The Queue Automaton Revisited

Jos Baeten & Bas Luttik

CWI & TU/e

Juggling formal analysis, security, privacy and trust
Luxemburg, 10 April 2026

CWI

Sjouke

CWI

Pushdown automaton <=> context-free grammar

Sjouke looked at BPA (= pushdown processes), proving a decidability result.
A pushdown automaton generates a transition system, pairs of state and stack content.

CWI

Pushdown automaton <=> context-free grammar

Sjouke looked at BPA (= pushdown processes), proving a decidability result.
A pushdown automaton generates a transition system, pairs of state and stack content.

A pushdown process given by process algebra with $0, 1, a._{\_}$, sequencing and sequential value passing.

CWI

Pushdown automaton <=> context-free grammar

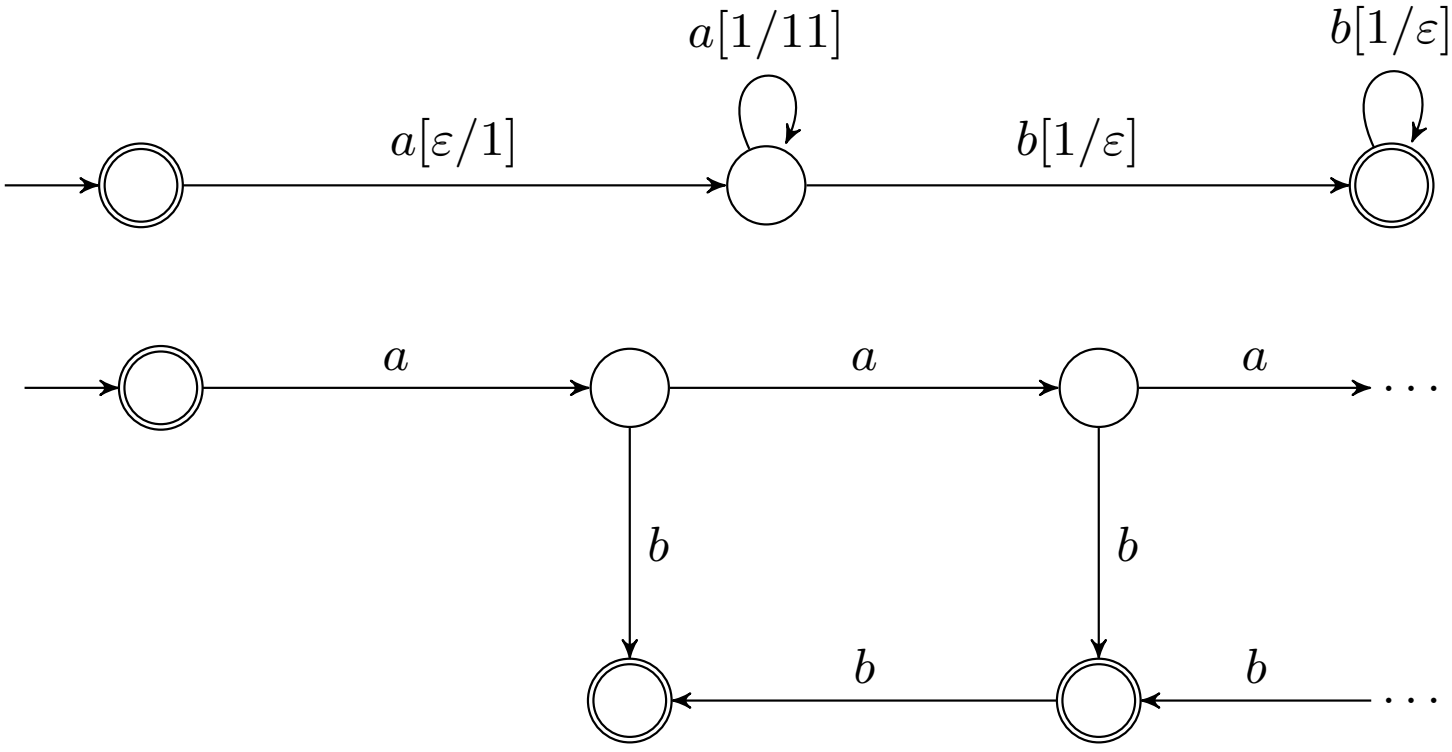
Sjouke looked at BPA (= pushdown processes), proving a decidability result.
A pushdown automaton generates a transition system, pairs of state and stack content.

A pushdown process given by process algebra with $0, 1, a._{\_}$, sequencing and sequential value passing.

A pushdown process is a regular process interacting with a stack.

CWI

Pushdown automaton <=> context free grammar



$$C = \mathbf{1} + C' \qquad C' = a.(b.\mathbf{1} + C' \,;\, (\mathbf{1} + b.\mathbf{1}))$$

CWI

Parallel pushdown automaton

A parallel pushdown automaton is an abstract model of a computer with bag-like memory.

CWI

Parallel pushdown automaton

A parallel pushdown automaton is an abstract model of a computer with bag-like memory.

A parallel pushdown process given by process algebra with $0, 1, a.\_$, parallel composition with synchronisation (flat priorities, renaming).

CWI

Parallel pushdown automaton

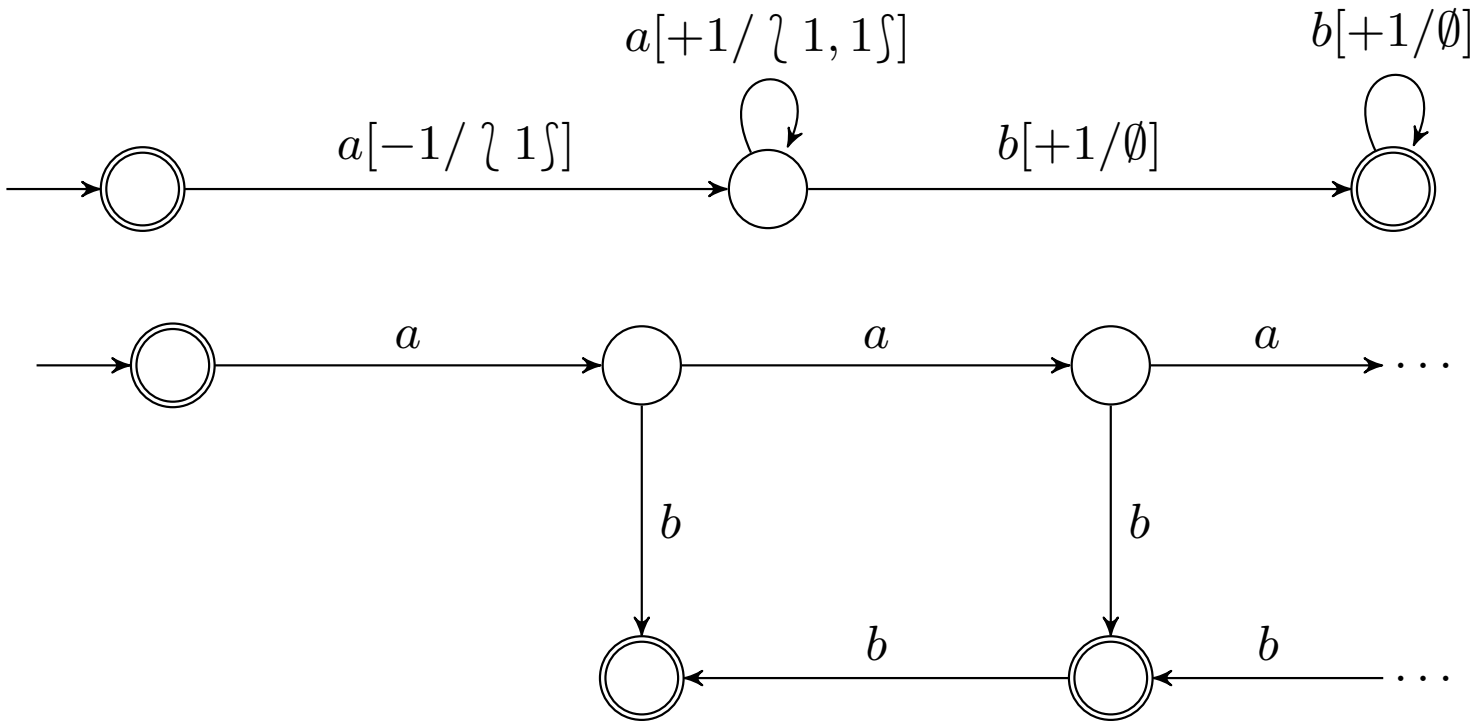
A parallel pushdown automaton is an abstract model of a computer with bag-like memory.

A parallel pushdown process given by process algebra with $0, 1, a.\_$, parallel composition with synchronisation (flat priorities, renaming).

A parallel pushdown process is a regular process interacting with a bag.

CWI

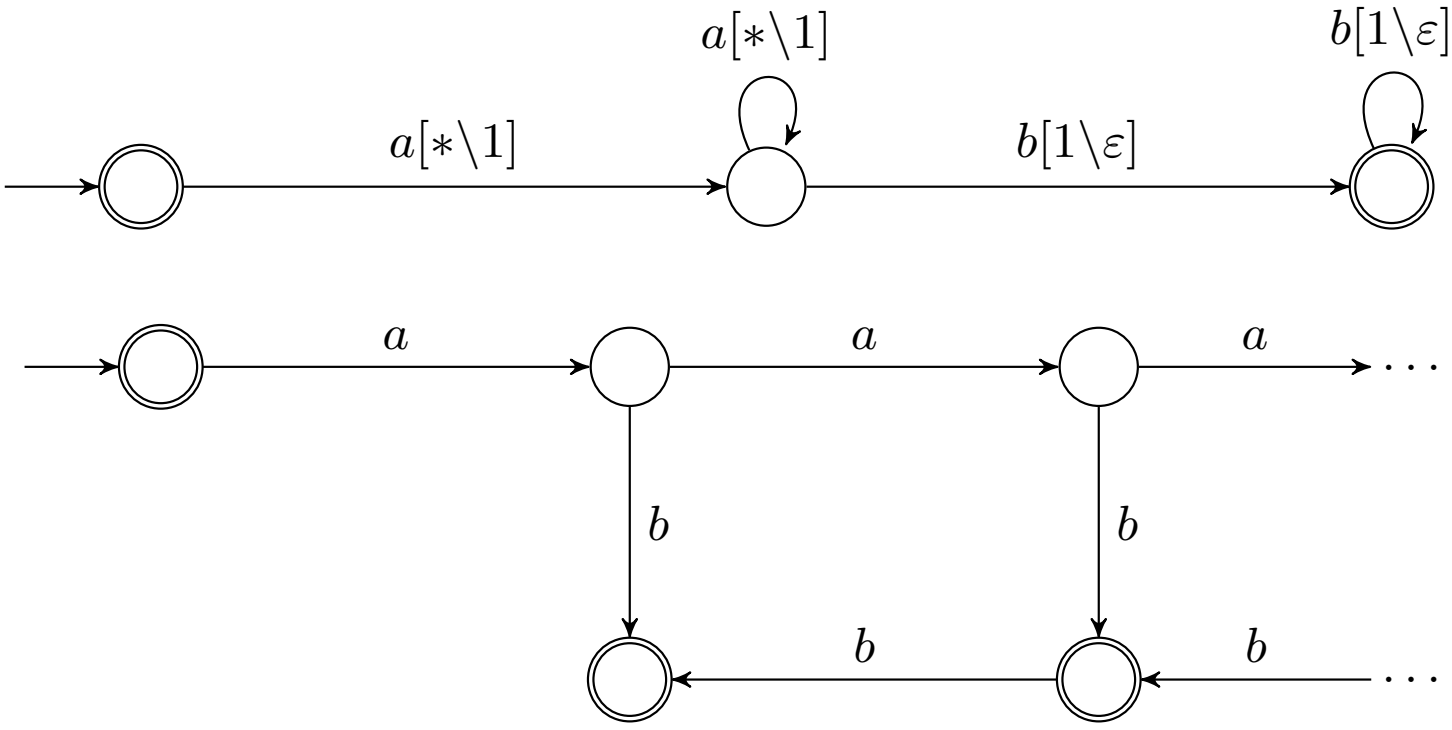
Parallel pushdown automaton



$$C = \mathbf{1} + \partial_b(C') \qquad C' = a.(b.B + C' \parallel (\mathbf{1} + b?.\mathbf{1})) \qquad B = \mathbf{1} + b!.B$$

CWI

Queue automaton?



$$C = \mathbf{1} + \partial_b(C') \qquad C' = a.(b.B + C' \parallel (\mathbf{1} + b?.\mathbf{1})) \qquad B = \mathbf{1} + b!.B$$

CWI

Queue automaton

Post (1943): deterministic queue automaton is equally expressive as a Turing machine

CWI

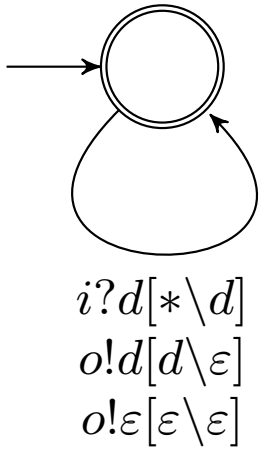
Queue automaton

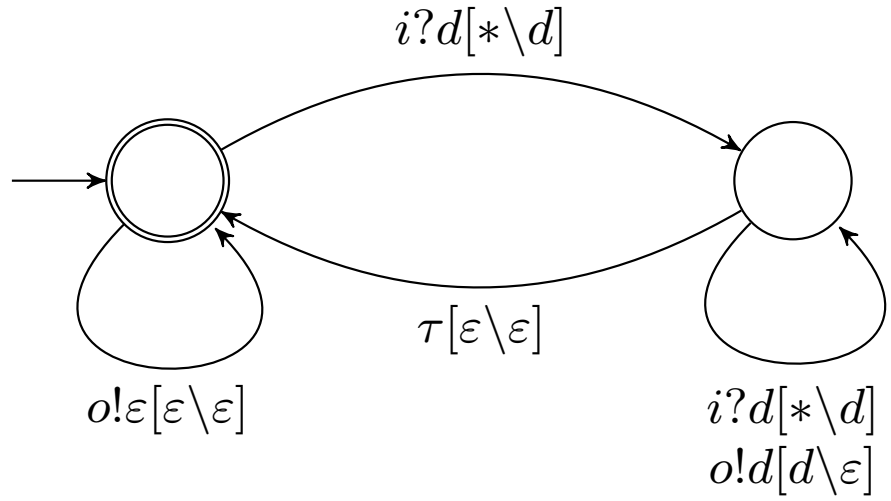
Post (1943): deterministic queue automaton is equally expressive as a Turing machine

This paper: non-deterministic queue automaton is equally expressive as a Reactive Turing Machine
With a queue automaton, we can define all computable functions and all executable processes
Definition of queue automaton is robust
Process algebra $\mathbf{0}, \mathbf{1}, a.\_, +$, parallel composition with communication, encapsulation, abstraction.
An executable process is a regular process interacting with a queue

CWI

Queue

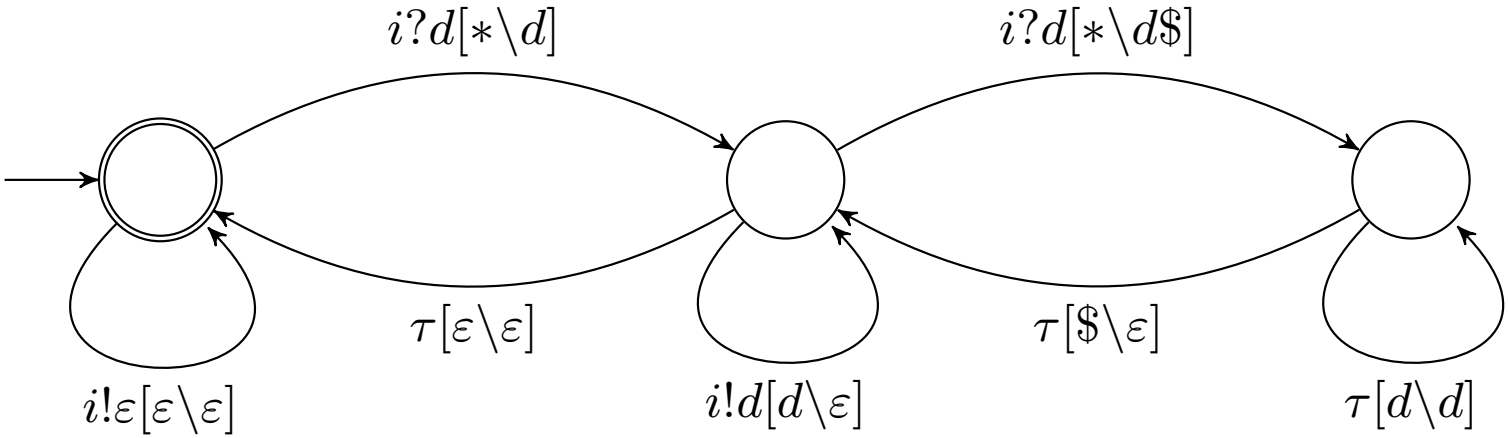




$$Q^{io} = \mathbf{1} + o!\varepsilon.Q^{io} + \sum_{d\in\mathcal{D}} i?d.\tau_{\mathcal{C}}(\partial_{\ell}(Q^{i\ell} \parallel o!d.Q^{\ell o}))$$

CWI

Stack



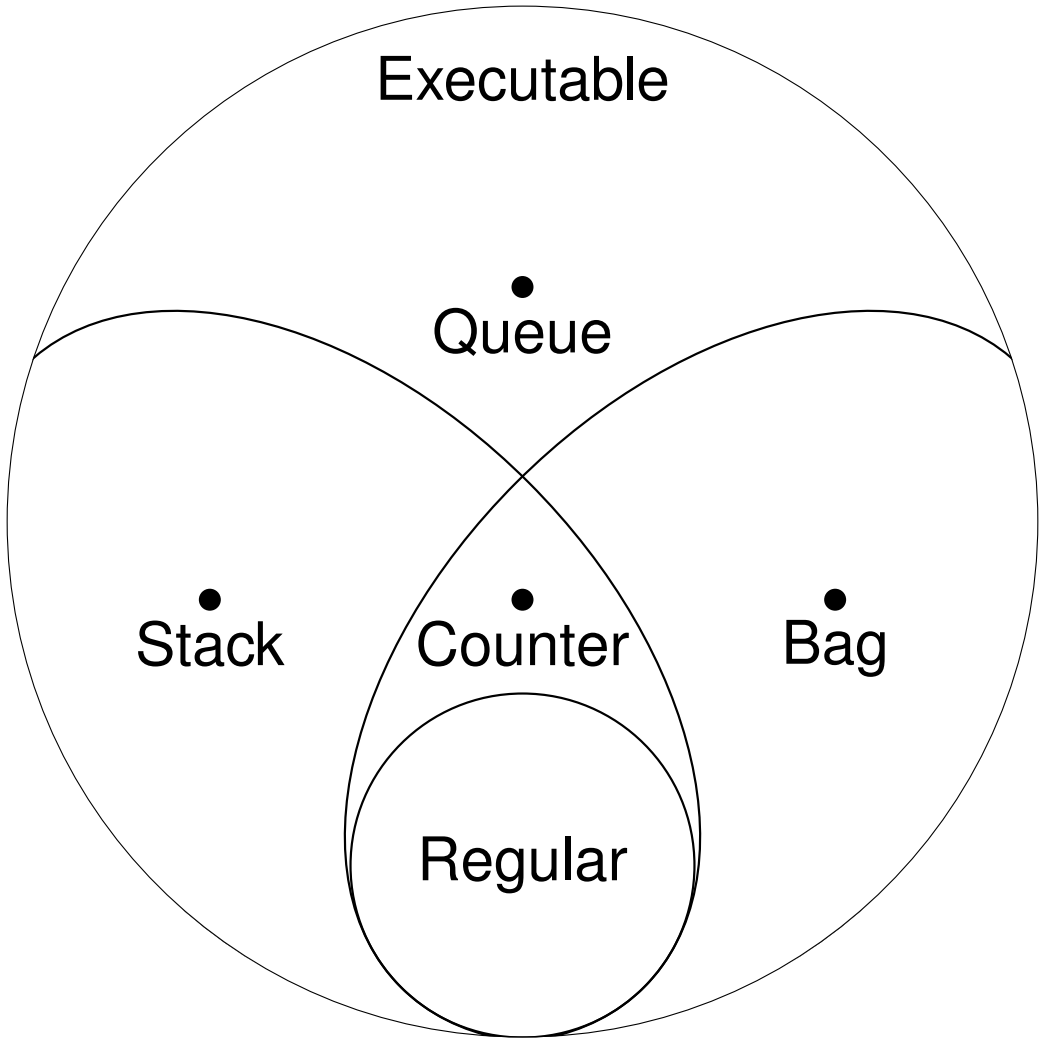
$$S^i = \mathbf{1} + i!\varepsilon.S^i + \sum_{d\in\mathcal{D}} i?d.\tau_{\mathcal{C}}(\partial_j(B_d^{ij} \parallel S^j))$$

$$B_d^{ij} = i!d.(j?\varepsilon.B_\varepsilon^{ij} + \sum_{e\in\mathcal{D}} j?e.B_e^{ij}) + \sum_{e\in\mathcal{D}} i?e.j!d.B_e^{ij}$$

$$B_\varepsilon^{ij} = \mathbf{1} + i!\varepsilon.B_\varepsilon^{ij} + \sum_{d\in\mathcal{D}} i?d.B_d^{ij}$$

CWI

Chomsky-Turing hierarchy



CWI

Conclusion

There is the book:
Models of Computation based on Automata: Formal Languages and Communicating Processes, pp. vi + 206.

CWI

Academic Publishing System is broken

We need to go to Diamond Open Access.

CWI

Academic Publishing System is broken

We need to go to Diamond Open Access.

Against my principles to publish with Springer.

CWI

Academic Publishing System is broken

We need to go to Diamond Open Access.

Against my principles to publish with Springer.

Preprint arXiv:2502.08345
Version of record https://ir.cwi.nl/pub/35037/35037.pdf