１５章　アルゴリズムとプログラミング

計算機の本質としてのプログラム (p158)

- 電卓
  - データやプログラムを記憶する装置がない
- 計算機(computer)　＝　計算するもの
  - プログラムが使える
- プログラム
  - 計算（仕事）の手順を書いたもの

連接，繰返し，条件分岐(p159)

- 3つの構成要素
  - どんなに複雑なプログラムも，連接，繰返し，条件分岐で記述できる。
- 連接
  - {文1； 文2； … …； 文n；}
- 繰返し
  - {条件Cが成立しているかぎり文1の実行を繰り返す}
- 条件分岐
  - {条件Cが成立していれば文1を実行し，成立していなければ文2を実行する}

埋め込み例1

- 埋め込み
  - {条件Cが成立しているかぎり文1の実行を繰り返す}
  - {条件Cが成立しているかぎり
    {文1; 文2; ……; 文n;}
    の実行を繰り返す}

埋め込み例2

- 埋め込み
  - {条件Cが成立しているかぎり
    {文1； 文2； … …； 文n；}
    の実行を繰り返す}
  - {条件Cが成立しているかぎり
    {文1；
    {条件Cが成立していれば文1を実行し，
    成立していなければ文2を実行する}
    … …；
    文n；}
    の実行を繰り返す}

抽象化

- 埋め込み
  - 入り口一つ、出口一つ (構造化プログラミング)
  - いくらでも複雑なプログラムを作ることが可能
- 大規模なプログラム
  - 連接、繰返し、条件分岐だけでは開発困難
  - 抽象化が必要

スタックと日本語の語順(p160)

- オブジェクト指向
  - プログラムはさまざまなオブジェクト（もの）の集まり。
  - オブジェクトの機能は、外部からわかる
  - オブジェクトの内側での働きは、わからない

オブジェクトの例としてのスタック (p163)

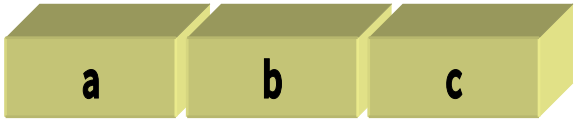
- スタックを使った文字列を反転するプログラム

{文字列の反転は
{文字を入れるスタックを構成し
文字を入力してから
以下のことを入力文字列の終わりがくるまで繰り返す
{入力した文字列をスタックに push する
文字を入力する}
以下のことをスタックが空になるまで繰り返す
{スタックの内容を pop する
pop した文字を出力する}
}
ことで実現できる}

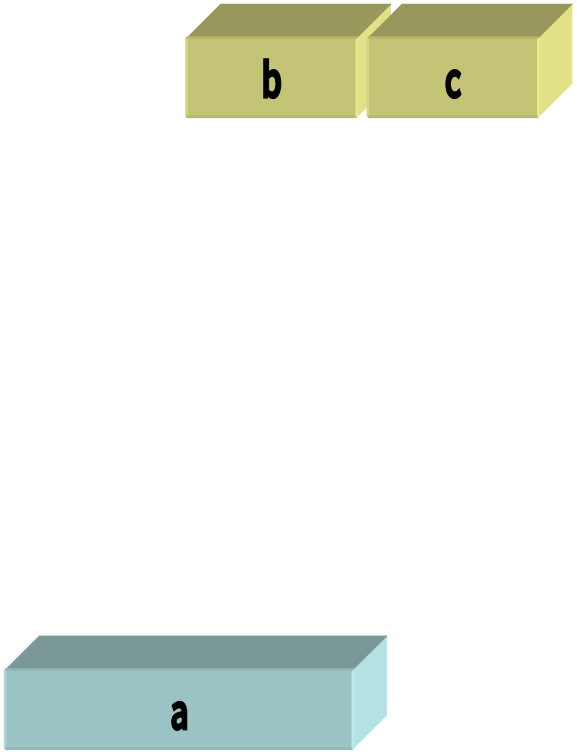
クラスの定義(p164)

- スタッククラス
  {スタックとは
    {スタック配列
     スタックトップ}
  という属性から成り、
    {スタックを構成する
     数値を push する
     数値を pop する
     スタックが空になっている}
  という機能を有するメソッドをもつ}

スタックの動作（文字列反転）

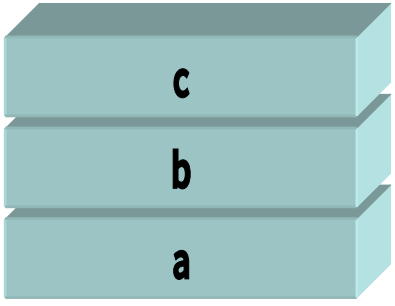


スタックの動作（文字列反転）



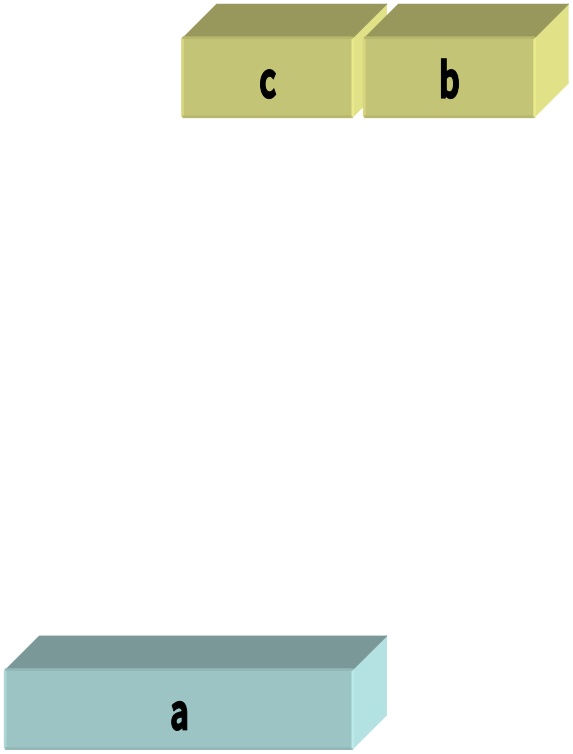
スタックの動作（文字列反転）

スタックの動作（文字列反転）

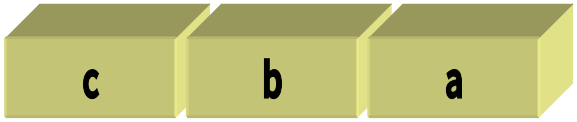


スタックの動作（文字列反転）

スタックの動作 (文字列反転)



スタックの動作（文字列反転）



電卓の例(p161)

- ＨＰ(Hewlett Packard)社の電卓
  - ａ＊（ｂ＋ｃ／ｄ）
  - ｃ／ｄを計算し，
    それにｂを足し，
    その結果にａを掛ける
- 計算機
  - 最初はアセンブラでプログラムを記述
  - 大域的な処理は、苦手

高級言語の出現(p162)

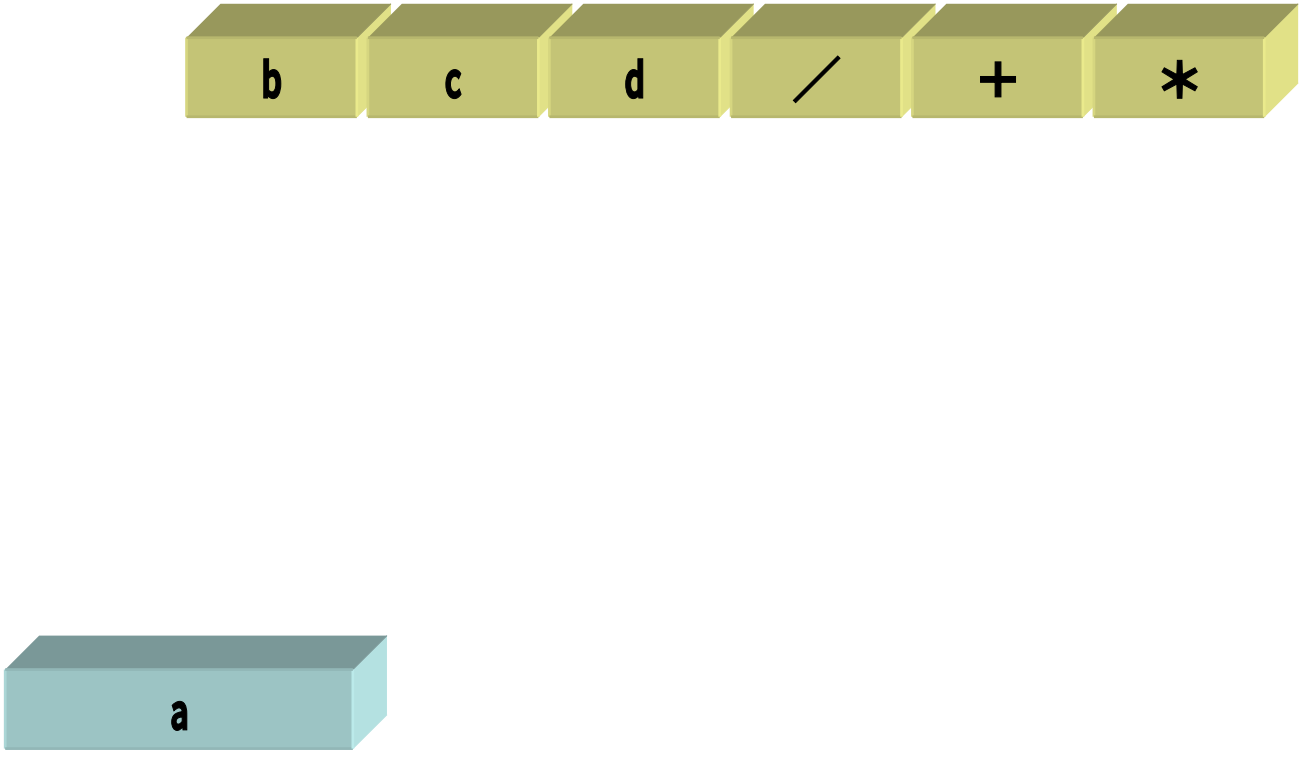
- 高級言語
  - FORTRAN、COBOL、BASIC等
- 中置形式(infix notation)
  - a＋b
  - a＊(b＋c／d)
- 後置形式(postfix notation)
  - ab＋
  - abcd／＋＊

スタック(p162)

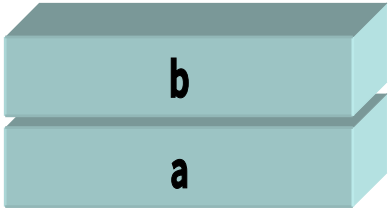
- 記憶装置
  - push (スタックに数値を積み上げる操作)
  - pop (スタックから数値を取り出す操作)
  - 演算 (スタックの最上部の2つの数値に対して実施)

スタックの動作

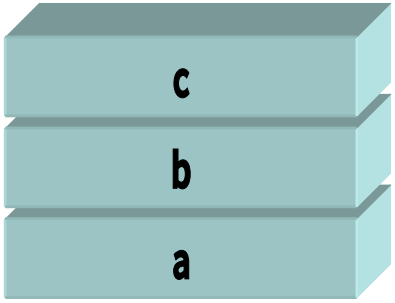
スタックの動作



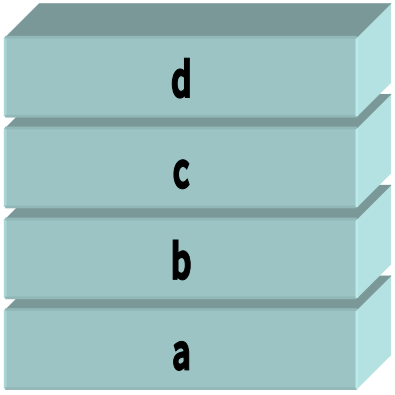
スタックの動作



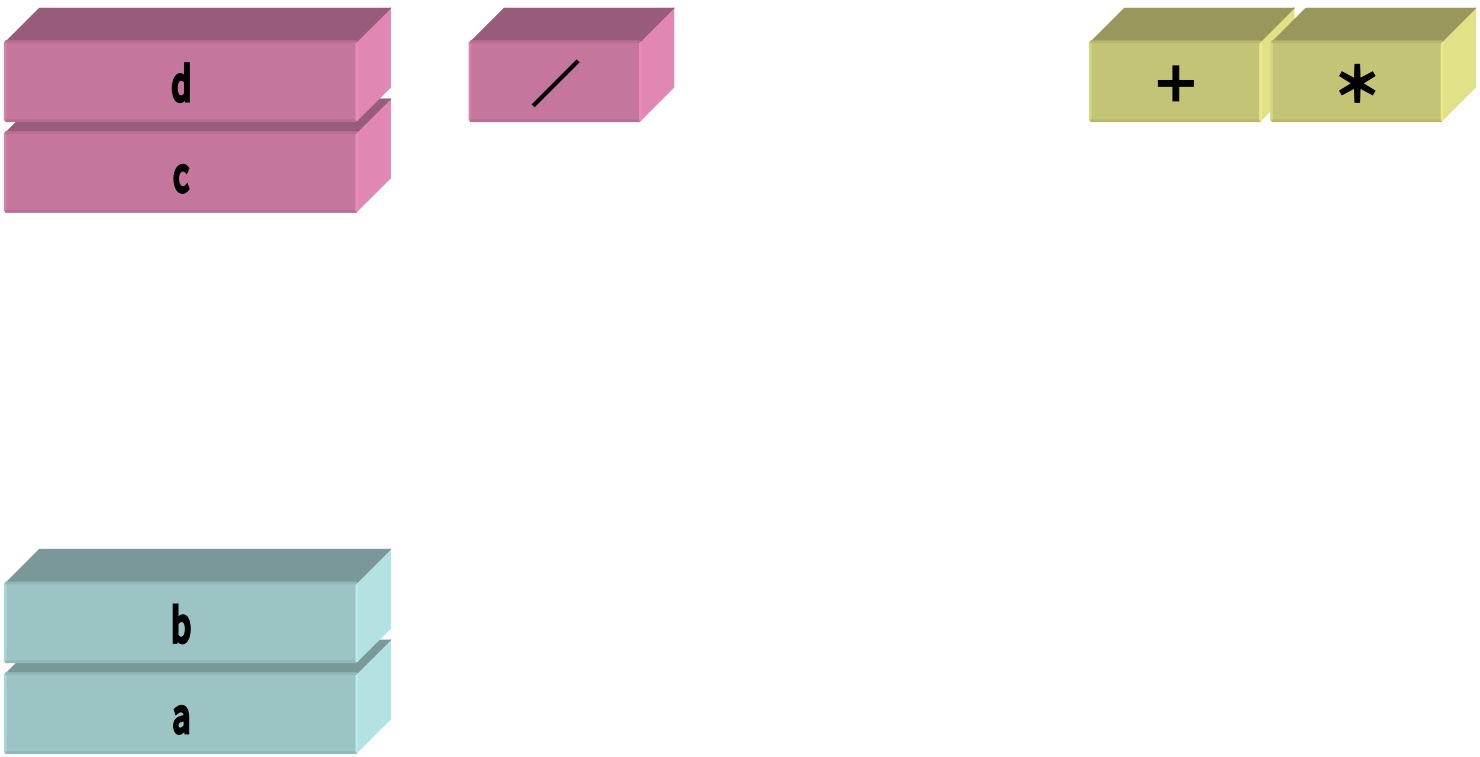
スタックの動作



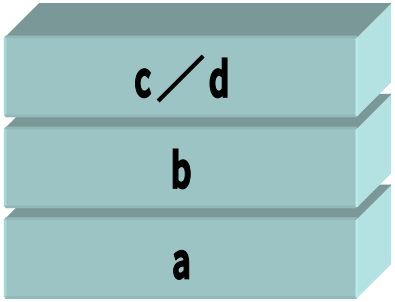
スタックの動作



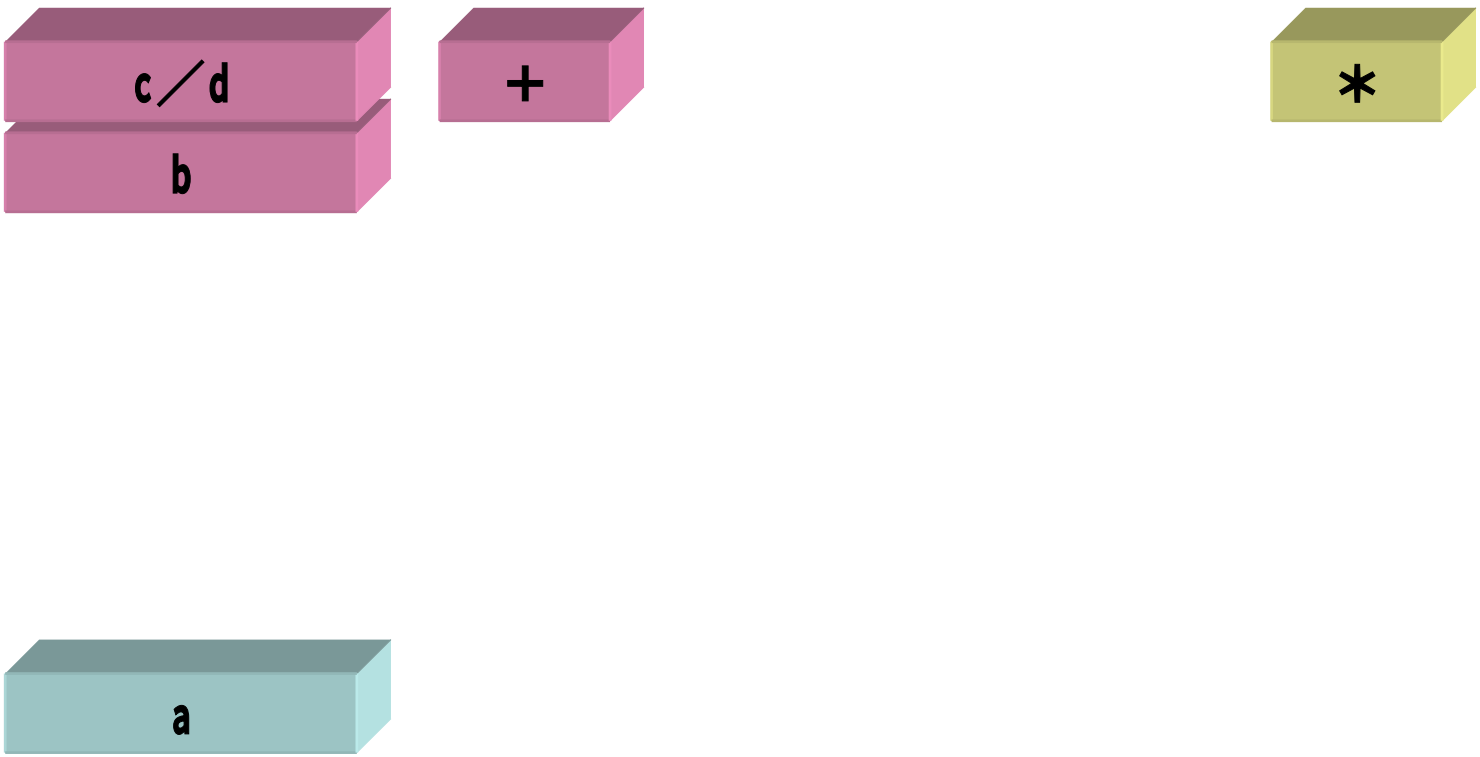
スタックの動作



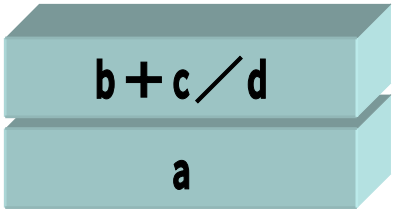
スタックの動作



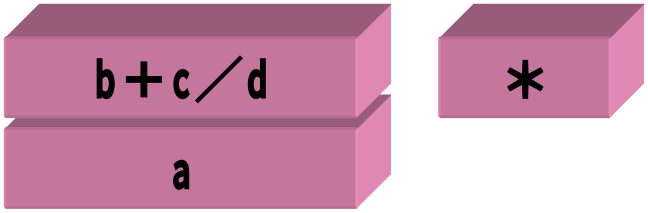
スタックの動作



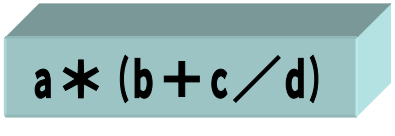
スタックの動作



スタックの動作



スタックの動作



HP社の電卓のシミュレート(p164)

- **レジスタの構成**
  - **4つのレジスタが存在**

t

z

y

x

電源投入直後

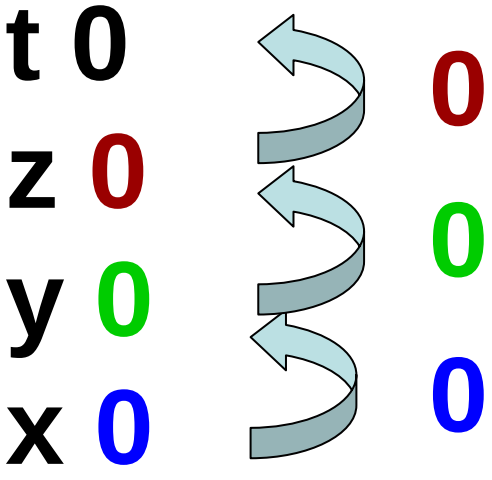
t 0

z 0

y 0

x 0

12enter33+21*



12enter33+21*

t 0

z 0

y 0

x 1

12enter33+21*

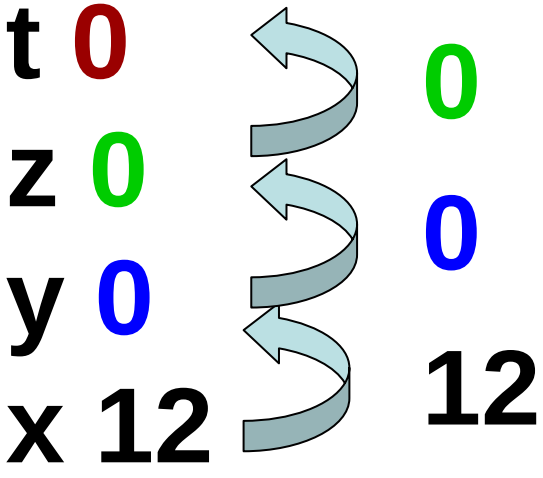
t 0

z 0

y 0

x 12

12enter33+21*



12enter33+21*

t 0

z 0

y 12

x 12

12enter33+21*

t 0

z 0

y 12

x 3

12enter33+21*

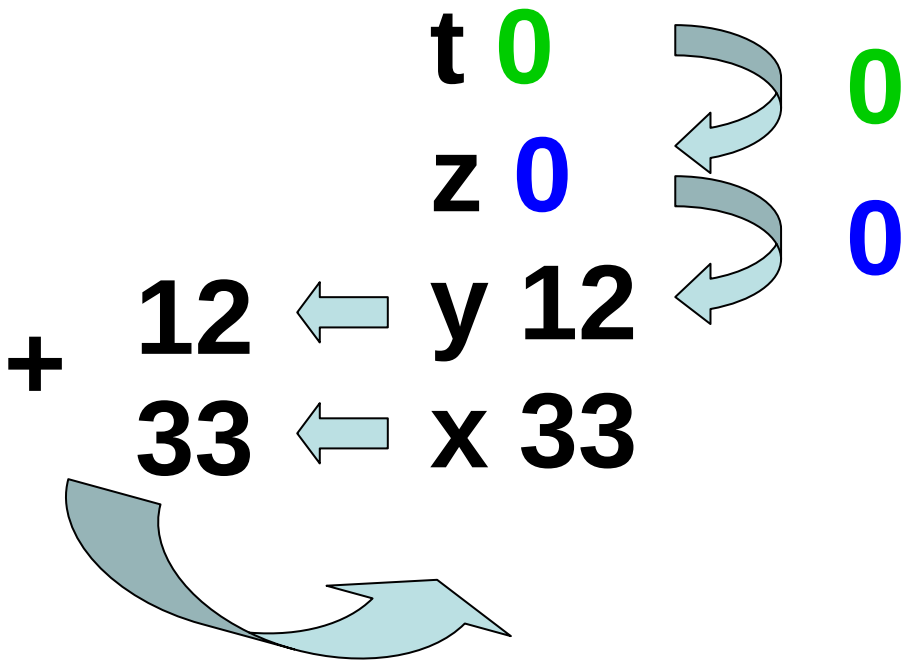
t 0

z 0

y 12

x 33

12enter33+21*



12enter33+21*

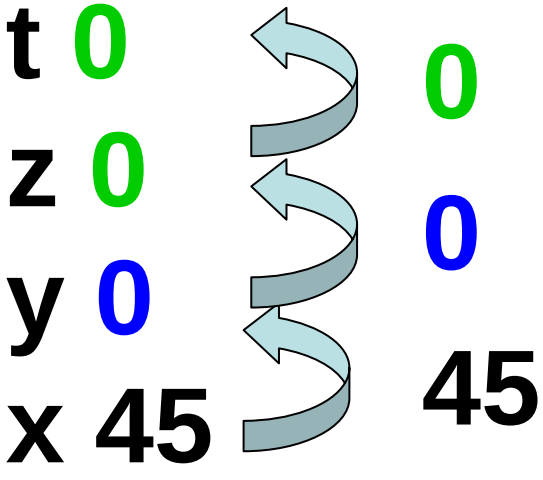
t 0

z 0

y 0

x 45

12enter33+21*



12enter33+21*

t 0

z 0

y 45

x 2

12enter33+21*

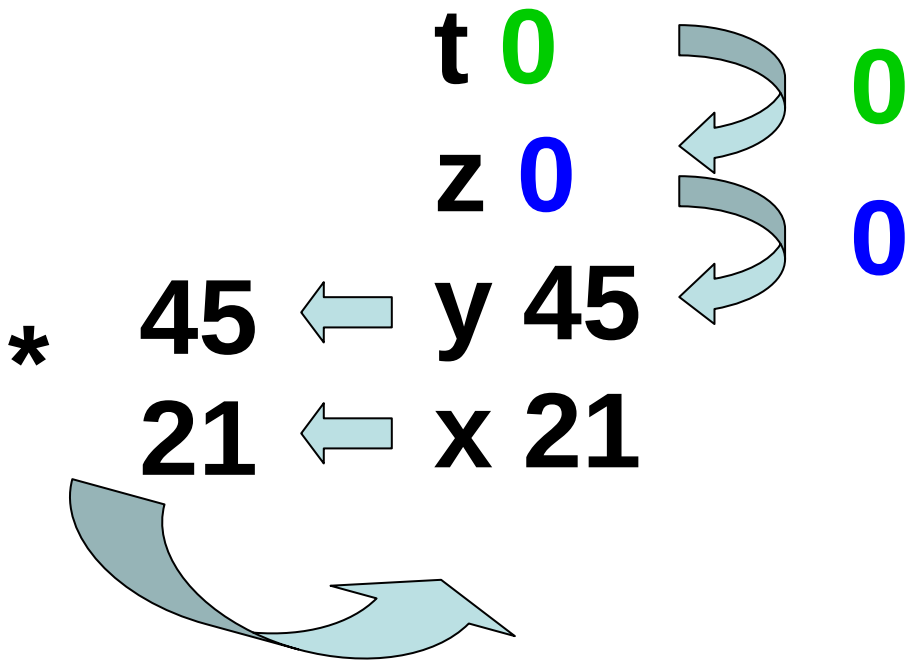
t 0

z 0

y 45

x 21

12enter33+21*



12enter33+21*

t 0

z 0

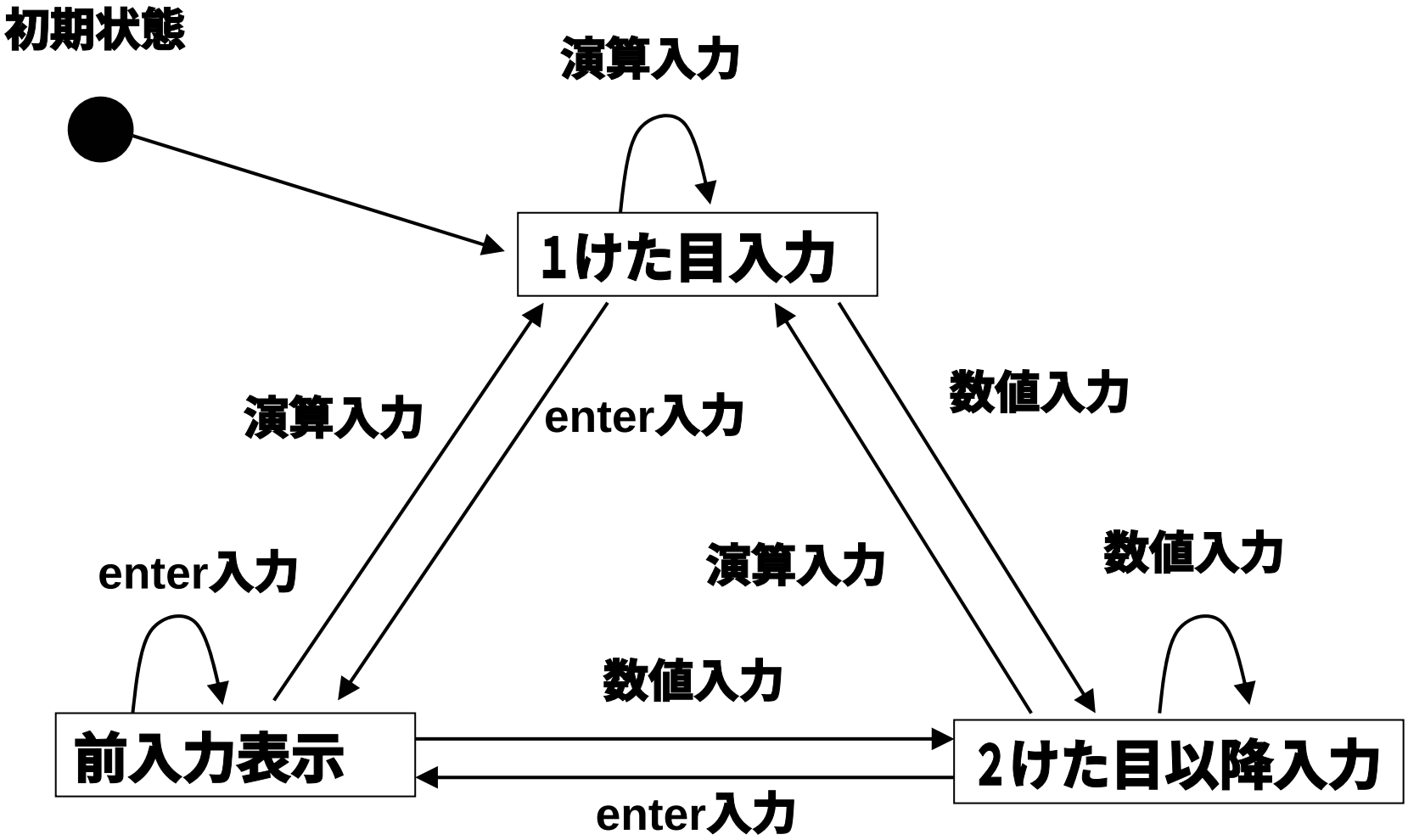
y 0

x 945

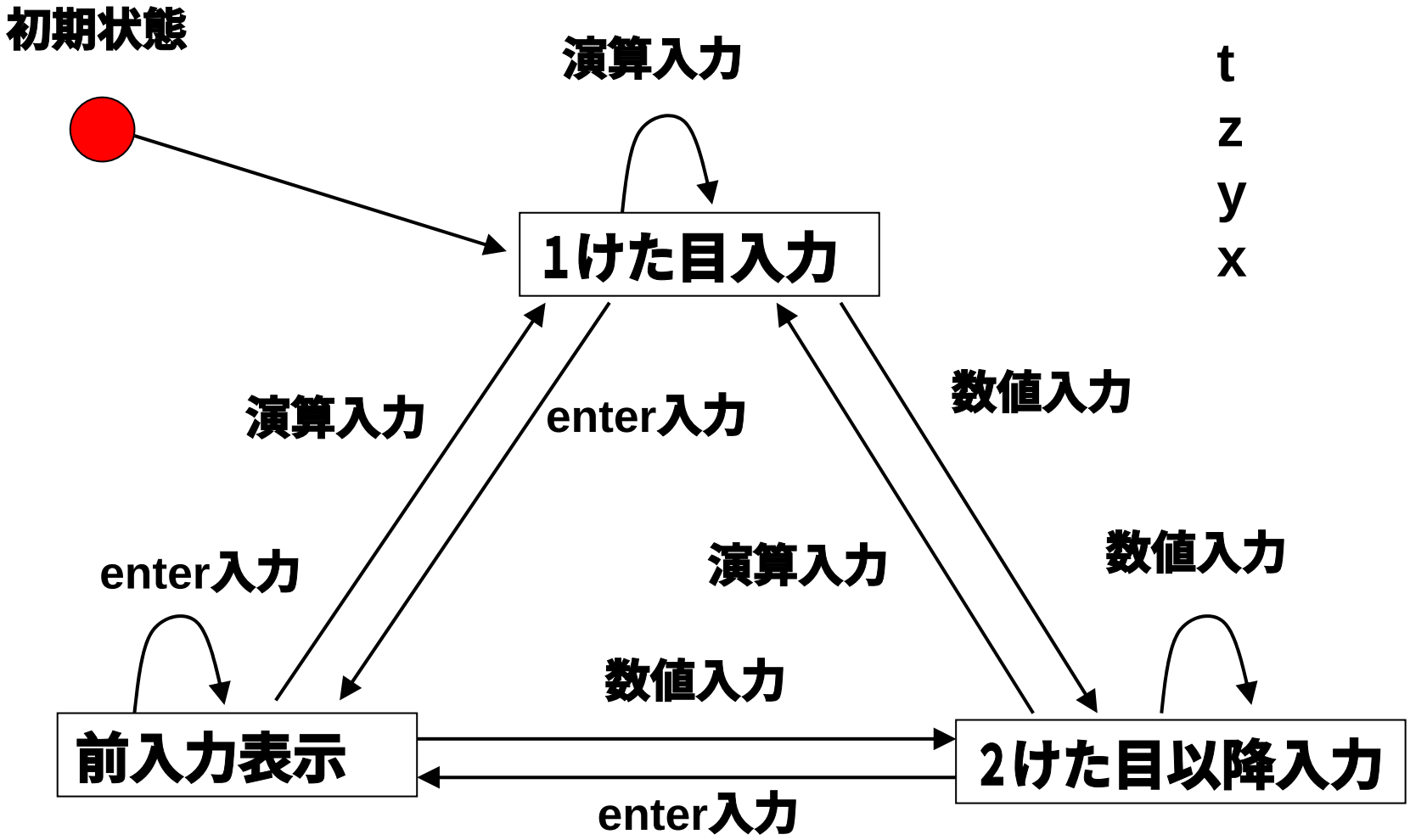
UML

- Unified Modeling Language
  - ユースケース図 (要件定義)
  - クラス図 (基本構造)
  - シーケンス図 (時間的な動きを表現)
  - コラボレーション図 (時間的な動きを表現)
  - ステートチャート図 (状態遷移図)
  - コンポーネント図 (物理的な構成)
  - アクティビティ図 (活動の流れ)
  - 配置図 (コンポーネントのシステム内での配置)
  - その他

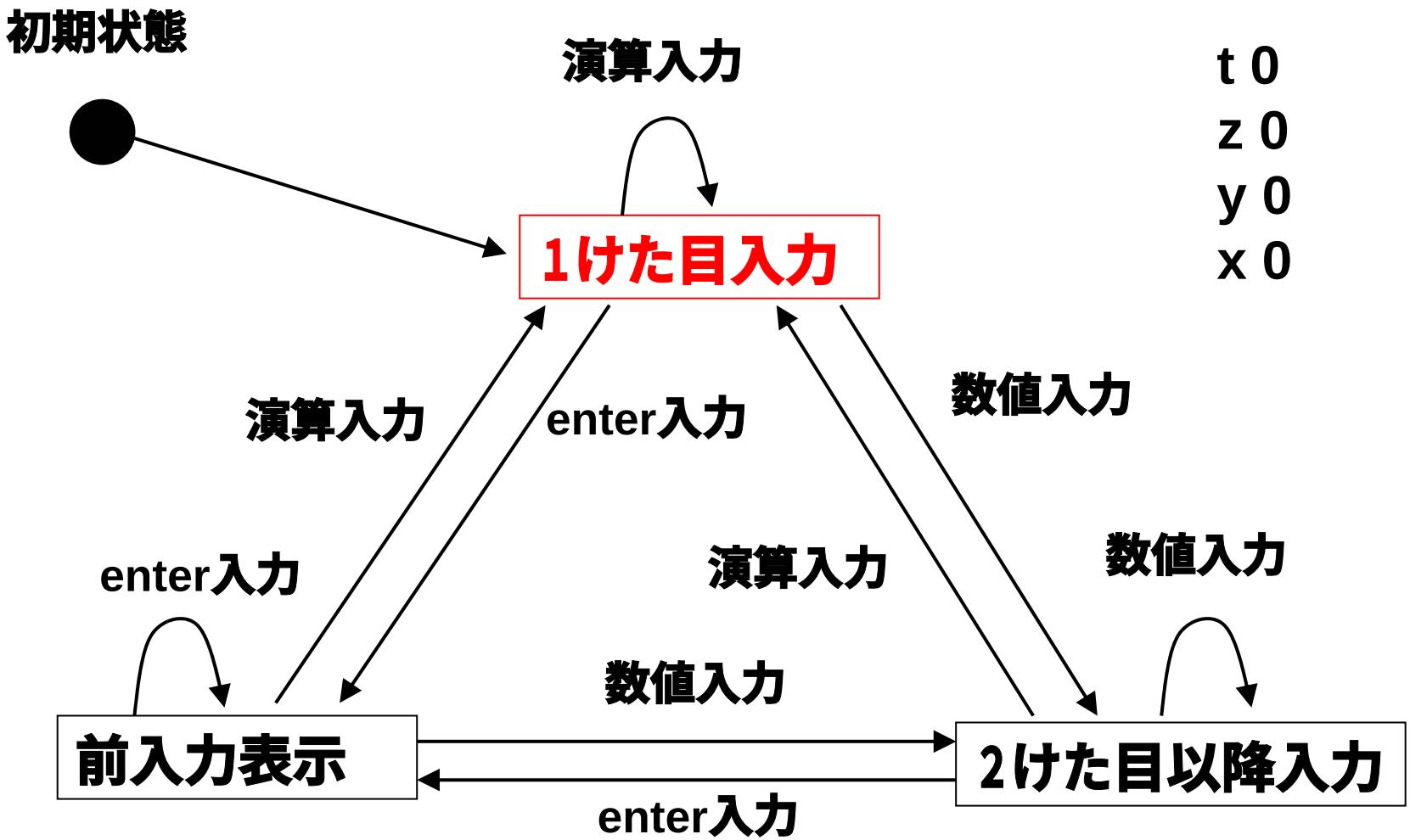
HP社の電卓の状態遷移図(p168)



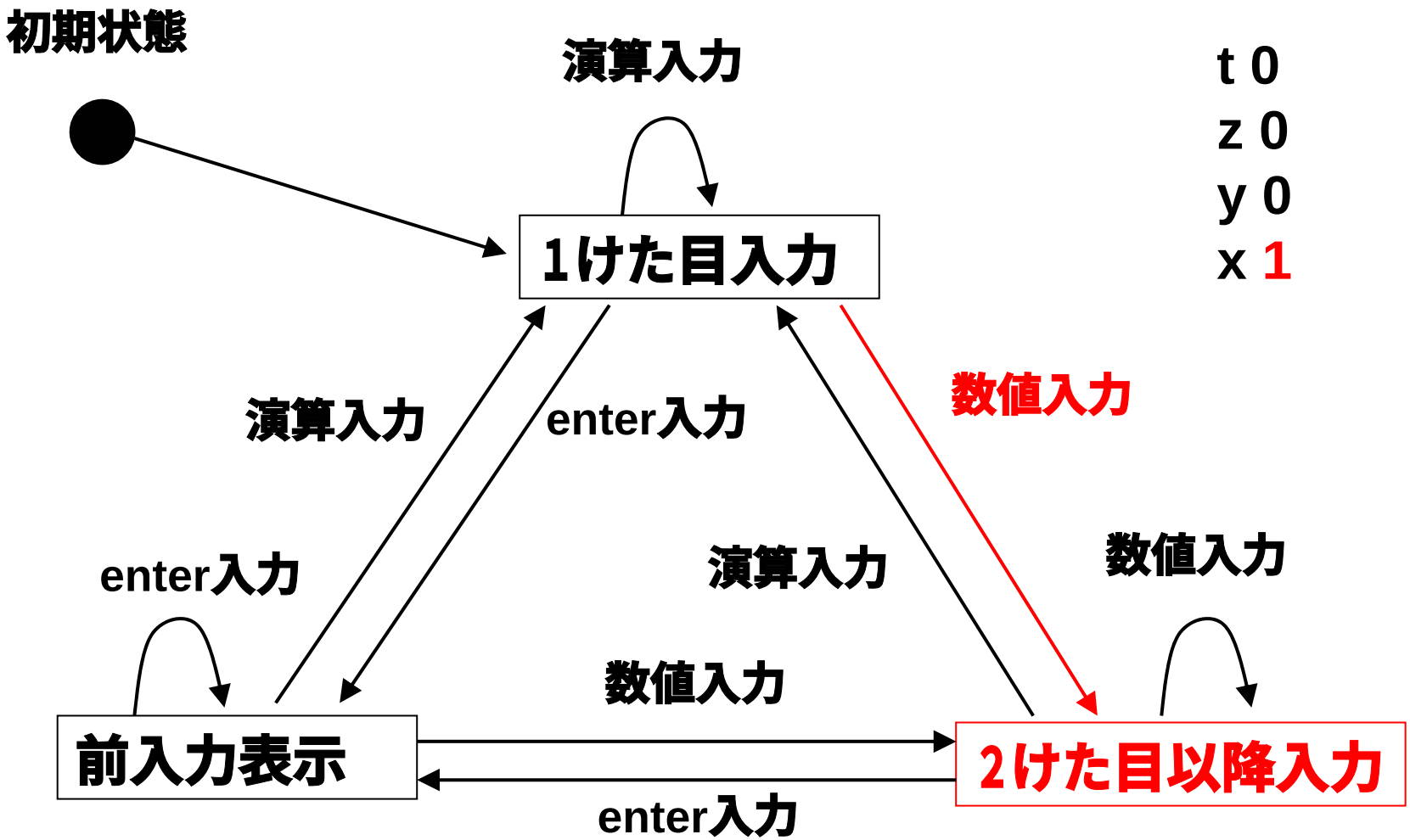
電源投入前


初期状態
演算入力
t
z
y
x
1けた目入力
演算入力
enter入力
数値入力
enter入力
演算入力
数値入力
数値入力
前入力表示
2けた目以降入力
enter入力

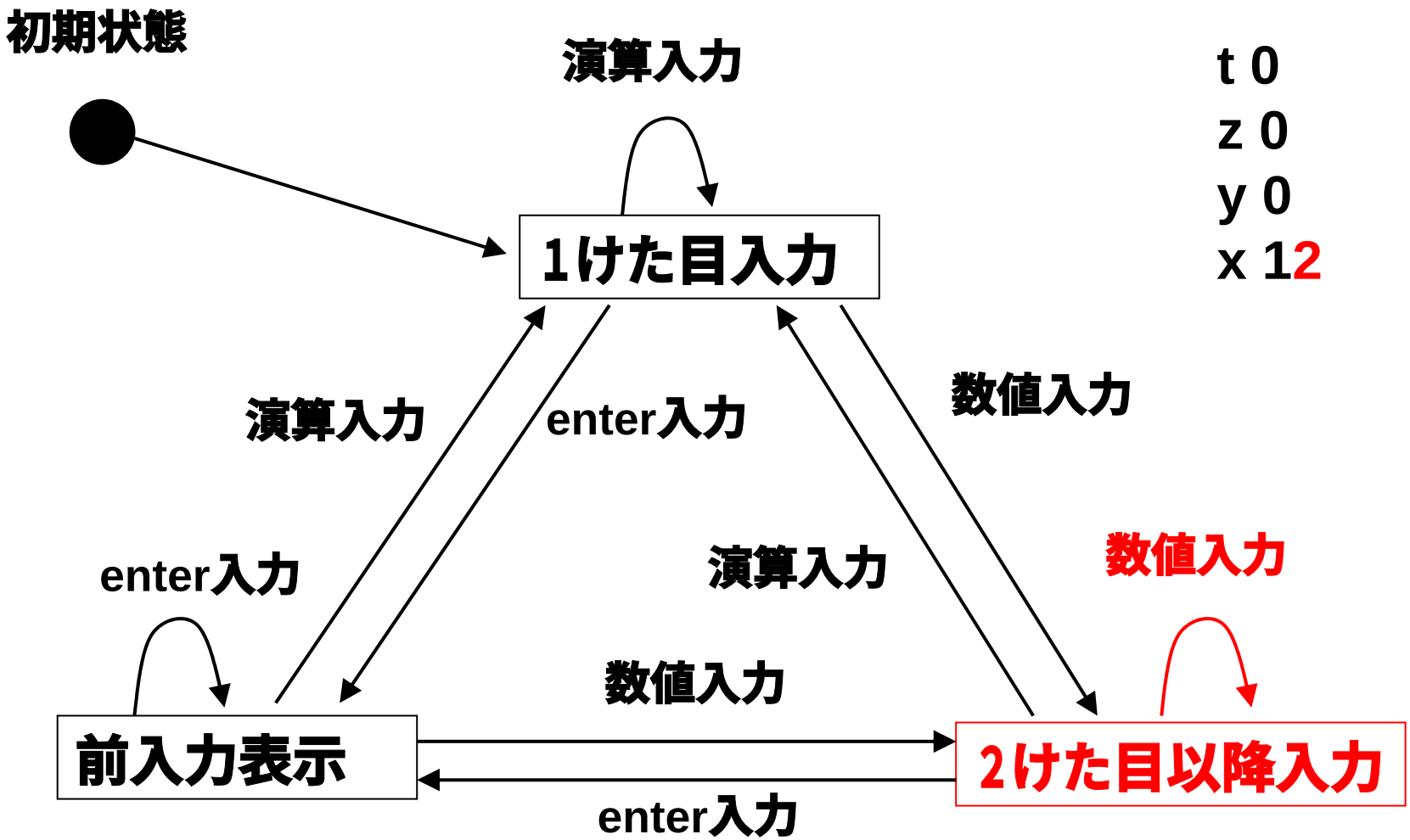
電源投入直後


初期状態
演算入力
1けた目入力
t 0
z 0
y 0
x 0
演算入力
enter入力
数値入力
enter入力
演算入力
数値入力
数値入力
前入力表示
2けた目以降入力
enter入力

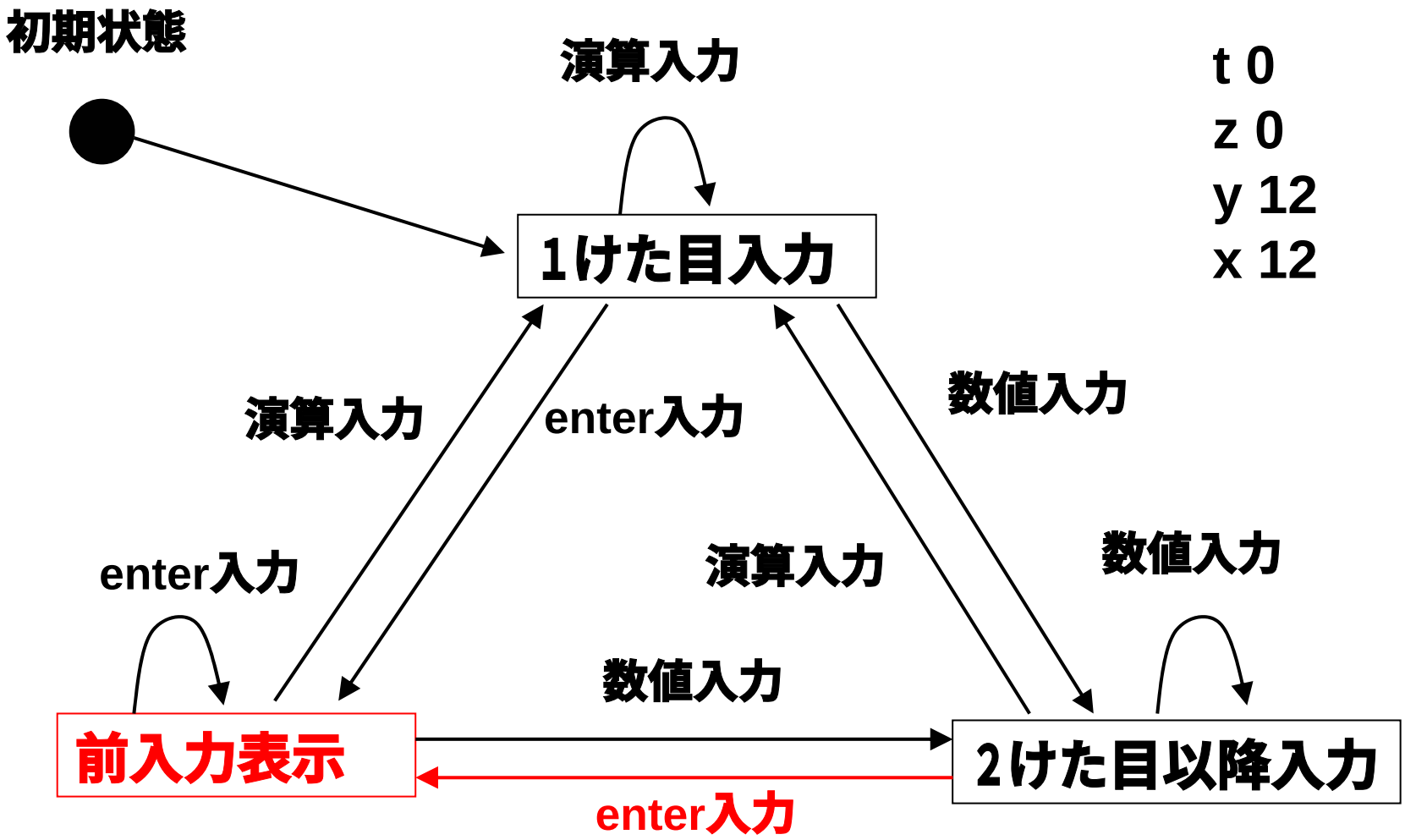
12enter33+21*



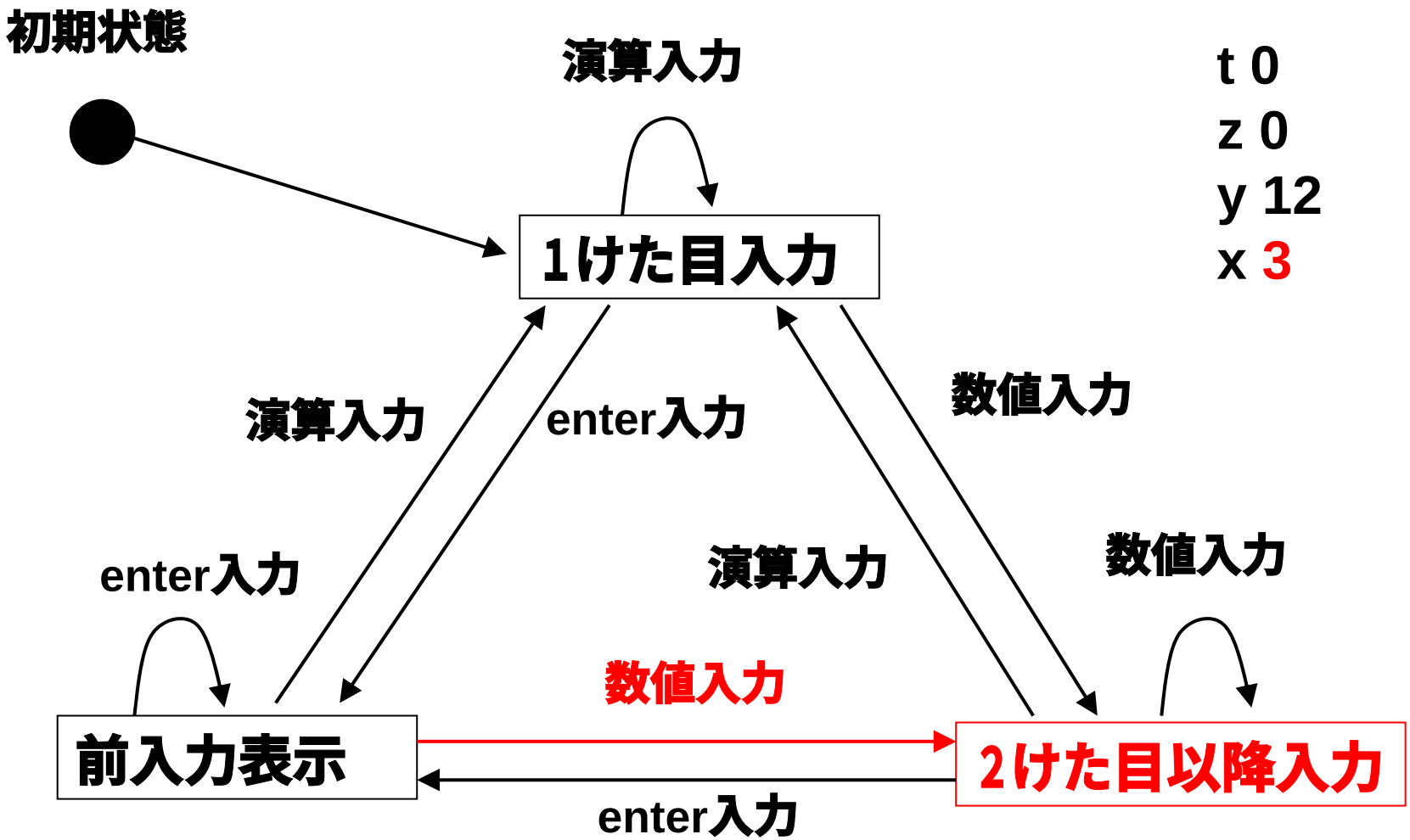
12enter33+21*



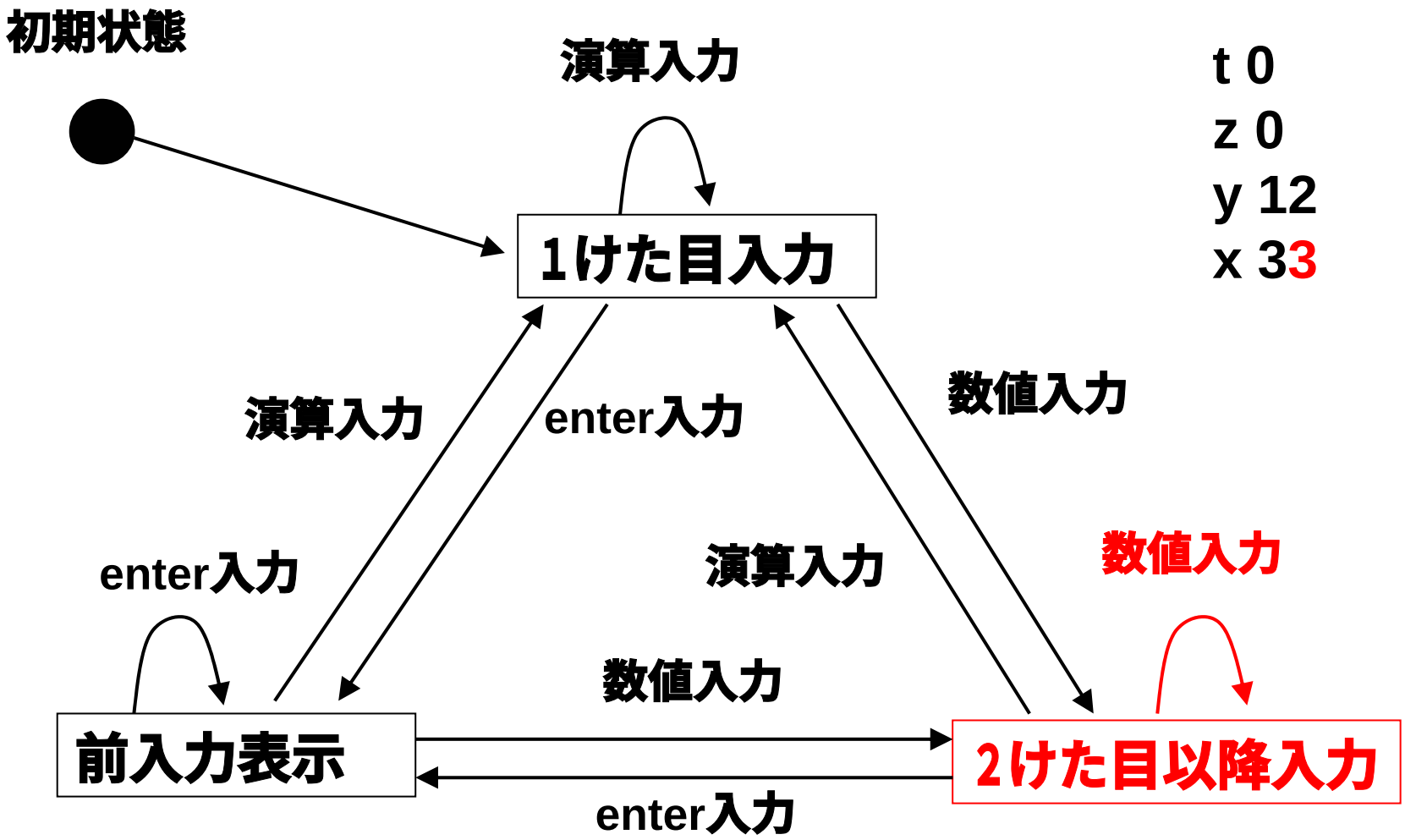
12enter33+21*



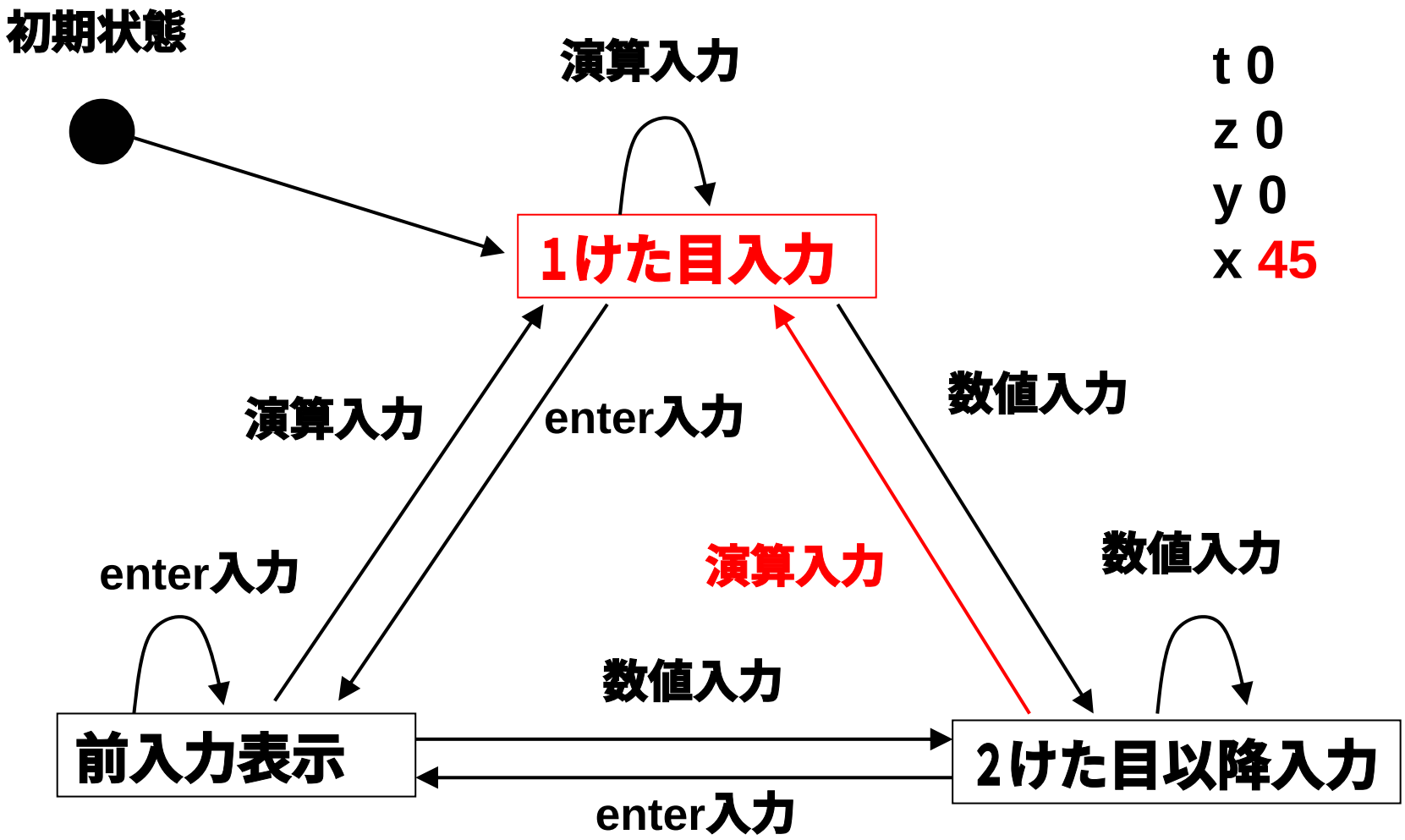
12enter33+21*



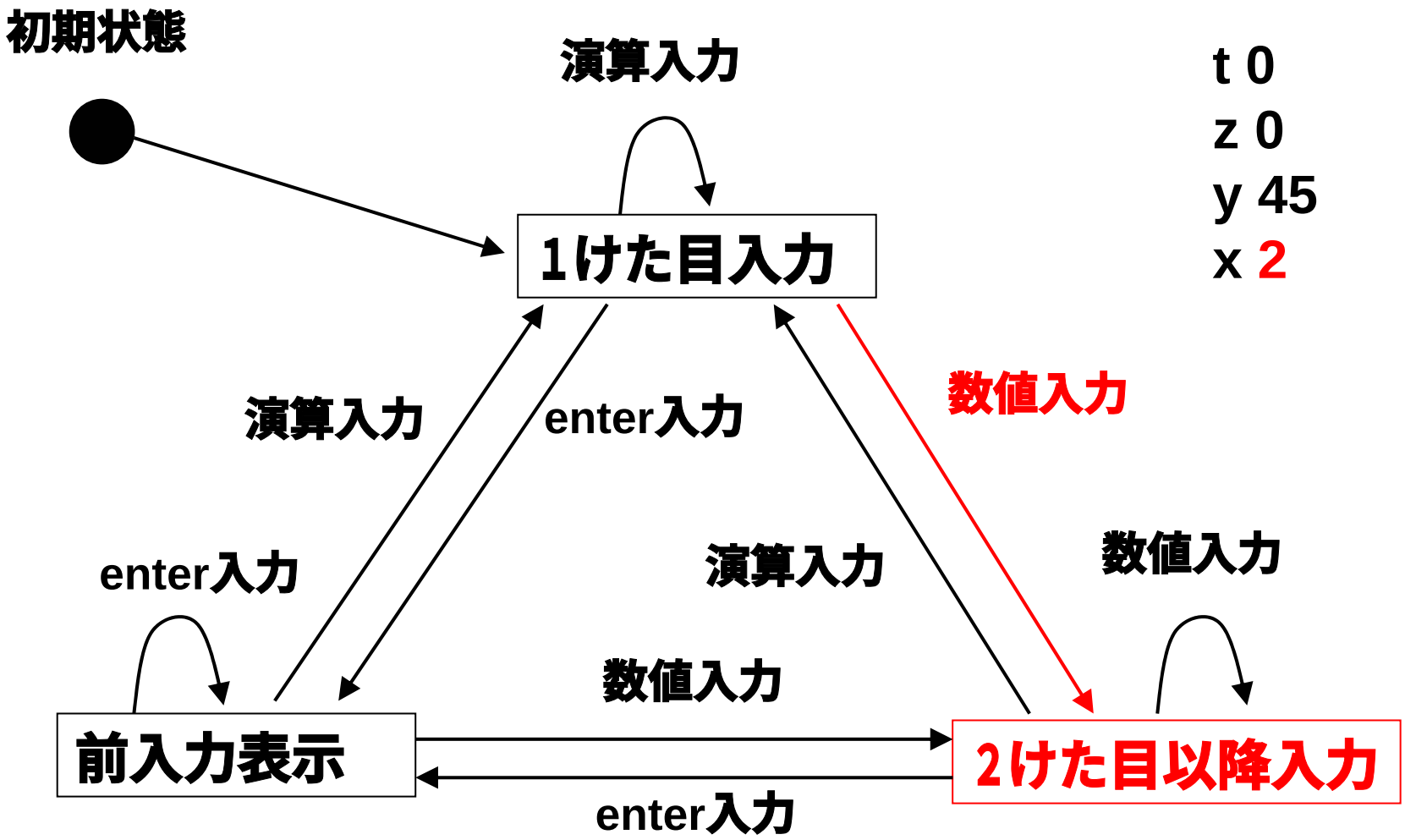
12enter33+21*



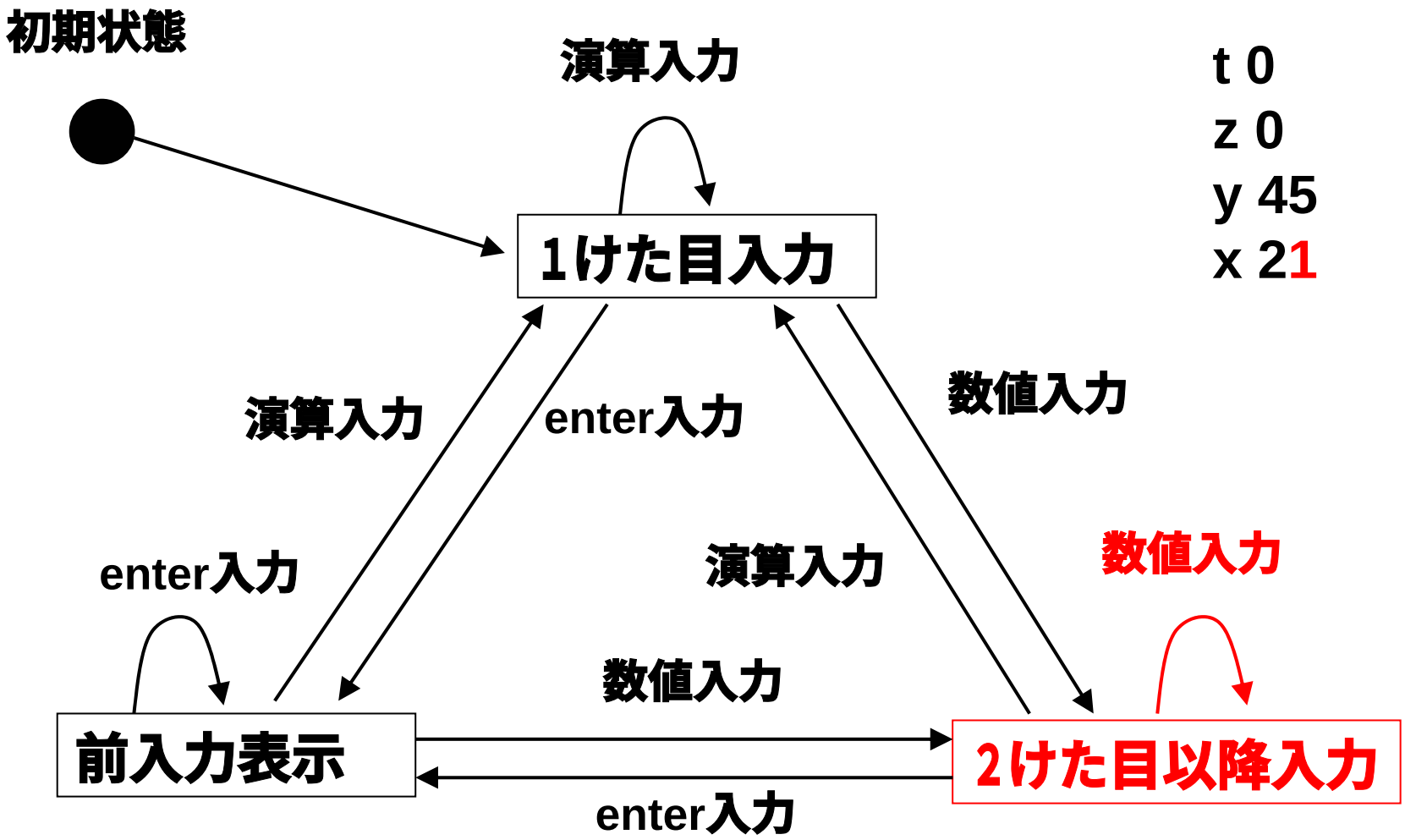
12enter33+21*



12enter33+21*



12enter33+21*



12enter33+21*