問010052解説

◆解答

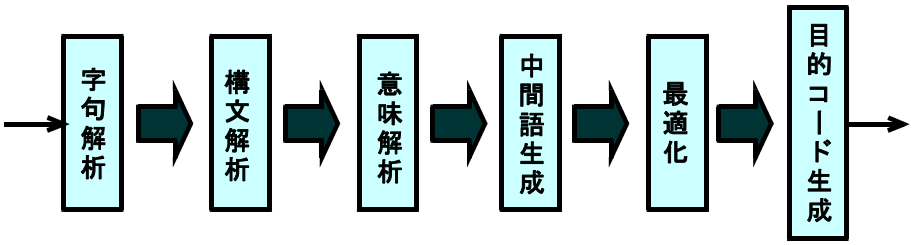
設問１　ａ　ア　　ｂ　イ

設問２　ｃ　ウ　　ｄ　ウ　　ｅ　ウ

◆解説

コンパイラの字句解析と構文解析に関する問題である。

①　コンパイラとは



コンパイラは、ある種のプログラム言語で記述されたソースプログラムモジュールの文字列をコンピュータ内部に読み込んで、２進化符号列のオブジェクトモジュールを出力する自動変換機能を提供する。字句解析、構文解析、意味解析、最適化、コード生成などのいくつかの処理単位別のフェーズから構成される。原始プログラムを字句解析で単語に分解し、文の構成を調べて中間言語を生成する。生成された中間言語は最適化された後、目的言語に変換される。

②　字句解析

字句は構文を構成する最小単位である。バッファに読み込まれた文字列は、字句またはトークンの生成規則に従って、順次文法上の意味のある語、すなわち字句という一つ一つのかたまりとして切り出されていく。これが字句解析である。字句解析では、正規表現を解析するプログラムが、ソースプログラムを読み込み、区切り記号に囲まれた構文要素を区別する。抽出された変数名、定数などは表にまとめられる。字句の生成規則、句の構文規則の表現には、正規表現が使用される。

③　字句の種類

㋐　名前　　：変数や手続きなどにつけられた参照のための字句である。英字で始まる英数字で構成される。

㋑　予約語　　：コンパイラが特別の意味に使うために、あらかじめ定めた字句である。予約語には宣言や制御構造などの構文を示すための字句やコンパイラが提供する標準の定数や関数などがある。

㋒　定数　　：リテラルと呼ばれる文字列や数字列の字句を定数と呼ぶ。

㋓　区切り記号：字句を切り出す際の目安となる文字を区切り記号という。区切り記号は、

スペース、カンマ、ピリオド、括弧、演算子などがある。

④　構文解析

構文解析は、字句の連なりとして抽出したものを、文法に照らし合わせて合法な文であるかどうかを判断することである。文の種類には、宣言文、代入文、構造文などがある。構文解析は、原始プログラムのどの範囲が、式、文、ブロック、手続きなど構文上のどの概念と対比しているかを定め、その情報を木構造に組み立て、原始プログラムが構文規則に従っているかを検査する。構文解析を行うプログラムをパーサまたはシンタックスアナライザという。パーサによって、中間語を生成するために必要な解析木または構文木を作り出し、各種の表を次のフェーズのために準備する。構文木は、文や字句の構造を木構造で表現したものである。構文木に三つ組、四つ組、逆ポーランド記法などの表現法がある。四つ組は、(演算子、被演算子１、被演算子２、結果)の形式で、被演算子１と被演算子２に演算子を作用させたものが結果であることを表す。

⑤　符号なし浮動小数点定数の構文規則

㋐　定義内容

ⓐ	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.
ⓒ	指数部	→	ｅ［符号］数字列
ⓓ	数字列	→	数字｜数字列　数字
ⓔ	符号	→	＋｜－
ⓕ	数字	→	０｜１｜２｜３｜４｜５｜６｜７｜８｜９

㋑　符号なし浮動小数点定数は、小数点定数または小数点定数　指数部または数字列　指数部のいずれかである。

㋒　小数点数定数の定義は、数字列. 数字列または. 数字列または数字列. のいずれかである。

㋓　指数部は、ｅ符号　数字列またはｅ　数字列のいずれかである。

㋔　数字列は、数字または数字列　数字のいずれかである。

⑥　状態遷移図で表現した構文規則

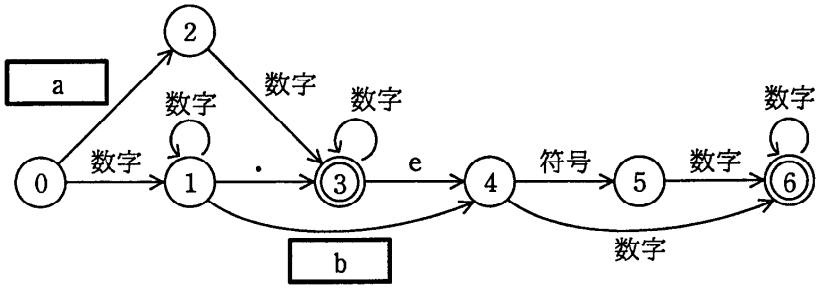


図１　符号なし浮動小数点定数の構文規則に対する状態遷移図

㋐　小数点定数は、数字列. 数字列または. 数字列または数字列. のいずれかであり、数字列. 数字列は０→①→③、. 数字列は０→②→③、数字列. は０→①→③でもとまる。

㋑　符号なし浮動小数点定数は、小数点定数または小数点定数 指数部または数字列 指数部のいずれかであるが、. 数字列 指数部は０→②→③→④→⑤→⑥で、数字列 指数部は０→①→④→⑤→⑥で、数字列. 指数部は０→①→③→④→⑤→⑥で求まる。

㋒　ａに入る答えは小数点「.」であり、ｂに入る答えは「ｅ」となる。

⑦　構文解析の処理

㋐　式の構文解析では，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で規定されている文法に合っているかどうかを解析し，その結果を構文木として出力する。

㋑　２項演算子op，名前 v，w，x を構成要素とする式 v op w op x、次の演算順序①，②になるように解釈され，その結果は，図２に示す２分木で表現する構文木として出力される。

〔演算順序〕

ⓐ　ｖとｗに対して演算 op を施す。

ⓑ　ⓐの結果とｘに対して演算 op を施す。

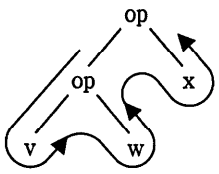


図２　式の構文木と演算順序の例

㋒　式の構文規則では，式の構文を規定するだけではなく，演算子の優先順位も規定する。２項演算子 op1 と op2，名前ｖ，ｗ，ｘ，ｙ，ｚを構成要素とする式の構文規則を定義する。ここで，“演算子 op1 の優先順位は，演算子 op2 の優先順位よりも高い”とする。これを規定する場合，式の構文規則は次のとおりになる。この構文規則で受理される式の例を，例１に示す。

〔式の構文規則〕

式　→　項 | [　c　]

項　→　因子 | 項 op1 因子

因子　→　名前

名前　→　ｖ | ｗ | ｘ | ｙ | ｚ

例１：v op2 w op1 x

㋓　さらに，式の構文に括弧を追加し，“括弧を含む式では，演算の優先順位は，括弧内の演算の方が高い”とする。これを規定する場合，因子の構文規則は次のとおりになる。この構文規則で受理される式の例を，例２に示す。

〔因子の構文規則〕

因子　→　名前 | （[　d　]）

例２：v op2 w op1 (x op2 y) op1 z

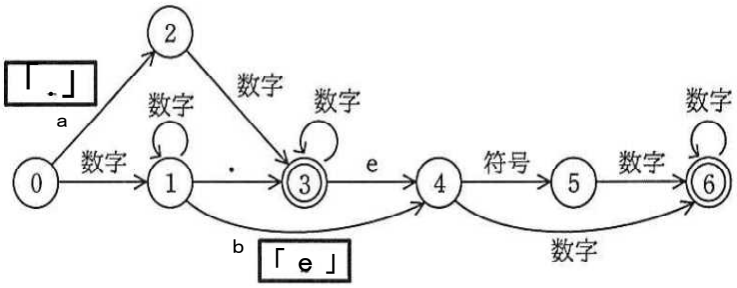
⑧　意味解析

意味解析は、変数名の未定義や二重定義、型の不一致など構文に含まれない制約を検査する。文字型として宣言された変数が算術式のなかに現れれば、意味上の誤りとして意味解析

の段階に検出される。意味解析では、構文解析の解析結果を解釈して、中間語のコードを生成する。中間語の生成には、逆ポーランド記法や四つ組または三つ組などの手法を利用する。

設問１

状態遷移図で表現した構文規則（ａ、ｂの答えを入力済み）



ａ、ｂの答えは、解説⑥の状態遷移図で表現した構文規則の㋐～㋒に説明しているように、０と２の矢印のａは「.」、１と４の矢印のｂは「ｅ」となる。従って、求める答えはａはア、ｂはイとなる。

設問２

ｃは、式を構成する構文規則であり、項をOPで結合しながら式を構成する内容になる。更に、複数ある演算子の優先順位を考慮して、式 OP2 項となる。求める答えはウとなる。

ｄは、式の構文に括弧を追加して、優先順位を高める構文規則の問題であり、例２に示すv op2 w op1 (x op2 y) op1 zの内容から､括弧の中には x op2 y の式が含まれることになる。従って、ｄは式となり、求める答えはウとなる。

ｅは、例２に示された例文を構文木で表した場合の結果を求める問題であり、例２の問題を次の構文木作成の手順に従った構文木を求めればよい。

例２：v op2 w op1 (x op2 y) op1 z

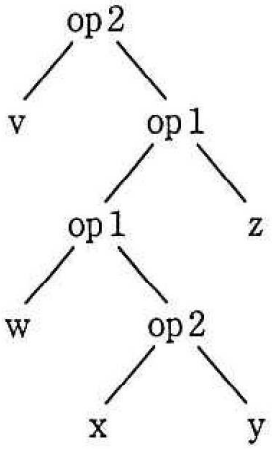
㋐　例２を左側から右に向かって順次名前を読んでいくと、最初は名前ｖとなり、続いて、ｗ、ｘ、ｙ、ｚの順に並ぶ。

㋑　次はOP2の後方にはOP1があり、この演算子の方が優先順位が高いため、先に演算を実行する。

㋒　更に、OP1の後方には括弧があり、この括弧の中の演算が先に実行されることになる。

㋓　括弧の後方には演算子OP1があり、この演算子は㋒のOP1の後で演算する。

㋔　㋓の演算子OP1が実行後に、㋑のOP2の演算が実行される。



㋐～㋔を構文木にまとめると右の図のようになる。求める答えはウとなる。