【１】形式言語　　　　　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

日本語や英語のように人が生活していくなかで自然に生まれた言語を**自然言語**といい、特定の目的のために人工的に作成された言語を**形式言語**と呼ぶ。

１．正規表現

**正規表現**とは、いくつかのパターンの文字列を、一定の形式で表現する方法の１つである。通常の文字と、メタキャラと呼ばれる特別な意味を持つ記号を組み合わせて表記する。

代表的なメタキャラには次のようなものがある。

｜ 　　いずれかの文字を選択する場合に使われる。

“tax|tex”はtaxまたはtexのことである。

( ) 　　まとめて表現する場合に使われる。

まず( )の中を処理し、その後( )の外の文字列を連結する。

“t(a|e)x”では、“(a|e)”の部分でaまたはeを選び、その後tとxを連結する。

？＊＋　　同じ文字を繰り返す場合に使われる。

？は直前の文字が、０個または１個、＊は０個以上、＋は１個以上あることを表す。

≪範例１≫

UNIXにおける正規表現［A－Z］＋［0－9］＊が表現する文字列の集合の要素となるものはどれか。ここで，正規表現は次の規則に従う。

［A－Z］は，大文字の英字１文字を表す。

［0－9］は，数字１文字を表す。

＋は，直前の正規表現の１回以上の繰返しであることを表す。

＊は，直前の正規表現の０回以上の繰返しであることを表す。

ア　456789 イ　ABC＋99 ウ　ABC99＊ エ　ABCDEF

≪解答≫　エ

ア　４５６７８９

文字列の先頭に、少なくとも英字が１文字なくてはなりません。

イ　ＡＢＣ＋９９

この文字は表現できる文字に含まれません。

ウ　ＡＢＣ９９＊

この文字は表現できる文字に含まれません。

２．BNF

**BNF**（Backus-Naur-Form）は、**バッカス記法**とも呼ばれ、バッカスがプログラム言語Algol60の構文規則の表記法として提案した。文字の並び方の規則を文字で定義し、反復や選択なども適当な文字記号で表現する。文字だけで定義するため、簡潔に表現でき、最終的な文の記述形式に近い表現になる。

BNFでは、数字と英字を次のように記述する。

＜数字＞::＝０|１|２|３|４|５|６|７|８|９

＜英字＞::＝ａ|ｂ|ｃ|…|ｚ|Ａ|Ｂ|Ｃ|…|Ｚ

“::＝”は、「左辺を右辺のように定義する」という意味であり、“|”は、「または」という意味となる。したがって、＜数字＞は、「“０～９”のいずれか一つで構成されている。」と解釈することができる。

１けた以上の＜数字＞で構成される＜識別子＞を定義するには、次の通りになる。

＜識別子＞::＝＜数字＞|＜識別子＞＜数字＞

≪範例２≫

次のBNFで定義されるビット列Ｓであるものはどれか。

＜Ｓ＞：：＝01 ｜ 0＜Ｓ＞1

ア　000111 イ　010010 ウ　010101 エ　011111

≪解答≫　ア

問題文でのＳの定義から、次のように表現できます。

＜Ｓ＞::＝01

＜Ｓ＞::＝0＜Ｓ＞1＝0(01)1＝0011

＜Ｓ＞::＝0＜Ｓ＞1＝0(0011)1＝000111

３．ポーランド記法

ポーランド記法には、前置記法の**ポーランド記法**と、後置記法の**逆ポーランド記法**がある。たとえば、

ｘ＝ａ＊ｂ－ｃ／ｄ

という算術式について２つの表記法を示すと、次の通りとなる。

ポーランド記法（前置記法） ：＝ｘ－＊ａｂ／ｃｄ

逆ポーランド記法（後置記法） ：ｘａｂ＊ｃｄ／－＝

通常の数式「ｅ＝(ａ－ｂ)÷(ｃ＋ｄ)」を逆ポーランド記法の数式に変換する場合。

(1) ｅ＝(ａ－ｂ)÷(ｃ＋ｄ)　　…ａ－ｂを逆ポーランド記法の数式へ→ａｂ－：これをＸとする。

(2) ｅ＝Ｘ÷(ｃ＋ｄ) 　　…ｃ＋ｄを逆ポーランド記法の数式へ→ｃｄ＋：これをＹとする。

(3) ｅ＝Ｘ÷Ｙ …Ｘ÷Ｙを逆ポーランド記法の数式へ→ＸＹ÷：これをＺとする。

(4) ｅ＝Ｚ　　　　　　　　 　…ｅ＝Ｚを逆ポーランド記法の数式へ→ｅＺ＝

(5) Ｘ，Ｙ，Ｚを逆ポーランド記法の数式に置き換える。（下線部は置き換えた部分）

ｅＺ＝　→　ｅＸＹ÷＝　→　ｅＸｃｄ＋÷＝　→　ｅａｂ－ｃｄ＋÷＝

通常の数式から逆ポーランド記法の数式への変換（例）

≪範例３≫

後置表記法(逆ポーランド表記法)では，例えば，式 Y＝(A－B)×Cを YAB－C×＝と表現する。次の式を後置表記法で表現したものはどれか。

Y＝（A＋B）×（C－D÷E）

ア　YAB＋C－DE÷×＝ 　　イ　YAB＋CDE÷－×＝

ウ　YAB＋EDC÷－×＝ 　　エ　YBA＋CD－E÷×＝

≪解答≫　イ

後置表記法（逆ポーランド表記法）は、２つの変数（オペランド）の後ろに演算子を記述する書き方であり、「A＋B」という計算式であれば「AB＋」と記述されます。

　①Y＝（A＋B）×（C－D÷E） →　Y＝AB＋×(C－DE÷)

　②Y＝AB＋×(C－DE÷) →　Y＝AB＋×CDE÷－

　③Y＝AB＋×CDE÷－ →　Y＝AB＋CDE÷－×

　④Y＝AB＋CDE÷－× →　YAB＋CDE÷－×＝

４．オートマトン

**オートマトン**は、コンピュータそのものを数学的観点からモデル化し、問題解決のためのアルゴリズム（処理手順）を定式化したものである。

このうち、有限個の状態と状態遷移からなるものを**有限オートマトン**と呼ぶ。有限オートマトンを表現した状態遷移図では、初期状態から、入力に応じて状態遷移を行い、一連の入力後、最終状態になれば入力が受理されたことになる。

≪範例４≫

図で表される有限オートマトンで受理される文字列はどれか。ここで，　　 　は初期状態を、◎は受理状態を表す。

ア　01011 　　イ　01111 　　ウ　10111 　　エ　11110

≪解答≫　ウ

問題の図に、下記のように状態に名前（ａ～ｅ）を追加します。

選択肢に与えられた文字列を有限オートマトンの図に当てはめ、受理状態（ｄ）で文字列が終了するかどうかを調べます。

ア　ａａｂａｂｃ

イ　ａａｂｃｄｅ

ウ　ａｂａｂｃｄ

エ　ａｂｃｄｅｅ

したがって、受理される文字列は、ウとなります。



ａ

ｂ

ｃ

ｄ

１

０

０

１

１

１

０

０

ｅ

０，１

【２】符号理論　　　　　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

符号理論は、情報を符号化する際の**効率化（情報の圧縮）**と信頼性の理論です。

情報（データ）量を減らすために、一部を省略することや、符号化する方法を変えることを情報の圧縮と呼ぶ。画像や音声のデータは、そのまま符号化すると莫大な情報量になってしまうので、情報の圧縮は通信回線や記憶媒体を効率良く使用するためには重要となる。

圧縮の代表的な方法には、ハフマン符号化とランレングス符号化がある。

１．ハフマン符号化

**ハフマン符号化**は、出現頻度がより高いデータに対してより短い符号を与えることによって、データ圧縮を行う方法である。

≪範例１≫

表は，文字Ａ～Ｅを符号化したときのビット表記と，それぞれの文字の出現確率を表したものである。１文字当たりの平均ビット数はいくらになるか。

|  |  |  |
| --- | --- | --- |
| 文字 | ビット表記 | 出現確率（％） |
| Ａ | 0 | 50 |
| Ｂ | 10 | 30 |
| Ｃ | 110 | 10 |
| Ｄ | 1110 | 5 |
| Ｅ | 1111 | 5 |

ア　1.6 　イ　1.8 　　ウ　2.5 　　エ　2.8

≪解答≫　イ

Ａは１ビット、Ｂは２ビット、Ｃは３ビット、Ｄ及びＥは４ビットで表記しています。１文字当たりの平均ビット数は、次のようになります。

平均ビット数 ＝１×0.5＋２×0.3＋３×0.1＋４×0.05＋４×0.05

＝1.8

２．ランレングス符号化

**ランレングス符号化**は、連続する同一の値を「データ×回数」という列（run）の長さ（length）を示す情報に置き換える方法である。たとえば、「AAAAABBBBCCCDDE」というデータ列を、Aが５回、Bが４回、Cが３回、Dが２回、Eが１回並んでいることから、「A5B4C3D2E1」というデータ列に置き換えることができる。こうして、もとのデータ列（15文字）から、置き換え後のデータ列（10文字）に圧縮する。

≪範例２≫

“連続する同一の文字コード（１バイトコードとする）の長さから１を減じたものを１バイトのバイナリで表し，その後に当該文字コードを配置する”というデータ圧縮方式がある。例えば，圧縮前に16進表示で，

41 41 41 41 41 42 43 43 43 43 43 43

であった12バイトの文字コードの列は，圧縮後に，

04 41 00 42 05 43

という６バイトで表され，この場合の圧縮率は50%（6バイト÷12バイト×100）となるものとする。このとき，当該方式に関する記述のうち，適切なものはどれか。

ア　10個の文字からなる文字列を圧縮したとき，最良の場合の圧縮率は最悪の場合の圧縮率の５分の１である。

イ　圧縮後の長さが圧縮前の長さを上回ることはない。

ウ　一度に256バイト(256の同じ文字)を２バイトに圧縮できるときが最大の圧縮率なので，圧縮率が0.7%以下の値になることはない。

エ　文字列に２回圧縮を行うと１回圧縮を行う場合の２分の１の圧縮率となる。

≪解答≫　ウ

最大の圧縮率は、2バイト÷256バイト×100＝0.78125％

となり、0.7％を下回ることはありません。

ア　10個の文字からなる文字列を圧縮したとき、最良とは10個の文字がすべて同じ場合です。このときの圧縮率は、2バイト÷10バイト×100＝20％となります。

逆に最悪は10個の文字がすべて異なる場合です。このときの圧縮率は、20バイト÷10バイト＝200％となります。したがって、最良の圧縮率は最大の圧縮率の10分の1です。

イ　同じ文字が連続していない場合には、圧縮後の長さが圧縮前の長さを上回ることがあります。

エ　問題にある例で、2回圧縮した場合を考えます。

41 41 41 41 41 42 43 43 43 43 43 43　(12byte)

1回圧縮すると、04 41 00 42 05 43　(6byte)

このときの圧縮率は、6÷12×100＝50％

2回目の圧縮を行うと、00 04 00 41 00 00 00 42 00 05 00 43　（12byte）

このときの圧縮率は、12÷6×100＝200％

となり、1回圧縮したときの圧縮率の2分の1となっていません。