

平成28年秋 ソフトウェア コンパイラの字句解析と構文解析

問 2 コンパイラの字句解析と構文解析 (ソフトウェア) (H28 秋・FE 午後問 2)

【解答】  
[設問 1] aーア, bーイ  
[設問 2] cーウ, dーウ, eーウ

【解説】  
コンパイラ的字句解析と構文解析に関する問題である。一見、難解に思えるかもしれないが、内容は状態遷移図に関する設問と、2 分木に関する設問で、問題文をよく読み解答していく。

[設問 1]  
字句解析の処理について問われている。  
図 1「符号なし浮動小数点定数の構文規則に対する状態遷移図」の空欄を埋める問題である。

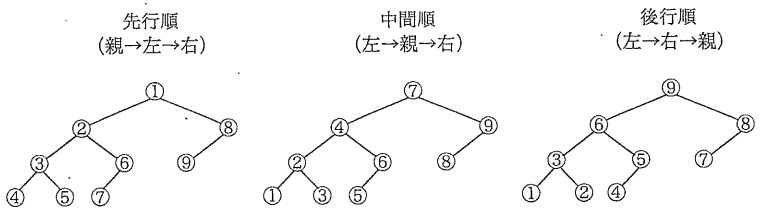
〔符号なし浮動小数点定数の構文規則〕には、次のように記述されている。  
符号なし浮動小数点定数 → 小数点定数 [指数部] | 数字列 指数部  
小数点定数 → [数字列]. 数字列 | 数字列.  
そこで、構文規則に沿って具体的な「符号なし浮動小数点定数」を幾つか考え、それを例として、図 1 に当てはめながら状態の遷移を考えてみる。  
例えば、「小数点定数 [指数部]」の指数部は省略可能なので、小数点定数である 0.1 や、0 を省略した .1 が候補になる。また、指数部をつけると 0.1e-1 も候補になる。また、「数字列 指数部」に注目すると、1e-1 など候補になる。  
これらを図 1 に当てはめて考えていく。状態遷移図の初期状態は 0、最終状態（終点）は二重円の 3 及び 6 であることに注意しながら、初期状態から文字の並びを 1 文字ずつ評価し、状態の遷移をトレースすれば、解答の糸口を見つけることができる。

・空欄 a：最初に 0.1 をトレースすると、0→1→3 で最終状態に遷移する。  
次に、.1 をトレースすると、最初は“.”であり、数字でないので、2 に遷移する必要がある。必然的に空欄 a は“.”となり、0→2→3 で最終状態に遷移する。したがって、(ア)の“.”が入る。

・空欄 b：0.1e-1 をトレースすると、0→1→3→4→5→6 で最終状態に遷移する。  
次に、1e-1 をトレースすると、0→1 のあとに“e”が出てくる。“.”ではないので、空欄 b を経て 4 に遷移する必要がある。必然的に空欄 b は“e”となり、後は 4→5→6 で最終状態に遷移することになる。したがって、(イ)の“e”が入る。

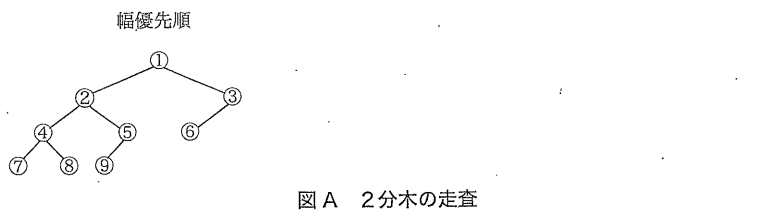
[設問 2]  
構文解析の処理の設問である。  
最初に、2 分木の走査（全ノードの数上げ）について確認しておく。2 分木の走査には、深さ優先（図 A①）と幅優先（図 A②）があり、深さ優先には、先行順（前順、行き掛け順）、中間順（間順）、後行順（後順、帰り掛け順）がある。

① 深さ優先順：根から始めて、左の子の方から、かつ葉の方から走査する方法で、木の深さ（縦）の方向に対して走査を繰り返す。



・先行順（前順、行き掛け順）：子（葉）よりも先に親（節）を先に走査する。  
・中間順（間順）：左の子を根とする部分木を先に走査してから節を走査し、次に右の子を根とする部分木を走査する。  
・後行順（後順、帰り掛け順）：左の子を根とする部分木を走査し、次に右の子を根とする部分木を走査、最後に節を走査する。

② 幅優先順：根から始めて、深さの浅い方からかつ左の方から走査を行う方法で、木の幅（横）の方向に対して走査を繰り返す。



この問題では、式を構成する演算子や名前などの字句を、式の左から右に読み込みながら、字句の並びが構文規則に合っているかどうかを解析し、それを図 2 のような 2 分木で表現する構文木として出力する。

〔演算順序〕に従って、式  $v \text{ op } w \text{ op } x$  を演算する。  
①  $v$  と  $w$  に対して演算  $\text{op}$  を施す。  
② ①の結果と  $x$  に対して演算  $\text{op}$  を施す。  
図 2 に分かりやすく追記したものが図 B である。点線で描いた①の三角形が、先に演算を行う箇所である。

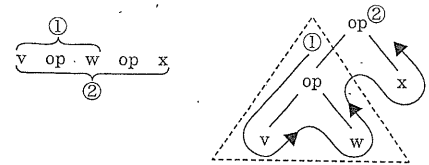


図 B 図 2 の演算木の演算順序

なお、後の構文規則では、括弧 ( ) をつけて優先度を表しているが、この式でも括弧 ( ) をつけて表現すれば、 $(v \text{ op } w) \text{ op } x$  となる。ちなみに、図 2 の構文木を、深さ優先の後行順（帰り掛け順）で、走査して、 $v \text{ w op } x \text{ op}$  の順番でスタックに格納（PUSH）すれば、プロセッサの演算向きのデータ構造（逆ポーランド記法）になる。また、構文木を中間順で走査すれば、元の式に戻る。これらを前提に、設問の空欄に入る答えを考える。

・空欄 c：〔式の構文規則〕の中の空欄である。〔式の構文規則〕には、 $\text{op1}$  は出てくるが、 $\text{op2}$  は出てこない。空欄 c に  $\text{op2}$  は必要なので、(ア)は正解ではないと判断できる。例 1 に (イ)～(エ)を当てはめて考えていく。

例 1： $v \text{ op2 } w \text{ op1 } x$   
イ：式  $\text{op2}$  因子……例 1 の後半部分の  $w \text{ op1 } x$  は演算子を含むので「式」、もしくは「項」であり、「因子」ではない。  
ウ：式  $\text{op2}$  項……「式」は、「式→項」、「項→因子」、「因子→名前」であり、例 1 の最初の  $v$  は「名前」であるので、間接的に「式」であるといえ、適切である。また、例 1 の後半部分の  $w \text{ op1 } x$  は、「項→項  $\text{op1}$  因子」であり、「項」である。したがって、「式 → 項 | 式  $\text{op2}$  項」となり、正しい。  
エ：式  $\text{op2}$  名前……例 1 の後半部分の  $w \text{ op1 } x$  は演算子を含むので「式」、もしくは「項」であり、「名前」ではない。

・空欄 d：括弧を含む場合の因子の構文規則である。  
「因子 → 名前」を括弧に対応させて「因子 → 名前 | ( d )」と定義を拡張している。ここで、例 2 を見る。

例 2： $v \text{ op2 } w \text{ op1 } (x \text{ op2 } y) \text{ op1 } z$   
括弧を含む式は  $(x \text{ op2 } y)$  となり、 $x$  は「名前」であるが、間接的に「式」にも当たる。また、 $y$  も「名前」であるが、間接的に「項」にも当たる。そして、 $\text{op2}$  を含むのは、空欄 c で説明したように「式 → 項 | 式  $\text{op2}$  項」のうち、「式→式  $\text{op2}$  項」に当たるため、「式」といえる。したがって、(ウ)が入る。

なお、(ア)の「因子」、(エ)の「名前」は、演算子を含まないので不正解である。また、(イ)の「項」の場合、 $\text{op1}$  しか扱えないので不正解である。

・空欄 e：例 2 の式を解析したときに、出力される構文木が問われている。例 2 の  $v \text{ op2 } w \text{ op1 } (x \text{ op2 } y) \text{ op1 } z$  は、 $\text{op1}$  が  $\text{op2}$  に比べ優先度が高いので、先に評価する必要がある。しかし、式の中に  $\text{op1}$  は二つあり、左から先に評価する必要があるが、優先順位が高い括弧の式  $(x \text{ op2 } y)$  にも注意する必要がある。  
そこで、 $(x \text{ op2 } y)$  を先に評価し、二つある  $\text{op1}$  は左から先に評価するので、演算の順番は図 C のとおりとなる。更に、①～④の演算順序を構文木の要素として組み立てていくと、図 D のようになり、(ウ)が正解である。

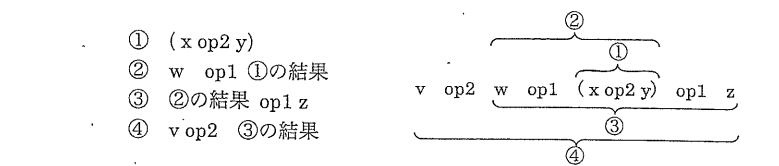


図 C 演算順序

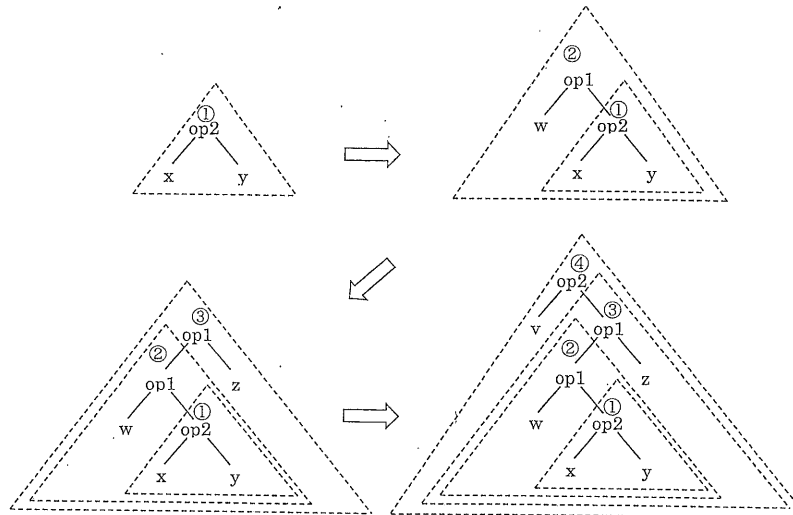


図 D 構文木の組立て

他の選択肢は次のようになり、不正解である。

ア： $v \text{ op2 } w \text{ op1 } (x \text{ op2 } y) \text{ op1 } z$   
イ： $(v \text{ op2 } w) \text{ op1 } ((x \text{ op2 } y) \text{ op1 } z)$   
エ： $(v \text{ op2 } (w \text{ op1 } (x \text{ op2 } y))) \text{ op1 } z$