Introduction to Programming

Bertrand Meyer

Lecture 8: Control structures (1)

Last revised 20 November 2004

The notion of algorithm

General definition:

An algorithm is the specification of a process to be carried out by a computer

Not quite an algorithm

PREPARAZIONE E TEMPI DI COTTURA
ZUBEREITUNG - PREPARATION

Versen le verdure ancora saporite in 1 litro abbondante d'acqua fredda con 2 cucchiai d'olio, sale e coccette secondo i tempi indicati. Tiefgefrorren Gemüse in einen Liter kalten Was ser geben, 2 Esslöffel Öl und Salz hinzufügen.

Versen les légumes surgelés dans 1 litre d'eau froide, ajouter deux cuillers à soupe d'huile et du sel.

45 min.

15 min.
5 properties of an algorithm

1. Defines data to which process will be applied
2. Every elementary step taken from a set of well-specified actions
3. Describes ordering(s) of execution of these steps
4. 2 and 3 based on precisely defined conventions, suitable for execution by an automatic computer
5. For any data, guaranteed to terminate after finite number of steps

Algorithm vs program

- "Algorithm" usually considered a more abstract notion, independent of platform, programming language etc.
- In practice, the distinction tends to fade:
  - Algorithms need a precise notation
  - Programming languages becoming more abstract
- However:
  - In programs, data (objects) are just as important as algorithms
  - A program typically contains many algorithms and object structures

What makes up an algorithm

- Basic steps:
  - Feature call \(x.f(a)\)
  - Assignment (yet to be studied)
  - ...

- Sequencing of these basic steps:

CONTROL STRUCTURES
Control structures

Definition: program construct that describes the scheduling of basic actions

Three fundamental control structures:

- Sequence
- Loop
- Conditional

"Control structures of Structured Programming"

Control structures as problem-solving techniques

- Sequence: "To achieve C from A, first achieve an intermediate goal B from A, then achieve C from B"

- Loop: solve the problem on successive approximations of its input set

- Conditional: solve the problem separately on two or more subsets of its input set

The sequence (or Compound)

\[ \text{instruction}_1 \]
\[ \text{instruction}_2 \]
\[ \ldots \]
\[ \text{instruction}_n \]
Semicolon as optional separator

\[ \text{instruction}_1; \]
\[ \text{instruction}_2; \]
\[ \ldots; \]
\[ \text{instruction}_n. \]

Correctness of a Compound

Precondition of \( \text{instruction}_1 \) must hold initially

Postcondition of each \( \text{instruction}_i \) must imply precondition of each \( \text{instruction}_{i-1} \)

Final effect is postcondition of \( \text{instruction}_n \)

Conditional instruction

```
if
  Condition -- Boolean_expression
then
  Instructions -- Compound
else
  Other_instructions -- Compound
end
```
### The conditional as problem-solving technique

![Diagram of problem space with regions and techniques]

#### Basic form

```plaintext
if Condition then
    Instructions
else
    Other_instructions
end
```

#### A variant of the conditional

```plaintext
if Condition then
    Instructions
end
```

(as if "else ")
Nesting

if Condition\textsubscript{1} then
  Instructions\textsubscript{1}
else
  if Condition\textsubscript{2} then
    Instructions\textsubscript{2}
  else
    Instructions\textsubscript{3}
end
Nesting

```plaintext
if Condition1 then
    Instructions1
else
    if Condition2 then
        Instructions2
    else
        if Condition3 then
            Instructions3
        else
            ...  
        end
    end
end
```

Nested structure

Comb-like structure
Comb-like conditional

```plaintext
if Condition then
  Instructions
elsif Condition then
  Instructions
elsif Condition then
  Instructions
elsif ...
else
  Instructions
end
```

Comb-like structure

![Comb-like structure diagram]

End of lecture 8