

ソフトウェア概論 A/B

-- データ構造 (配列/構造体) --

数学科 栗野 俊一

2011/11/11 ソフトウェア概

伝言

私語は慎むように !!

□ 色々なお知らせについて

- 栗野の Web Page に注意する事

<http://edu-gw2.math.cst.nihon-u.ac.jp/~kurino>

□ 廊下側の一列は遅刻者専用です(早く来た人は座らない)

□ 講義開始前にすませておくこと

- PC の電源を入れる
- ネットワークに接続しておくこと
- 今日の資料に目を通しておくこと

□ 講義前の注意

- 講義前は、栗野は準備で忙しいので TA を捕まえてください

□ やる気のある方へ

- 今日の資料は、すでに上っています
 - ▷ どんどん、先に進んでかまいません

前回の復習

□ 前回の復習

○ 繰返し構文(loop) : while/for/do ~ while

- ▶ 繰返し構文 : while (「条件」) {「本体」}
- ▶ 「条件」が成立する限り、「本体」部分を繰り返す
- ▶ 「本体」部分で「代入」により「条件」が変化するのが普通
- ▶ 「目的(=条件)成立」まで、繰り返すという考え方

○ 繰返し構文と一緒に利用する命令

- ▶ break : loop を中断(break)し、loop を抜けてしまう
- ▶ continue : loop を継続(continue)するために、loop の先頭に戻る

○ 色々な繰返し

- ▶ n 回数繰り返す : 本当に同じ事の繰返しをしたい場合
- ▶ i を 0 ~ n-1 まで : パラメータ i を変化させながら、それぞれの処理をする
- ▶ 条件が成立するまで繰り返す : 繰返し回数は不定で、「条件」が成立するまで

○ 状態機械モデル

- ▶ 状態機械とは ? : 「状態」を持つ機械 / 「状態」によって振舞が異なる
- ▶ 入力によって、状態を変更する / 出力は入力だけでなく状態にも依存する

○ 文字列のパターンと正規表現

- ▶ 正規表現:文字列パターン(文字列の集合)の表現方法の一つ
- ▶ 正規表現で記述された文字列パターンは状態機械で判定できる

お知らせ

- 本日の予定
 - データ構造と型定義 (配列/構造体)
- 本日の目標
 - 演習
 - ▷ 課題の提出

前回 (2011/11/04) の課題

□ 前回 (2011/11/04) の課題

○ 課題 1:

▷ ファイル名 : 20111104-1-QQQQ.c (QQQQ は学生番号)

▷ 内容 :

◇ 1 から n 個の奇数の和を求める関数を作れ

◇ ファイル形式 : テキストファイル(C 言語プログラムファイル)

○ 課題 2:

▷ ファイル名 : 20111104-2-QQQQ.c (QQQQ は学生番号)

▷ 内容 :

◇ 入力された文字列が、C 言語の 8 進整数を表すかどうかを判定する関数を作れ

◇ 引数は文字列で、結果は関数の返り値とする

◇ 返り値の意味 : 1 -- 8 進整数 / 0 -- それ以外

◇ 8 進整数の正規表現 : `-(0[0-7]*)`

▷ ファイル形式 : テキストファイル(C 言語プログラムファイル)

今週 (2011/11/11) の課題

□ 今週 (2011/11/11) の課題

○ 課題 1:

- ▶ ファイル名 : 20111111-1-XXXX.c (XXXX は学生番号)
- ▶ 内容 :
 - ◇ 二次元のベクトル型(Vector2D) を定義し、以下の機能を実現する
 - ◇ 座標は整数でよい
- ▶ 機能 :
 - ◇ 入力,出力,和,定数倍
- ▶ ファイル形式 : テキストファイル(C 言語プログラムファイル)

情報の表現

□「情報」の「表現」方法

- コンピュータは、「数値」の「計算」しかできない
- 現実には、「様々な情報」の「処理」がしたい
 - ▶ この二つのギャップを埋めるのは何か？

□ 情報を巡る、三つの「形態」

- 情報：「現実」世界での「何か(例:音)」
 - ▶ Input(入力)：「情報」を「データ」に変換する
 - ▶ Output(出力)：「データ」を「情報」に変換する
 - ▶ I/O：ハードウェア (音:マイク/スピーカ) によって実現
- データ：「コンピュータ」内での「情報」の表現(例:正弦波)
 - ▶ Encode(符号化)：「データ」を「数値」に変換する
 - ▶ Decode(解釈)：「情報」を「データ」に変換する
 - ▶ コーディング規則：ソフトウェア(プログラム)によって実現
- 数値：「コンピュータ」が直接扱える形 (数値)
 - ▶ 「計算」する事ができる
 - ▶ 計算：コンピュータ自身の基本的な機能によって実現

情報の処理

□ 情報処理の目的

- 「現実」世界での「何か(例:音)」を「操作」したい

- ▷ 機能 (Function) : 情報を操作する能力

□ 情報処理の流れ

- 情報(Information)は、入力(Input)され、データ(Data)になる
- データ(Data)は、符号化(Encode)され、数値(Number)になる
- 数値(Number)は、計算(Cluculus)され、別の数値になる
- 別の数値は、解釈(Decode)され、別のデータになる
- 別のデータは、出力(Output)され、別の情報になる

□ 同型構造

- 情報とデータは I/O により同型構造を持つ

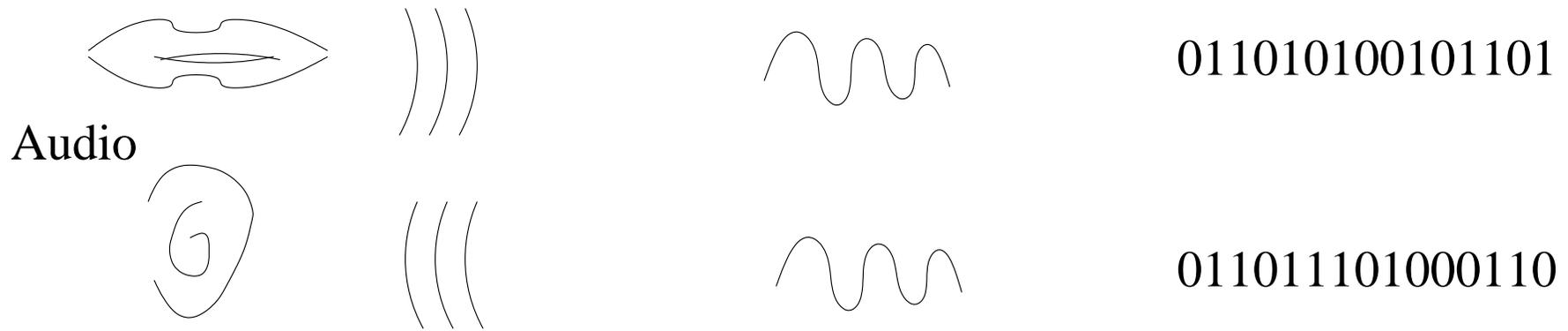
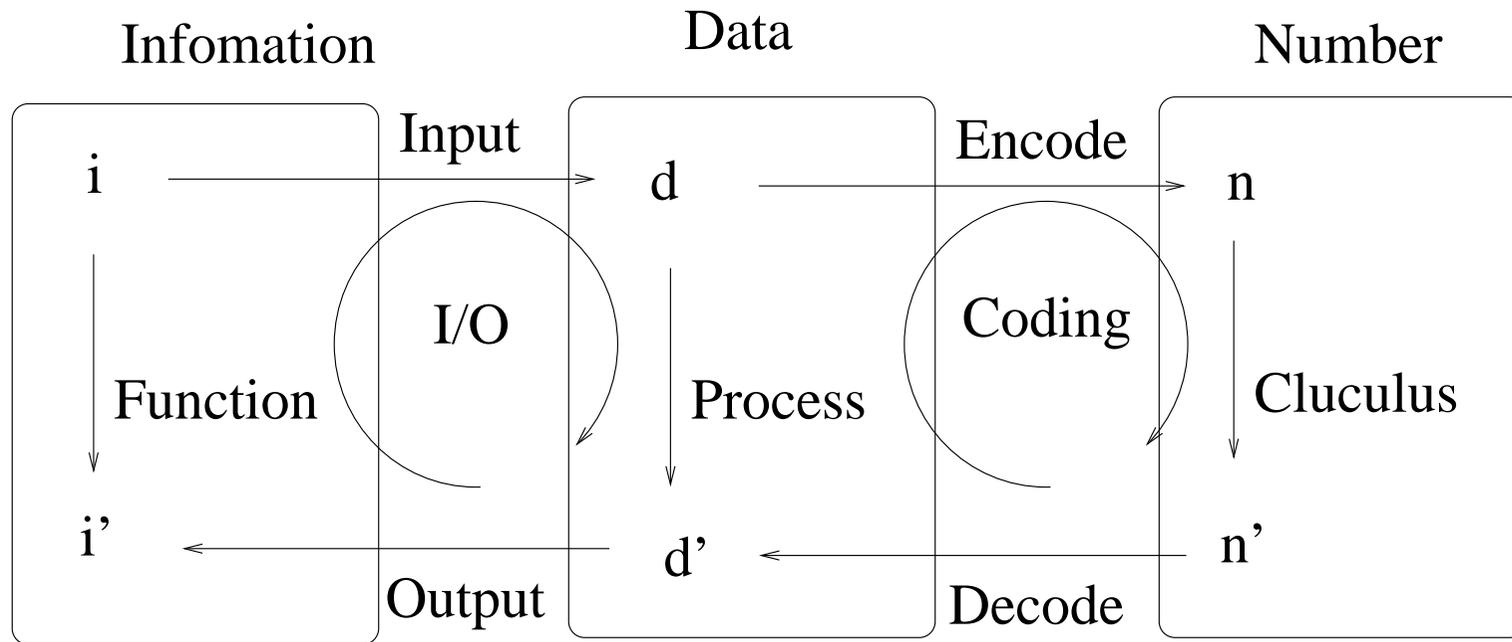
- ▷ 対応は「全射」でなければならない

- データと数値は Coding により同型構造を持つ

- ▷ 同型構造によって、機能(Function)は、処理(Process)を経由して計算(Cluculus)される

- ▷ これらは「可換」になっている

情報の処理の構造図



コーディングの仕組み

□ コーディングの仕組み

○ 基本は「直積」(既存の情報の組み合わせ)で行う

- ▶ 例 1: 平面上の「点」の表現 → 「座標」=「数値の二つ組(x,y)」で表現
- ▶ 例 2: 有理数の表現 → 「分母・分子」=「数値の二つ組 p/q」で表現
- ▶ 直積は、空間を拡大する

○ 「制約」で、同型にする

- ▶ 無意味な組み合わせ(中への対応)や、重複(一対一でなう)はプログラムで処理
- ▶ 例 1: 有理数で 3/0 に対応するものはない (エラー / 例外処理)
- ▶ 例 2: 有理数で 3/6 と 1/2 は同じ物に対応 (同値類 / 正規化)

□ C 言語でのコーディング

○ データの表現

- ▶ 直積: 構造体
- ▶ 制約: 正規化のプログラムを作成する

○ 機能の実現

- ▶ 「計算(数値の操作)」によって、「機能」を実現するプログラムを作る

構造体

□ 型

○ 「空間」の名前

- ▶ 集合(どんな要素が含まれているか..) と機能(どんな演算ができるか..) からなる

○ 基本型

- ▶ 予め C 言語で、定義されており利用できる型 (int, double, etc..)

○ 導出型

- ▶ プログラマが作成する型 (集合と、機能は自分で実装する..)

○ typedef

- ▶ typedef によって、新しい型に名前をつける事ができる
- ▶ 変数の宣言と、代入が可能になる
 - ◇ 構文 : typedef 型 新しい型名;
 - ◇ 例 1 : typedef int myInt; myInt を int で定義
 - ◇ 例 2 : typedef struct { int x; int y } Point;

□ 構造体

○ 複数の既存の型から、その直積となる新しい型を作る

- ▶ 例1 : int x と int y の組み合わせ
- ▶ struct { int x; int y; }; ≡ { <x,y> | x:int, y:int }
- ▶ struct {int x;int y} v; // Point v;
- ▶ v ≡ <p,q>, v.x ≡ p, v.y ≡ q
- ▶ 例 2 : 三次元 (int x, int y, int z) の場合