Technische Universiteit Eindhoven University of Technology

Models of Computation: Automata and Processes

Paul van Tilburg

Nederlandse Organisatie voor Wetenschappelijk Onderzoek

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 and Process Theory: Similarity and Differences

Automata and Equivalences

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:

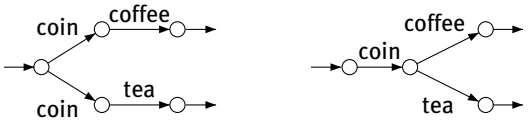

Figure 1: Two language equivalent automata

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 *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.:

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

- 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

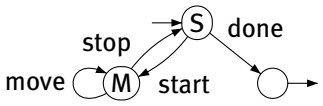

Figure 2: A regular process

- 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:

$$\begin{array}{l|l} S \to start\ M \mid done & S = start \cdot M + done \\ M \to move\ M \mid stop\ S & M = move \cdot M + stop \cdot S \end{array}$$

Obtained Research Results

- In automata theory a context-free language can be accepted by an automaton using a stack (a pushdown automaton). In process theory, a context-free process can be transformed into a process communicating with the *Stack process*, making the interaction explicit. [2]
- Similar for basic parallel processes: a basic parallel process can be transformed into a process communicating with the *Bag process*. [3]
- Relative expressiveness between several classes has been investigated.

Extending the Chomsky Hierarchy

The Chomsky hierarchy discerns classes of languages (regular, context-free, etc.). The additional operators present in process theory create *new classes* of languages, such as the basic parallel class. The new classes create an extended, more fine-grained version of this hierarchy.
What does this new hierarchy look like? What can be expressed by each of these new classes? Do they have some finite axiomatisation?

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] J.C.M. Baeten, F. Corradini, and C.A. Grabmayer. A characterization of regular expressions under bisimulation. *Journal of the ACM*, 54(2), 2007.
[2] J.C.M. Baeten, P.J.L. Cuijpers, and P.J.A. van Tilburg. A Context-Free Process as a Pushdown Automaton. *Proceedings of CONCUR'08*, *LNCS* 5201, pp. 98–113, 2008.
[3] J.C.M. Baeten, P.J.L. Cuijpers, and P.J.A. van Tilburg. A Basic Parallel Process as a Parallel Pushdown Automaton. *Proceedings of EXPRESS'08*, *ENTCS*, 2008.