sylparen                   ||sy に sylparen を入れる
                                           ||
        end;                              ||
                                           ||
```

コンパイラレポート-手続き 4 と5の解析

```

:::
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;
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に 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
|||に飛ぶ
|||
||
||
insymbol 終了

```