

# 逆ポーランド記法

# 数式の表記法

## ① 前置記法(ポーランド記法)

- ① 前置記法は、  
演算子をオペランドの前に記述する記法である。
- ② 具体例  
 $1 + 2 \rightarrow + 1 2$
- ③ 二分木ならいの先行順を使用して求めることができる。

## ② 中置記法

- ① 中置記法は、二つのオペランドに対して  
その間に演算子を置く、  
数式やプログラムを記述する方法である。
- ② 具体例  $1+2$ などの通常の数式の記述方法である。
- ③ 二分木ならいの中間順を使用して求めることができる。

## ③ 後置記法(逆ポーランド記法)

- ① 後置記法は、演算子をオペランドの後に  
記述する記法である。
- ② 具体例  $1+2 \rightarrow 12+$
- ③ 二分木ならいの後行順を使用して求めることができる。

# 後置記法のための二分木の作成

- ① 四則演算式を逆ポーランド記法に変換する場合  
二分木のならいの後行順を使用する。
- ② 次の手順で二分木を作成する。
  - ① 変数を葉に配置する。
  - ② 演算子を節に配置する。
  - ③ 四則演算の優先順位が成立するように  
各節間の枝を設ける。

# 具 体 例

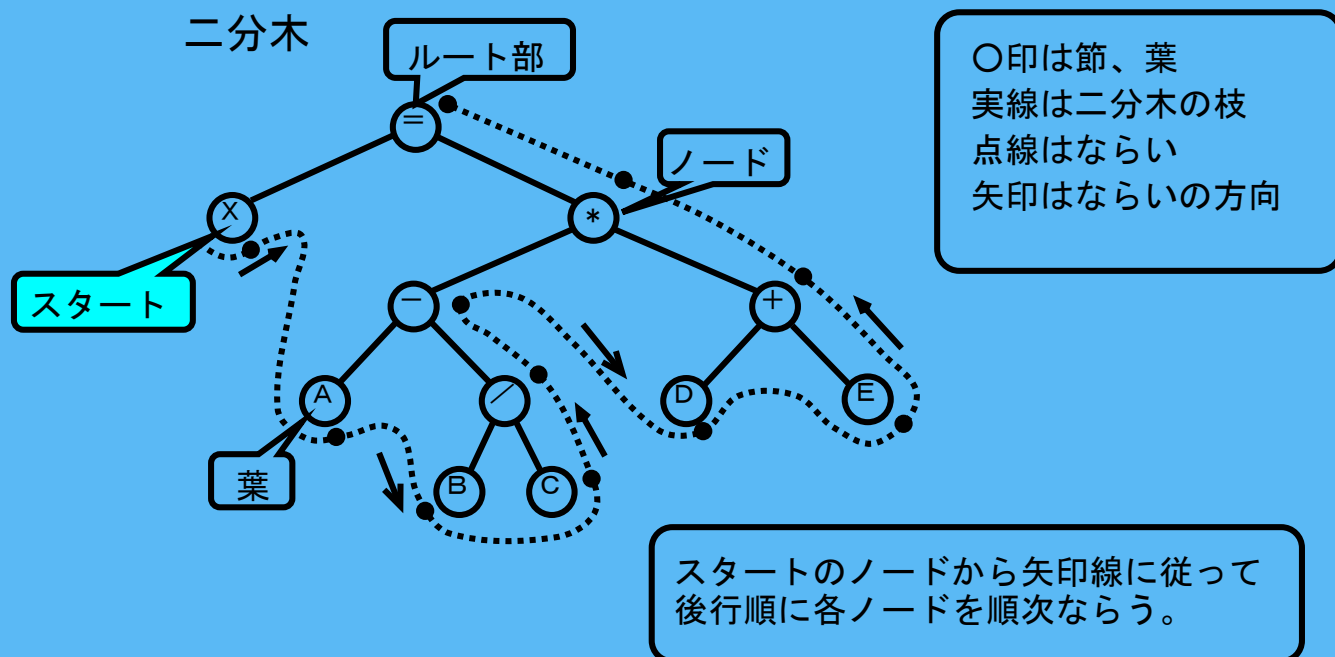
$X = (A - B / C) * (D + E)$ の  
四則演算は次の手順で二分木を作成し、  
逆ポーランド記法を求める。

- ①  $=$ を根に配置する。
- ②  $X$ を根の左側の葉に配置する。
- ③  $A$ 、 $B$ 、 $C$ 、 $D$ 、 $E$ を右部分木の葉に配置する。
- ④ 四則演算子の優先レベルの高いものから  
演算子で結合する。
- ⑤ 完成した二分木に後行順のならいを適用すると、  
 $XABC / - DE + * =$   
の逆ポーランド記法を求めることができる。

# 後置記法の求め方

- ① 作成した二分木に  
後行順のならいを使用して  
後置記法(逆ポーランド記法)を求めることができる。
- ②  $X = (A - B / C) * (D + E)$  の四則演算から  
二分木を作成すると、次ページの図のようになる。
- ③ この作成した二分木に後行順のならいを使用し、  
図の矢印に従ってならうと、  
次の後置記法を求めることができる。  
 $XABC / - DE + * =$

## ④ 後行順ならいの図





# スタックを利用した四則演算

逆ポーランド記法を  
左から順に解析する。

- ① 数値であればスタックに格納する。
- ② 演算子であれば、スタックから2つの数値を取り出す
- ③ 取り出した2つの数値で、次の演算を行う。  
(後に取り出した数値)演算子(先に取り出した数値)
- ④ 演算結果をスタックに格納する。



# 演算のアルゴリズム

- ① スタックポインタを初期化する。
- ② 先頭から＝が表れるまで、③、④の処理を繰り返す。
- ③ 数値ならば、スタックにPUSHする。
- ④ 演算記号ならば、次の処理を行う。
  - ① スタックから数値を2つPOP する。
  - ② 取り出した2数の演算を行う。
  - ③ 演算結果をスタックにpushする。
- ⑤ ＝の時演算結果をPOP して処理を終了する。

# 具体例

元の四則演算  $X = (A - B / C) * (D + E)$

逆ポーランド記法  $XABC / - DE + * =$

演算手順

- ① X、A、B、Cの順にスタックにPUSHする。

- ② 次は演算子の $\div$ であるから、  
C、BをPOPLし、 $B \div C$ を計算し、PUSHする。
- ③ 演算子 $-$ であるから、  
 $B \div C$ 、AをPOPLし、 $A - B \div C$ を計算し、PUSHする。
- ④ D、Eの順にスタックにPUSHする。
- ⑤ 演算子 $+$ であるから、  
E、DをPOPLし、 $D + E$ を計算し、PUSHする。

⑥ 演算子 \* であるから、  
D+E、A-B/CをPOPLし、  
(A-B/C)\*(D+E)計算し、PUSHする。

⑦ 演算子 = であるから、  
(A-B/C)\*(D+E)、XをPOPLし、  
次の答えを求める。

$$X = (A - B / C) * (D + E)$$

## ⑧ スタック操作の図

