technische universiteit eindhoven

Models of Computation: Automata and Processes

ir. P.J.A. van Tilburg
Formal Methods Group, Department of Mathematics and Computer Science
NWO project 612.000.630

Motivation

- Automata theory provides simple models of computation for understanding the principles of computing and analysis of *computability*.
- Process theory has its origins in automata theory but focuses more on studying the notion of *interaction* and parallel behaviour.
- Goal: the *integration* of automata and process theory.
- The attempt at integration will reveal differences and similarities. We can use *analogies* between the theories to make the integration explicit.
- Add process theory to the undergraduate curriculum.

Automata

Automata accept a language (a set of sequences of symbols) as correct or wanted behaviour. An automaton can for example model a coffee-vending machine:

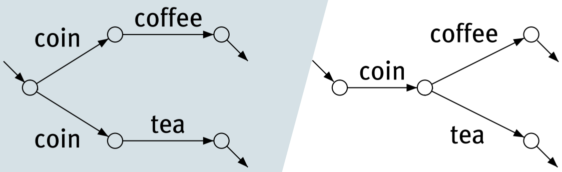

Figure 1: Two language equivalent automata

The above automata accept the same language, they are *language equivalent*:

- a **coin** followed by **coffee**
- a **coin** followed by **tea**

Process theory differentiates between them using the *bisimulation equivalence*:
For a person using the machine it would make a difference whether inserting a coin predetermines the result or the choice is still available after inserting the coin.

Regular Expressions and Process Terms

- Regular expressions describe languages:

$$\text{coin} \cdot \text{coffee} + \text{coin} \cdot \text{tea}, \qquad \text{coin} \cdot (\text{coffee} + \text{tea})$$

- While regular expressions can describe all regular languages, their process term counterparts cannot describe all regular processes (shown in [1]).
- Process terms have calculation rules (axioms). E.g.:

$$\begin{aligned} (A3) \quad & x + x &= x \\ (A4) \quad & (x+y)z &= xz + yz \end{aligned}$$

- The axiom $x(y+z) = xy + xz$ holds in automata theory but it does not hold in process theory!
- In process theory there are additional operators, such as $\|$, $|$, and $\lfloor\!\lfloor$, for describing parallel behaviour which are not present in automata theory.

Grammars and Recursive Specifications

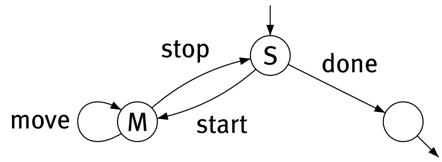

Figure 2: The context-free process S

- Grammars can also describe languages. The right-linear grammars from automata theory are equivalent to the recursive specifications of process theory.
- We can give both for the automaton in Figure 2:

$$\begin{array}{ll|ll} S & \rightarrow \mathit{start}\ M\ S \mid \mathit{done} & S & = \mathit{start} \cdot M \cdot S + \mathit{done} \\ M & \rightarrow \mathit{move}\ M \mid \mathit{stop} & M & = \mathit{move} \cdot M + \mathit{stop} \end{array}$$

- In automata theory a context-free language can be accepted by an automaton using a stack (the push-down automaton). In process theory, a context-free process can be transformed into a process communication with a Stack process, making the interaction more visible.

Research Questions

A selection of some of the research questions of the project:

- The additional operators present in process theory create new classes of languages, such a basic parallel class or a communicating class. What can be expressed by each of these new classes? Do they have some finite axiomatisation?
- In automata theory the Chomsky hierarchy discerns several classes of languages (regular, context-free, etc.). The new classes create an extended, more fine-grained version of this hierarchy. What does this hierarchy look like?
- Similar to the way a context-free language can be transformed into a process communicating with a typical process such as the Stack, can such a typical process be found for the other classes as well?

Research Team

- prof.dr. J.C.M. Baeten,
- dr. C.A. Grabmayer,
- prof.dr. J. Karhumäki,
- dr. B. Luttik,
- prof.dr.ir. C.A. Middelburg,
- ir. P.J.A. van Tilburg.

References

[1] C.A. Grabmayer J.C.M. Baeten, F. Corradini. A characterization of regular expressions under bisimulation. *Journal of the ACM*, 54(2):6, 2007.

Netherlands Organisation for Scientific Research