L211 Logic and Mathematics

14. Lecture

Norbert PREINING
preining@jaist.ac.jp

http://www.preining.info/jaist/l211/2015e/

Previous lecture

Graphs, modules of computation

- spanning trees
- Primitive recursive functions
- Ackermann function
- μ-recursive functions
- Alan Turing
- Enigma
- Turing machine

while programming language

extremely simple programming language, all possible instructions are

- input: *read X*
- compute:
  - $X := 0$, $X := X + 1$, $X := X - 1$
  - $Y := X$
- while: *while* $X \neq Y$ *do* S
- S; S

Example program

```
read X
Z := 0
while X /= 0 do
    Y := X
    while Y /= 0 do
        Z := Z + 1
        Y := Y - 1
    X := X - 1
Z
```

With input 4, what is the output?

Computational model

The following statements are equivalent:

- f is μ-recursive
- f is Turing-computable (can be computed with a Turing machine)
- f is while-computable (can be computed with a while program)

The concept of *computable* is *stable*.

Sp what are *computable* functions?

Church-Turing These

Software Verification

Famous bugs

- Therac-25 x-ray machine: over-exposition error (5 dead, 1985–1987)
- Ariane 5 rocket, Flight 501 (overflow, 1996, $370 million!)
- Mars Climate Orbiter: (lbf vs. N, 1999)
- Intel Pentium F00F bug: planning error
- Toyota's Electronic Throttle Control System (ETCS): sudden break accidents (2009–2011)
- Heartbleed bug (OpenSSL, 2012-2014)

Hoare calculus

$$\{P\}\ S\ \{Q\}$$

P, Q expressions, S command

Axiom system

- Axiom: $\{P\}\ \mathrm{skip}\ \{P\}$
- Axiom: $\{P[E/X]\}\ X := E\ \{P\}$
- Rule:

$$\frac{\{P\}\ S\ \{Q\} \quad \{Q\}\ T\ \{R\}}{\{P\}\ S; T\ \{R\}} \qquad \frac{\{P \wedge B\}\ S\ \{P\}}{\{P\}\ \text{while } B \text{ do } S\ \{\neg B \wedge P\}}$$

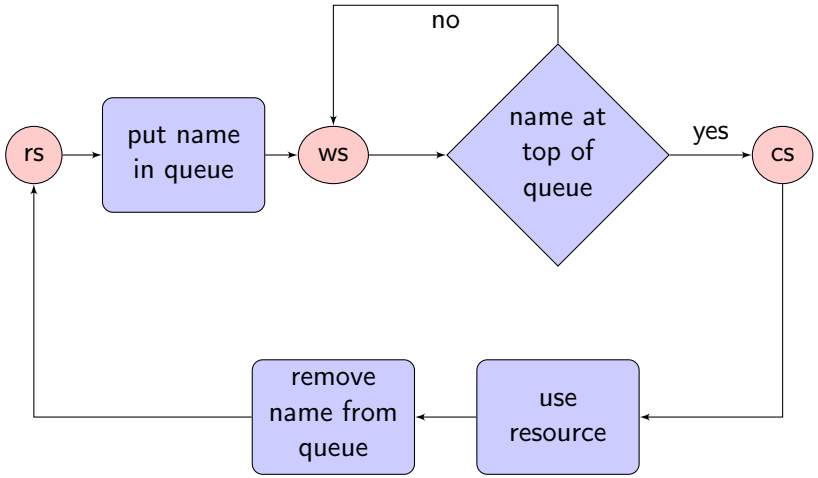
Formal specification and formal specification languages

- mathematical specification of a protocol
  while program
- input conditions
  $X > 0$
- necessary conditions (output conditions)
  $Z = \sum_{k=1}^{X} k$
- formal proof

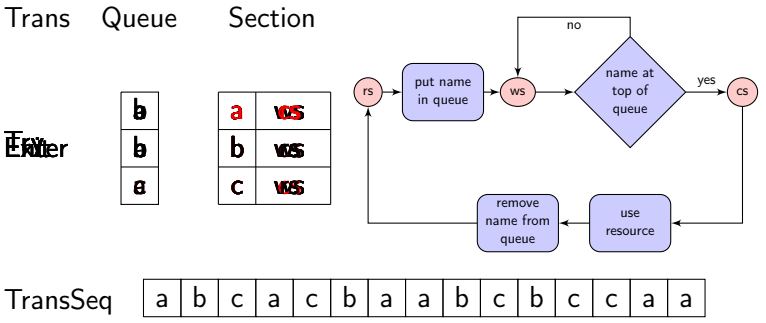
QLOCK

- Several processes are competing for the same common resource (e.g., keyboard)
- Exclusion control: we want to guarantee that only one process at a time has access to the resource
- Method:
  - A process that wants to use the resource, puts his name into a queue;
  - if the name of a process is at the top of the queue, it can use the resource;
  - after finishing, the process removes his name from the top of the queue.

QLOCK flow



Qlock Example



Various conditions

Necessary condition: mutual exclusion

Only one process at a time can use the resource.

Fairness

A process that wants to use the resource will eventually gain access to it.

Closing

Summary of the lecture

- Number systems
  - $\mathbb{N} \to \mathbb{Z} \to \mathbb{Q} \to \mathbb{R} \to \mathbb{C}$
- Proofs
  - direct, indirect, by contradiction, by induction
- the Infinite
  - cardinality, countably infinite

Summary cont.

- Axiomatic method
  - constructions, Euclid, Peano
- Functions
  - Limit, sequence, series, continuity
- Computational model
  - recursive functions, Turing machine, while

Abstraction

Generalization

Formalization

Neil deGrasse Tyson – Cosmos

To make this journey, we need imagination, but immagination alone is not enough! Because the reality of nature is far more wondrous than anything we *can* imagine.
This adventure is made possible by generations of researchers strictly adhering to a simple set of rules:

- test ideas by experiment and observation,
- build on those ideas that pass the test,
- reject the ones that fail,
- follow the evidence wherever it leads,
- and question everything!

Accept these terms and the Cosmos is yours!

Literature

- Richard Courant and Herbert Robbins, What is Mathematics?, 2nd ed., Oxford University Press, 1996.
- Keith Devlin, Introduction to Mathematical Thinking, 2012.
- Harold Coxeter, Introduction to Geometry, John Wiley & Sons, 1969.
- Simon Singh, Fermat's last THeorem, Anchor Books, 1998.

Sources

Wikimedia