Control structures

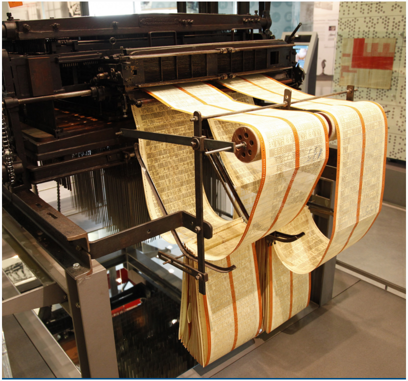
Introduction and course outline

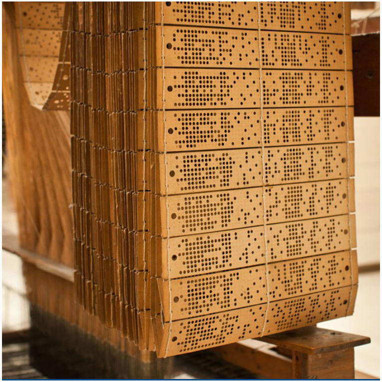
Xavier Leroy

2024-01-25

Collège de France, chair of software sciences
xavier.leroy@college-de-france.fr

How to describe and control the sequencing of operations?





For example: using punched cards that form a loop!

Control in processor architectures

The dominant model since 1945 (von Neumann, Turing):

- most instructions are executed sequentially, in the order in which they occur in memory;
- branch instructions jump to a specific code address.

```
 0:  set r1, 1
 4:  set r2, 1
 8:  output r1
12:  add r2, r2, 1
16:  mul r1, r1, r2
20:  branch 8
```

Control in programming languages

The first languages follow the processor model:
unstructured control with jumps — the infamous `goto`.

Later, languages evolve towards structured control
expressed using linguistic mechanisms called control structures:

- conditionals, counted loops, general loops, …
- procedures, functions, methods, coroutines, …
- exception handlers, continuation handlers, effect handlers.

Other languages try to make control implicit:
the declarative languages
(dataflow, functional, logic, constraint-based, …).

In this lecture: control paradigms

A journey through the space and time of programming languages, from the viewpoint of control:

- what language constructs are available to programmers to express and constrain the sequencing of computations?
- what structure and organization of source programs are enabled or encouraged by these constructs?

Our approach will be descriptive, often comparative, but also formal.

Course outline

25/01 The birth of control structures: from “goto” to structured programming.

- Assembly, FORTRAN, Algol, and descendants.
- The movement for structured programming.
- Expressiveness of structured control.

01/02 Non-local control: from subroutines to functions and coroutines.

- Subroutines, procedures, functions: for structure and for reuse.
- Non-local `goto` vs. exceptions.
- Iterators, generator, coroutines, cooperative threads.

08/02 Declarative programming: getting rid of control?

- Expression languages
- Dataflow languages.
- Functional languages.
- Logic languages.

Course outline

15/02 Continuations and control operators: building blocks for control structures

- Continuations as a semantic tool.
- Programming in continuation-passing style.
- Control operators: using continuations in direct style.

22/02 The practice of effects: from exceptions to effect handlers.

- Exceptions, restartable exceptions, continuation capture.
- Effect handlers in OCaml 5.
- Examples of uses: coroutines, cooperative threads.

29/02 The theory of effects: from monads to algebraic effects

- Monads: a theory of effect propagation.
- Free monads and interaction trees.
- Algebraic effects: a theory of effect generation.
- Effect handlers: in the semantics, in the language.

Course outline

07/03 Static typing of effects

- Type safety for advanced control structures.
- Type and effect systems.

14/03 Program logics for control and effects

- Hoare logics for classic control structures.
- Separation logics for effect handlers.