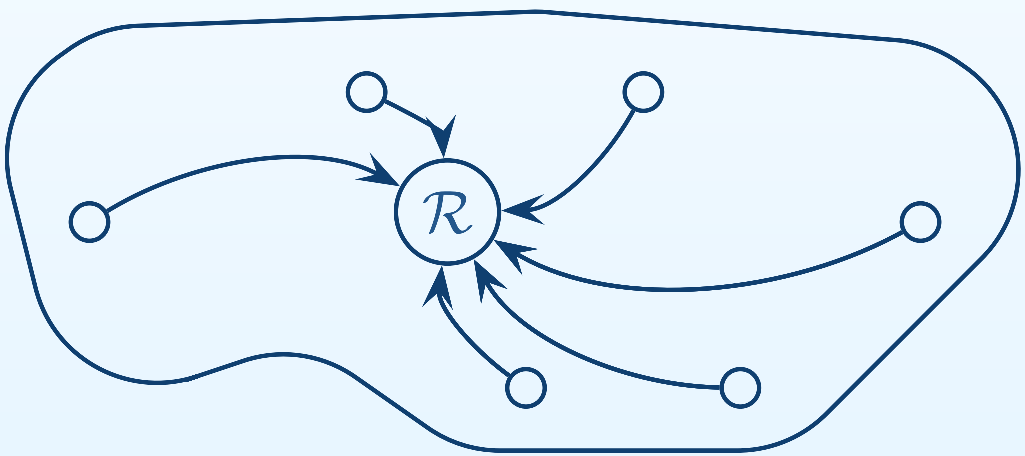
Symbolic Backwards Reachability Analysis of Higher Order Pushdown Systems

M. Hague and C.-H. L. Ong

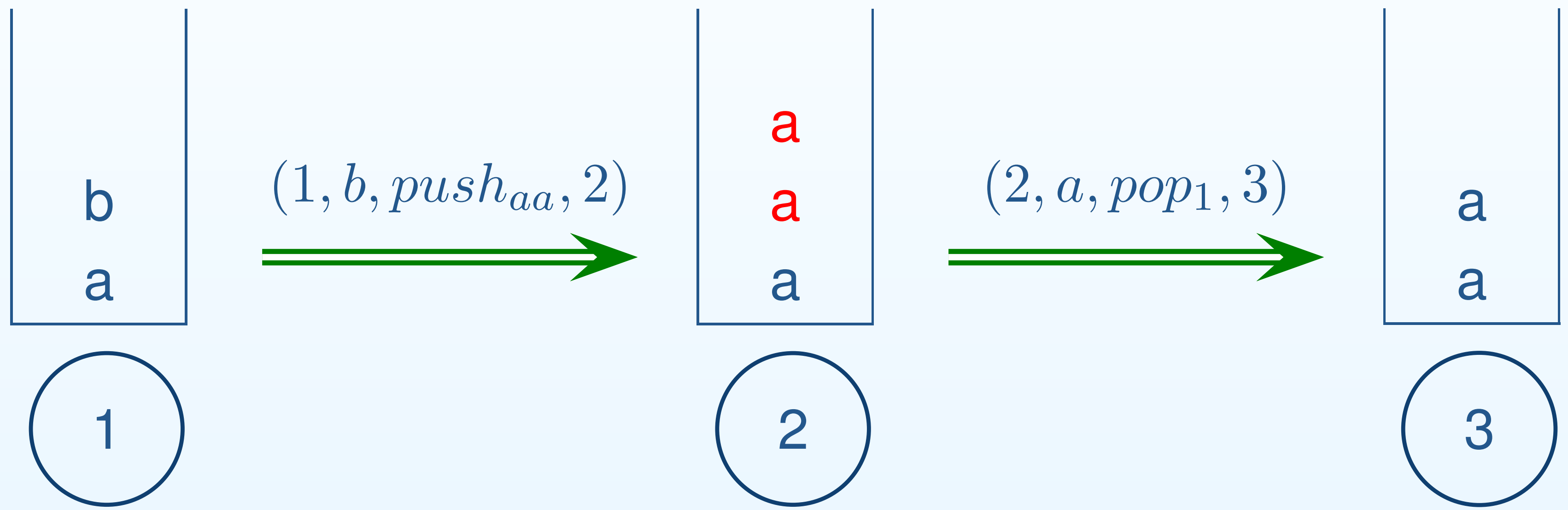
Abstract

- Pushdown systems — finite state + stack.
- Higher-order pushdown systems – stacks of stacks, etc.
- Backwards reachability analysis:



Pushdown Systems

Control states + order-one stack.

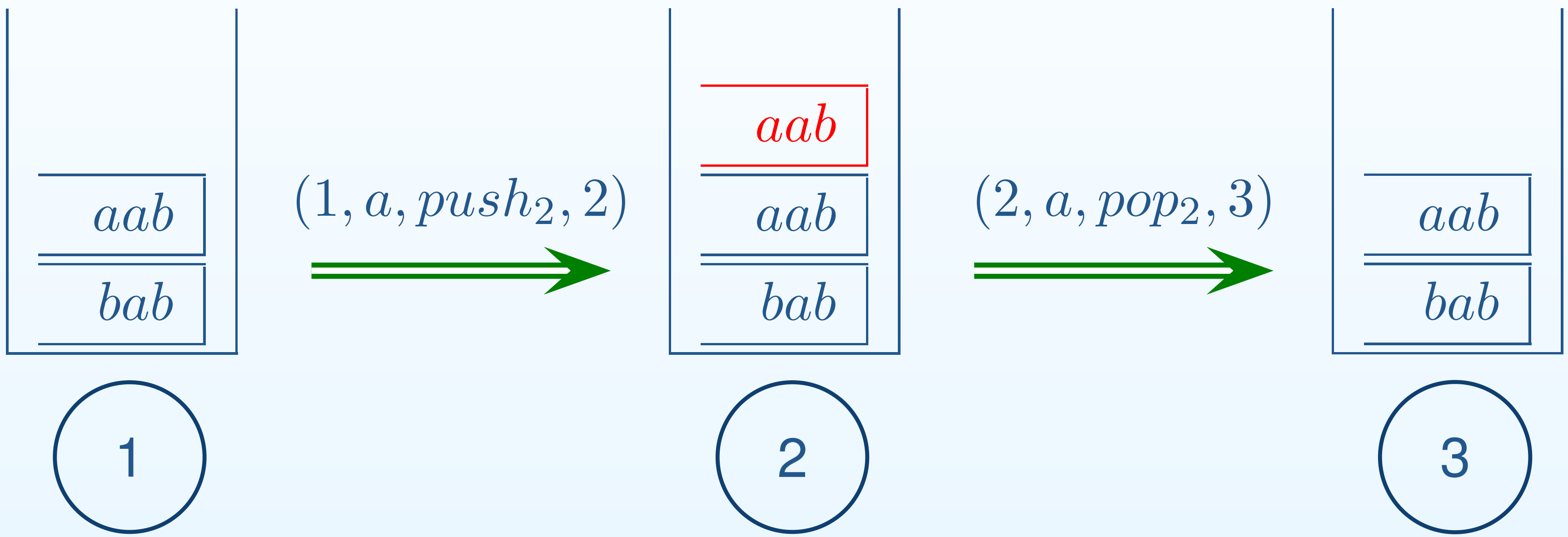


More behaviors than finite-state: $1^n 2^n \ldots$

Note $pop_1 = push_\varepsilon$.

Higher-Order Pushdown Systems

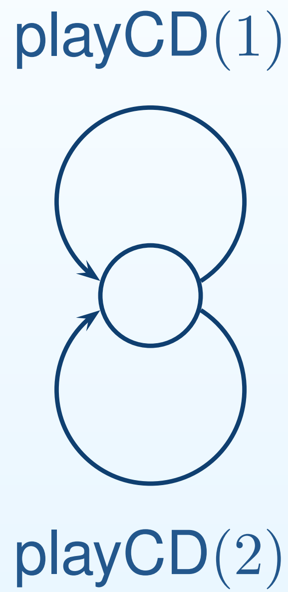
Control state + order-n stack.



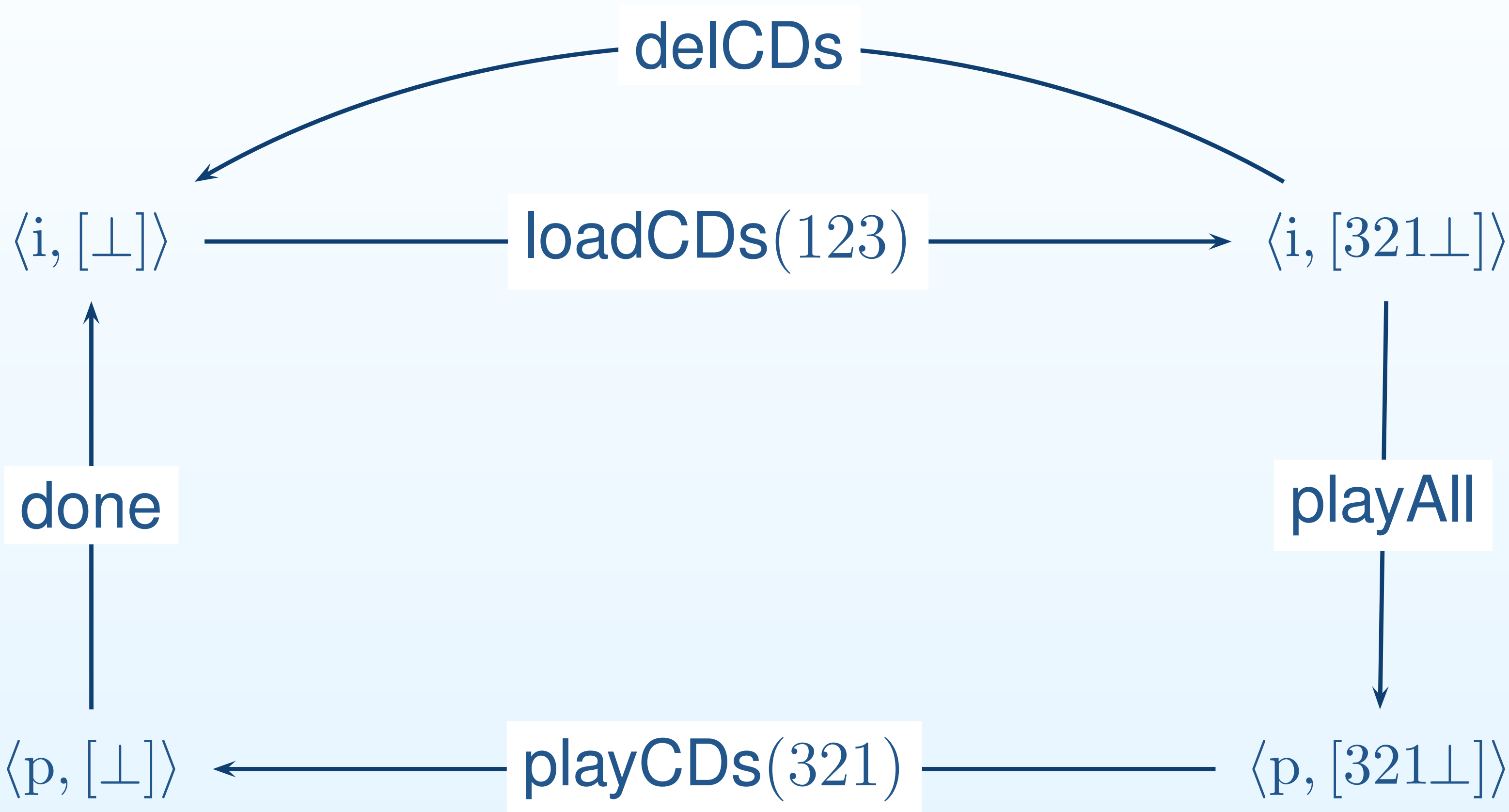
More behaviors than order-1: $1^n 2^n 3^n \dots$

Order-n PDSs form a strict hierarchy.

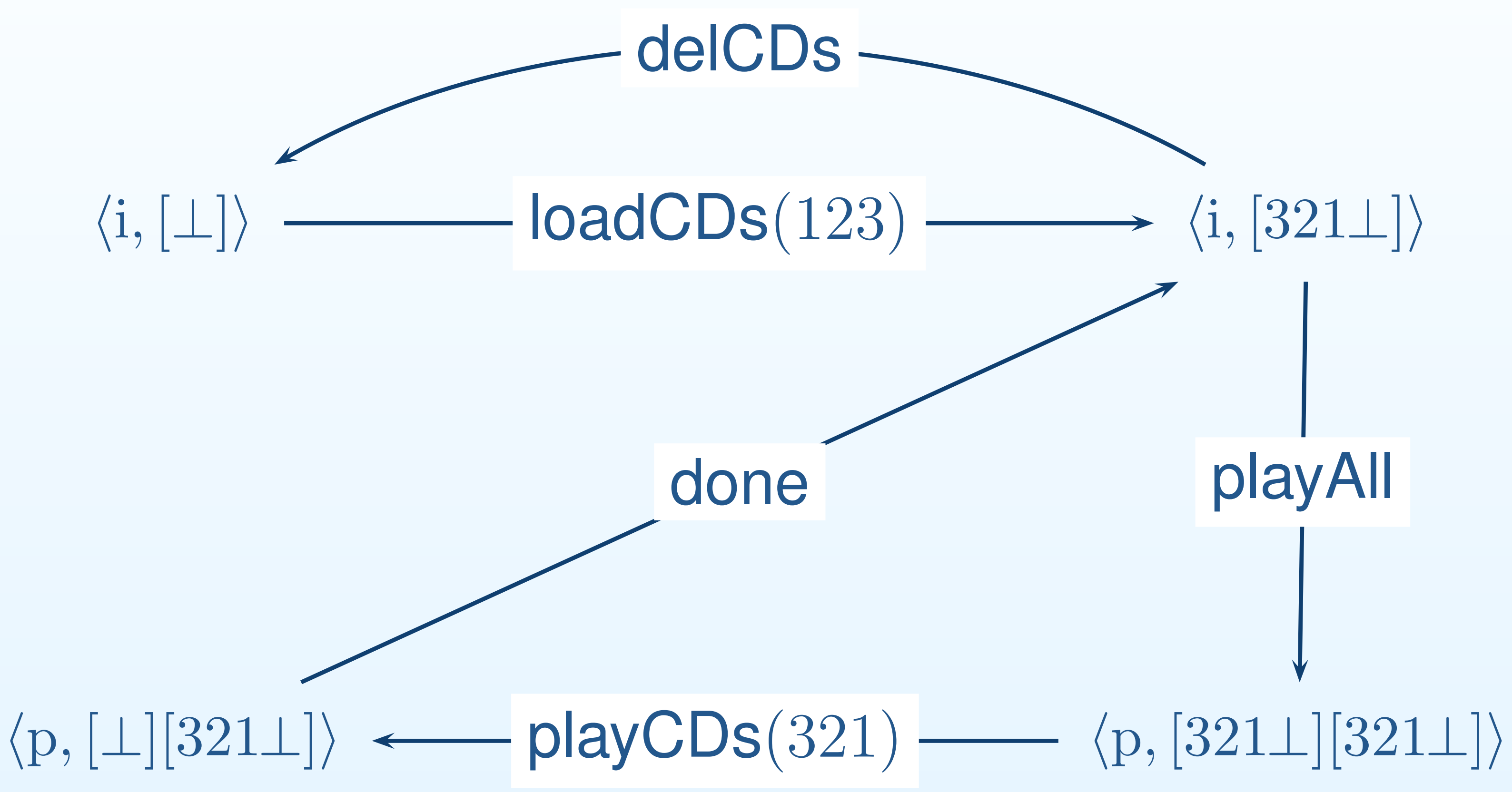
A Finite State CD Player



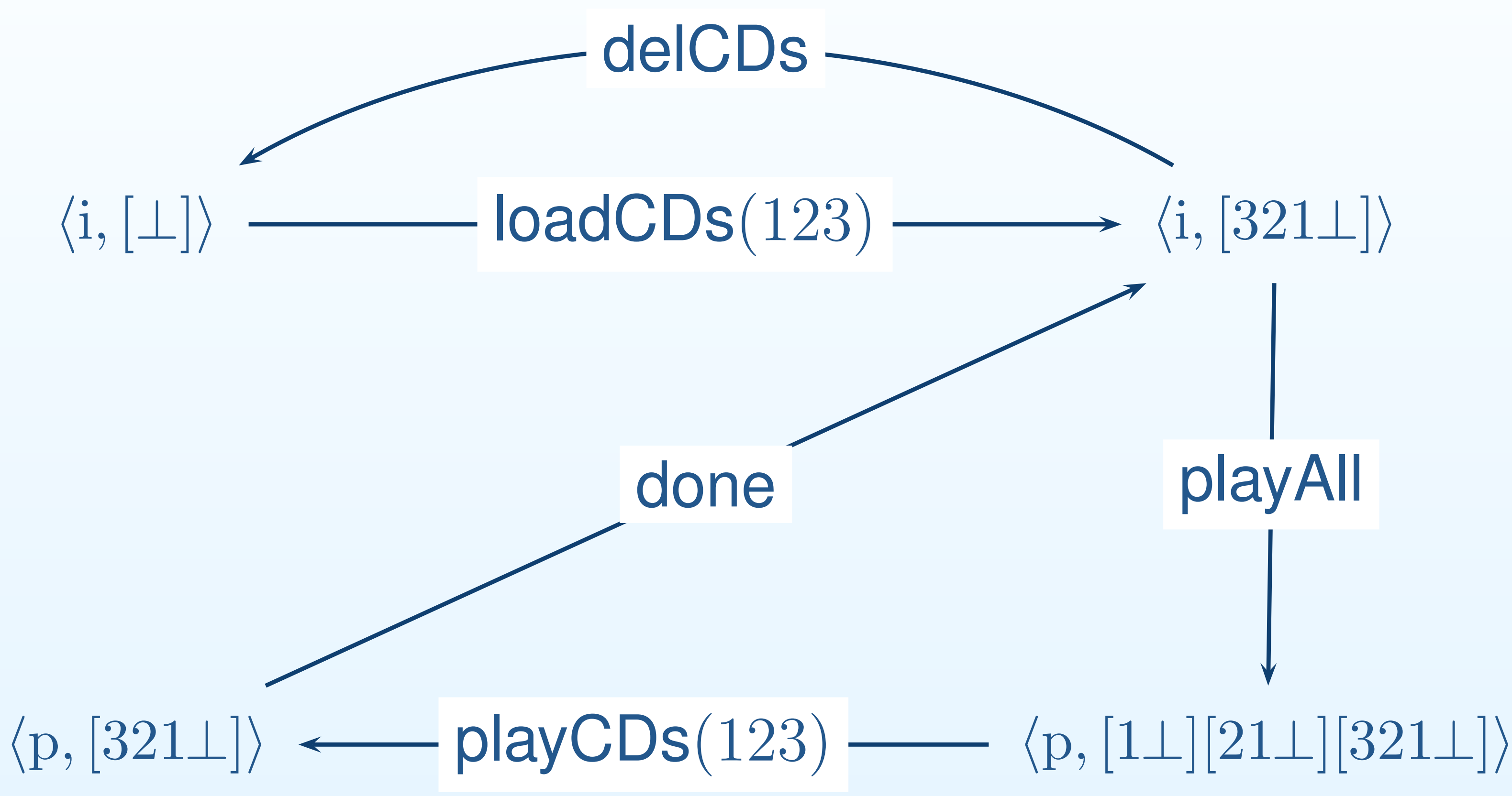
An Order-1 CD Player



An Order-2 CD Player



An Order-2 CD Player (Improved)



Model Checking and Games

From a single configuration we can determine:

- Parity games (order-one) [Walukiewicz, 1996].
- More complicated winning conditions (order-one) [Cachat, Duparc, Thomas, 2002], [Bouquet, Serre, Walukiewicz, 2003], [Gimbert 2004].
- Parity games (higher-order) [Cachat, 2003].

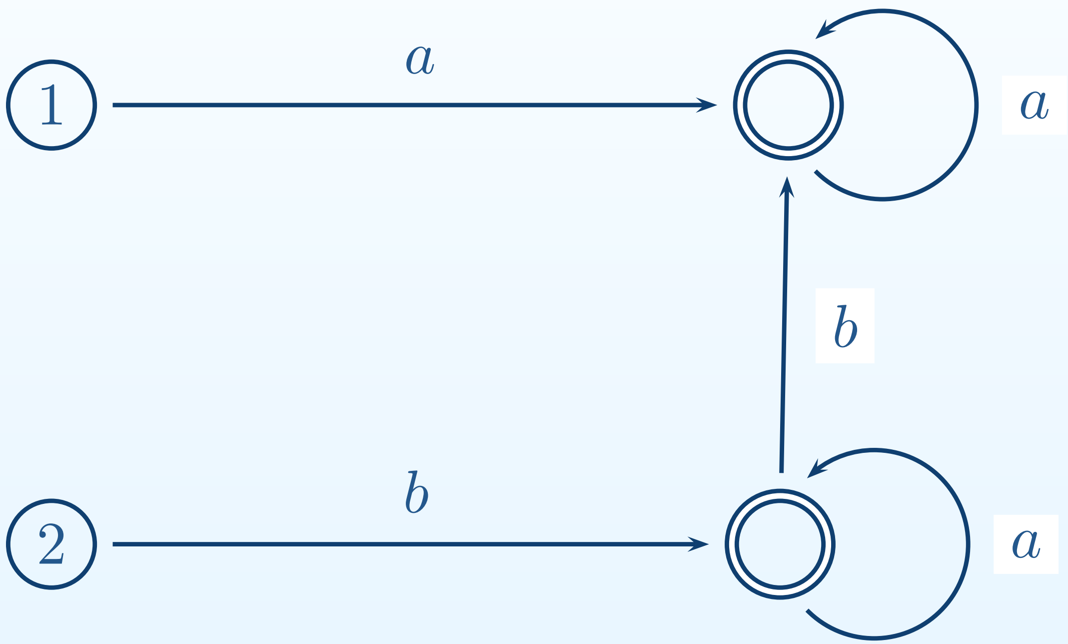
Model Checking and Games

We can determine the winning regions of:

- Reachability games (order-one) [Bouajjani, Esparza, Maler, 1997].
- Büchi and Parity games (order-one) [Cachat, 2002/03][Serre, 2004].
- Single player/control state reachability (higher-order) [Bouajjani, Meyer, 2004].
- Reachability games (higher-order) [Hague, Ong, 2007].

Reachability Over Pushdown Systems

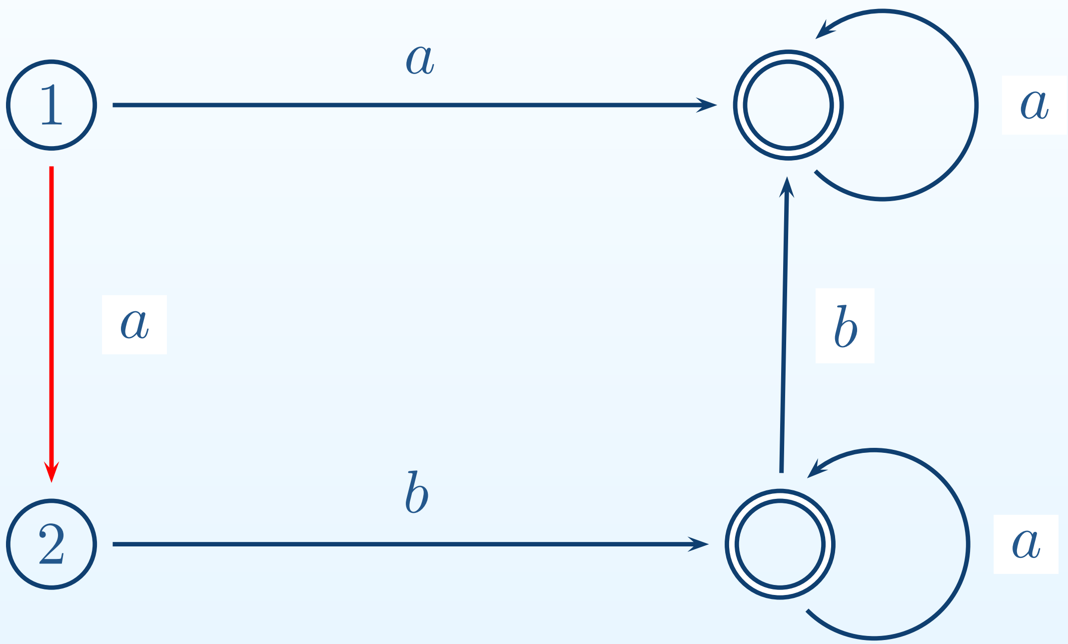
Determine the set of configurations that can reach $\mathcal{R}$.



$$\langle 1, aa^* \rangle, \langle 2, ba^* \rangle, \langle 2, ba^*ba^* \rangle$$

Reachability Over Pushdown Systems

Determine the set of configurations that can reach $\mathcal{R}$.

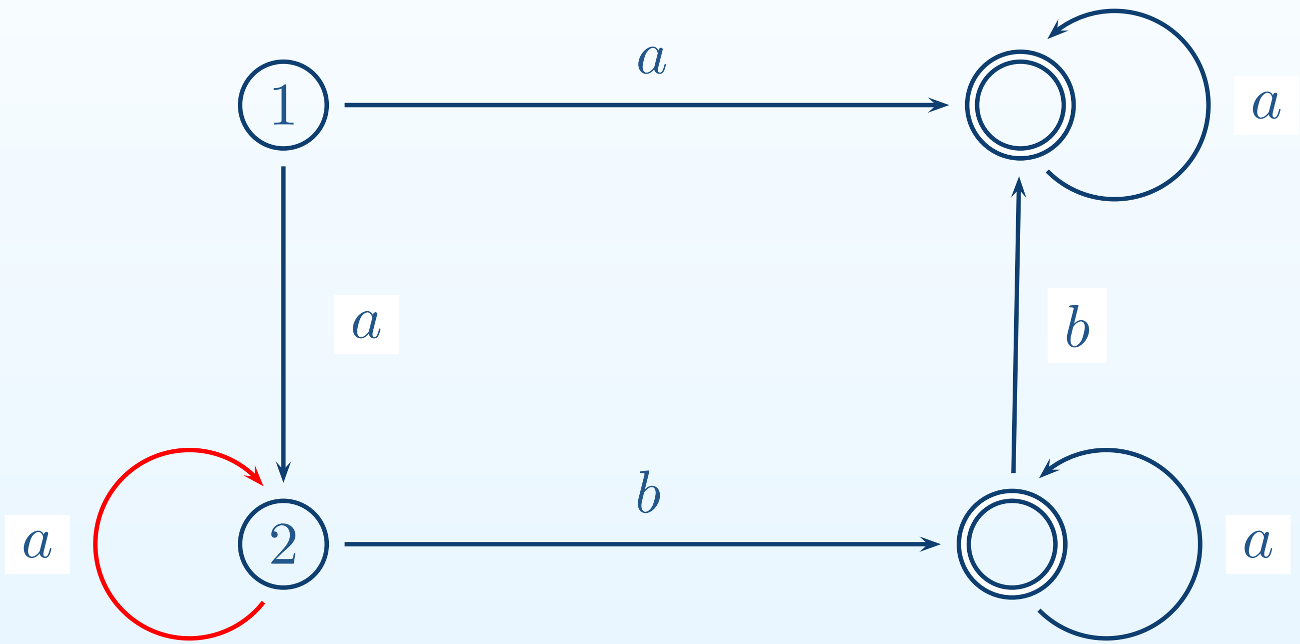


$$(1, a, pop_1, 2)$$

$$\langle 1, aba^* \rangle \hookrightarrow \langle 2, ba^* \rangle$$

Reachability Over Pushdown Systems

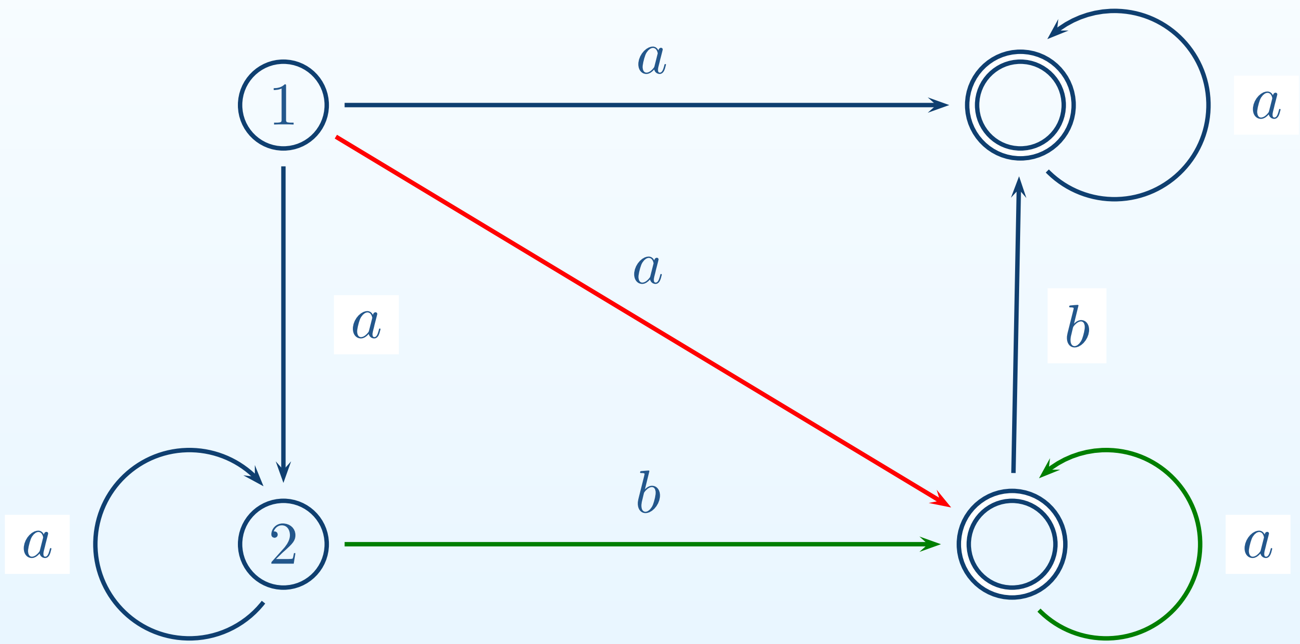
Determine the set of configurations that can reach $\mathcal{R}$.



$(2, a, pop_1, 2)$

Reachability Over Pushdown Systems

Determine the set of configurations that can reach $\mathcal{R}$.

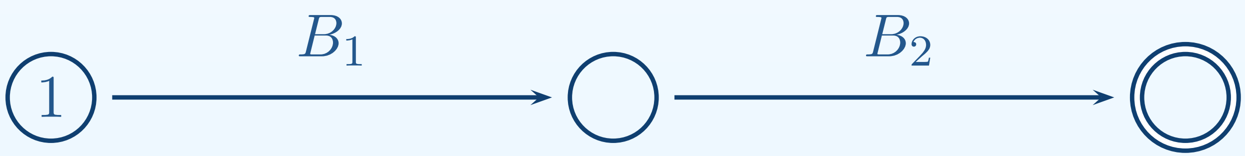


$$(1, a, push_{ba}, 2)$$

$$\langle 1, aa^* \rangle \hookrightarrow \langle 2, baa^* \rangle$$

Order-2 Pushdown Systems (Single Control State)

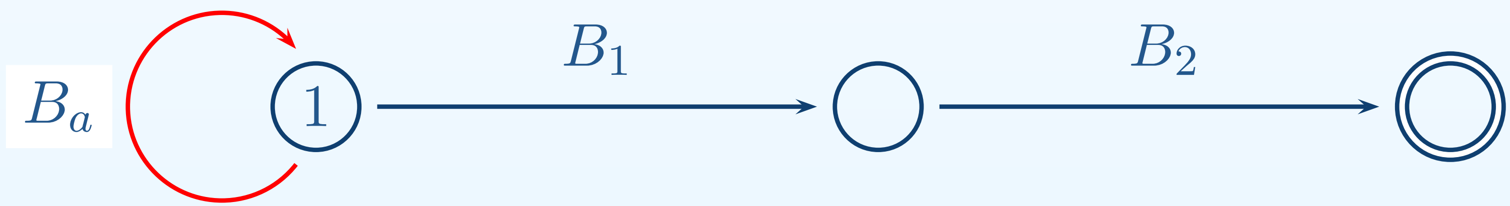
Nested store automata — transitions labelled by automata accepting order-1 stacks.



$$\langle 1, [u][v] \rangle \quad u \in \mathcal{L}(B_1), v \in \mathcal{L}(B_2)$$

Order-2 Pushdown Systems (Single Control State)

Nested store automata — transitions labelled by automata accepting order-1 stacks.

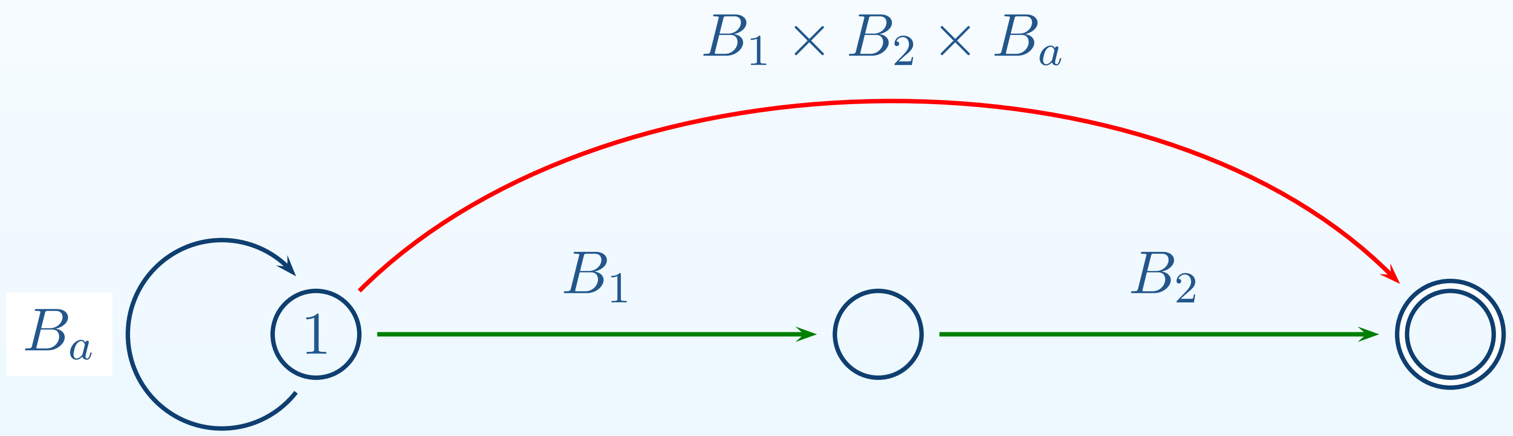


$$(a, pop_2)$$

$$\langle 1, [w][u][v] \rangle \hookrightarrow \langle 1, [u][v] \rangle$$

Order-2 Pushdown Systems (Single Control State)

Nested store automata — transitions labelled by automata accepting order-1 stacks.

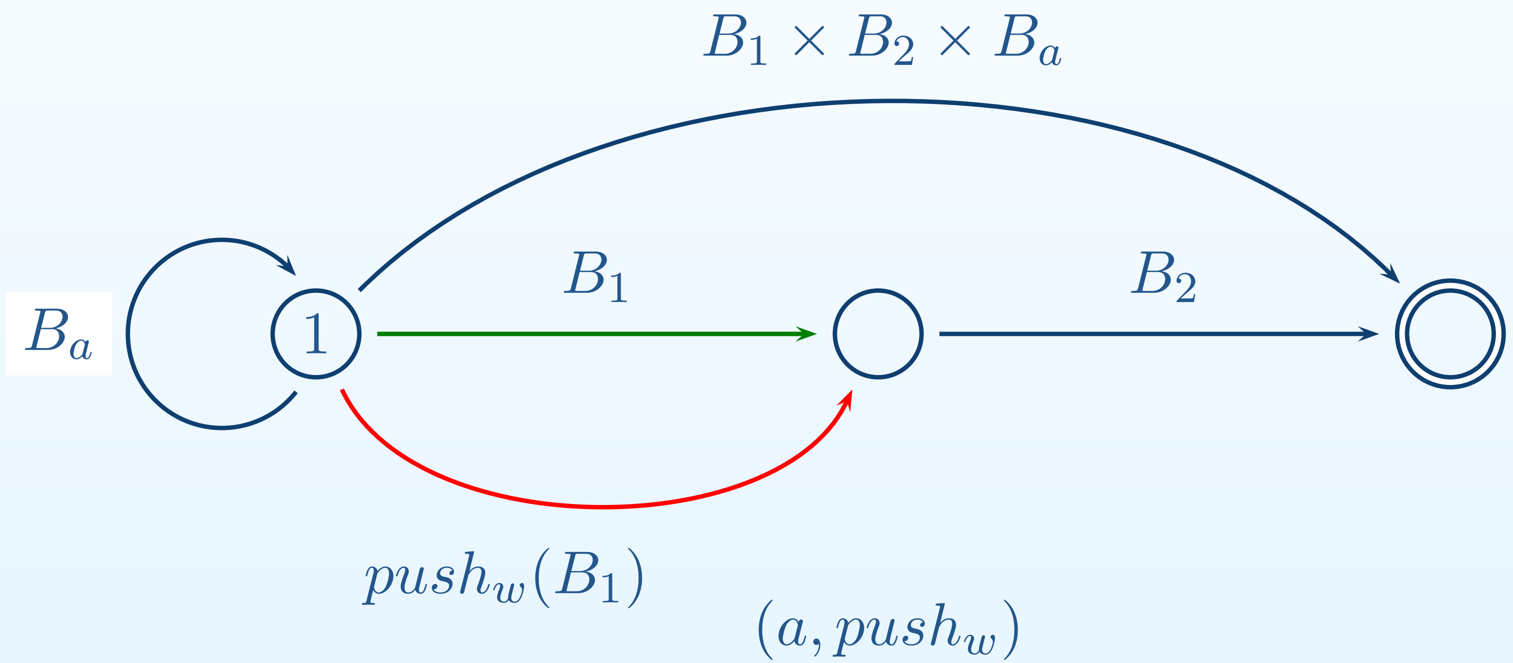


$$(a, push_2)$$

$$\langle 1, [w] \rangle \hookrightarrow \langle 1, [w][w] \rangle$$

Order-2 Pushdown Systems (Single Control State)

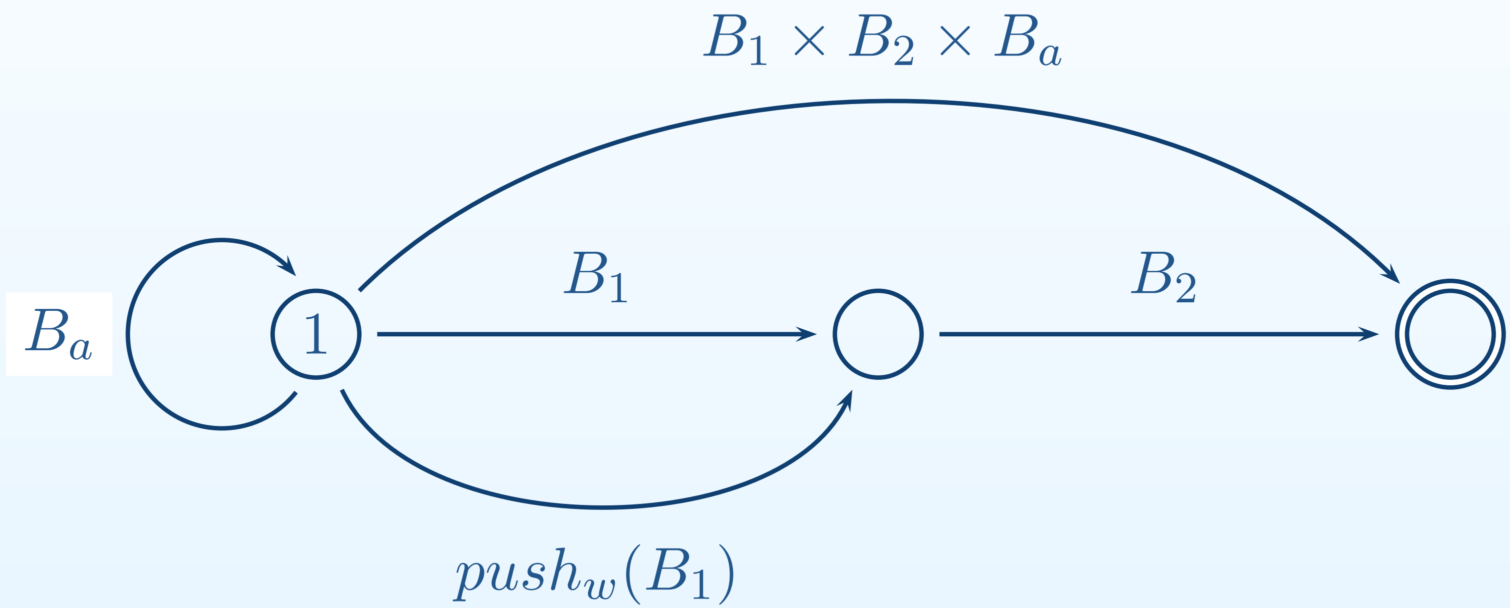
Nested store automata — transitions labelled by automata accepting order-1 stacks.



$$\langle 1, [au][v] \rangle \hookrightarrow \langle 1, [wu][v] \rangle$$

Order-2 Pushdown Systems (Single Control State)

Nested store automata — transitions labelled by automata accepting order-1 stacks.

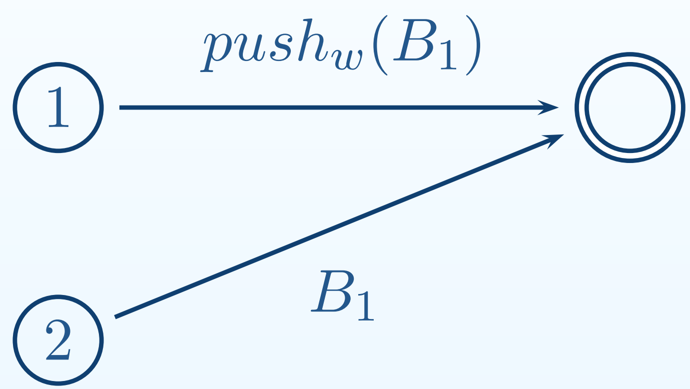


Finite number of order-1 automata wrt state-space,

$$\mathcal{Q}_a \times \mathcal{Q}_1 \times \mathcal{Q}_2$$

Multiple Control States

A command $(1, a, push_w, 2)$.



Let $\mathcal{L}(B_1) = w$ $\qquad \langle 2, [w] \rangle$

We require $\mathcal{L}(push_w(B_1)) = a$ $\qquad \langle 1, [a] \rangle \hookrightarrow \langle 2, [w] \rangle$

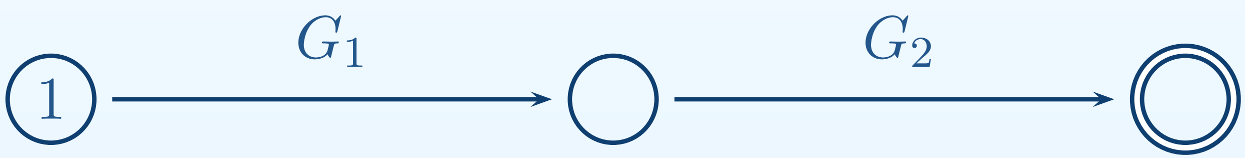
But $\mathcal{L}(push_w(B_1)) = a \cup w$ $\qquad \langle 1, [w] \rangle \hookrightarrow ???$

Adding new states breaks the termination argument.

Multiple Control States

Commands $(1, a, pop_1, 1), (1, a, push_2, 1)$.

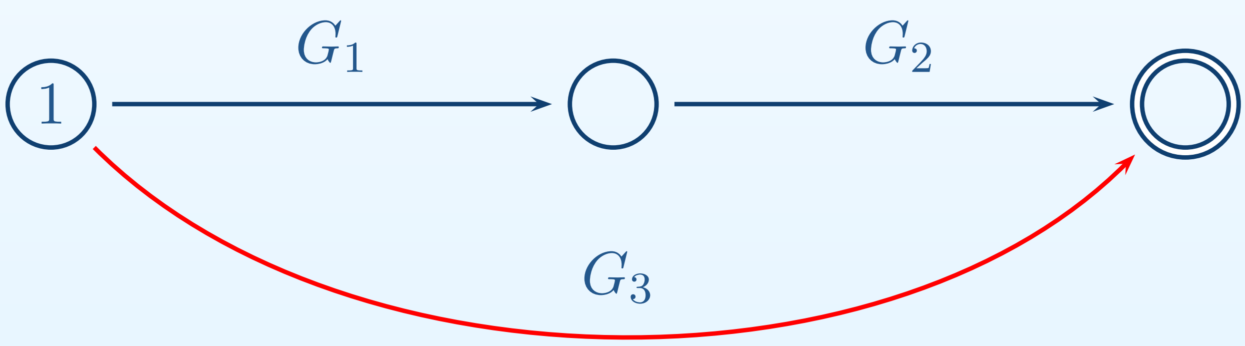
$$
\begin{aligned}
G_1 &= B_1 \\
G_2 &= B_2
\end{aligned}
$$



Multiple Control States

Commands $(1, a, pop_1, 1), (1, a, push_2, 1)$.

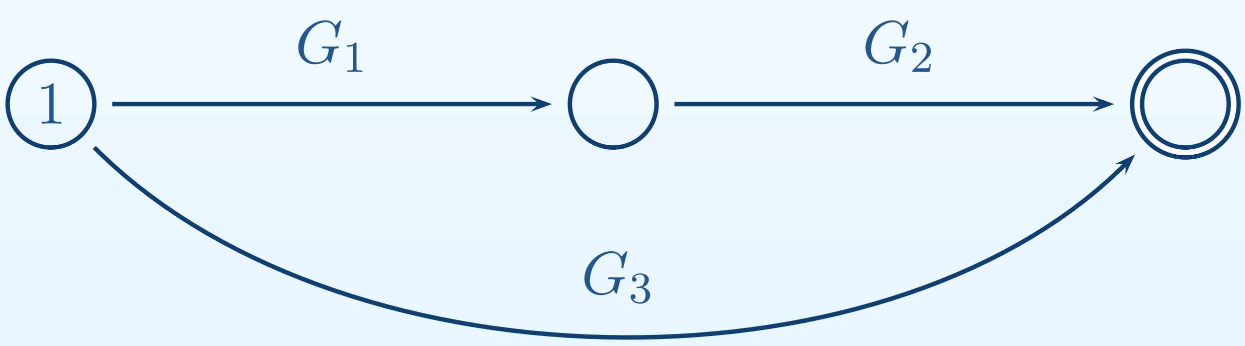
$$
\begin{array}{lcl|l}
G_1 & = & B_1 & pop_1(G_1), B_1 \\
G_2 & = & B_2 & B_2 \\
G_3 & = & & G_1 \times G_2
\end{array}
$$



Multiple Control States

Commands $(1, a, pop_1, 1), (1, a, push_2, 1)$.

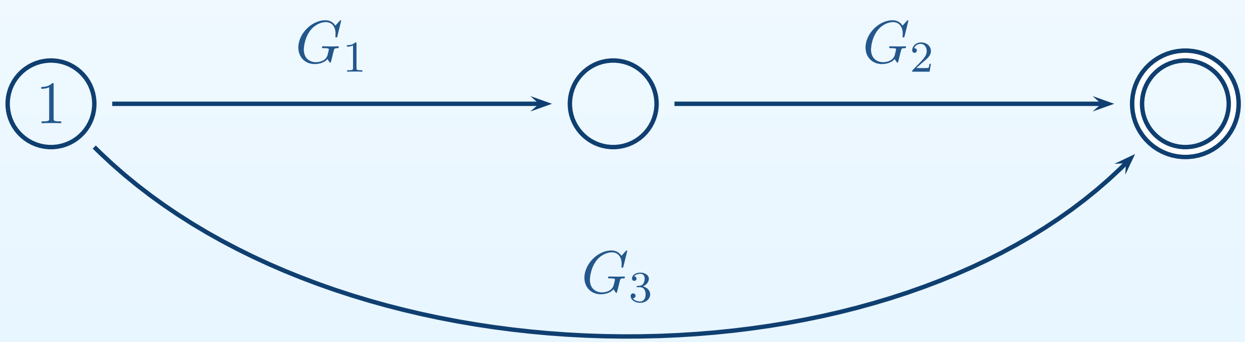
G_1	$=$	B_1	$pop_1(G_1), B_1$	$pop_1(G_1), B_1$
G_2	$=$	B_2	B_2	B_2
G_3	$=$		$G_1 \times G_2$	$pop_1(G_3), G_1 \times G_2$



Multiple Control States

Commands $(1, a, pop_1, 1), (1, a, push_2, 1)$.

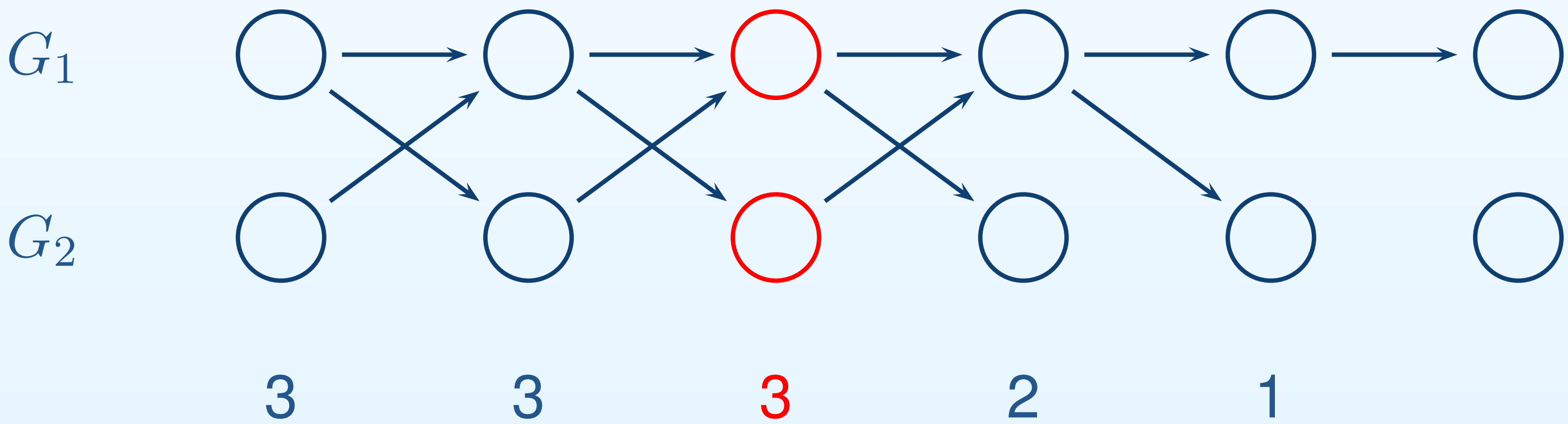
$$
\begin{array}{lcl:l:l:l}
G_1 & = & & pop_1(G_1), B_1 & pop_1(G_1), B_1 & \\
G_2 & = & \ldots & B_2 & B_2 & \ldots \\
G_3 & = & & pop_1(G_3), G_1 \times G_2 & pop_1(G_3), G_1 \times G_2 &
\end{array}
$$



Multiple Control States

Constructing the order-1 automata:

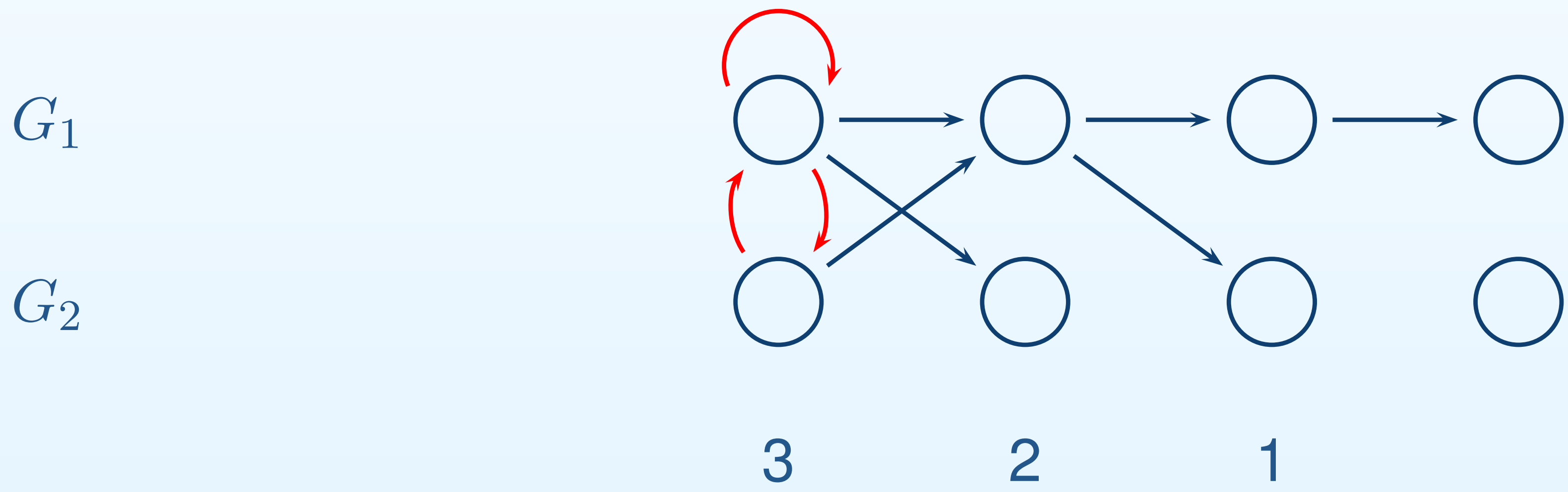
$$
\begin{array}{lcl}
G_1 & = & \alpha_1 \mid \alpha_2 \mid \alpha_3 \mid \alpha_3 \mid \alpha_3 \mid \ldots \\
G_2 & = & \beta_1 \mid \beta_2 \mid \beta_3 \mid \beta_3 \mid \beta_3 \mid \ldots
\end{array}
$$



Multiple Control States

Constructing the order-1 automata:

$$G_1 = \alpha_1 \mid \alpha_2 \mid \alpha_3 \mid \alpha_3 \mid \alpha_3 \mid \ldots$$
$$G_2 = \beta_1 \mid \beta_2 \mid \beta_3 \mid \beta_3 \mid \beta_3 \mid \ldots$$



Applications

Higher-order pushdown systems:

- Model checking linear time properties.
- Model checking alternation-free μ-Calculus.

Games over higher-order systems:

- Reachability games.
- Büchi games?

Higher-order pushdown automata:

- Non-emptiness
  - Shows n-EXPTIME-completeness of reachability.

Conclusions and Future Work

Conclusions:

- An (optimal) n-EXPTIME algorithm for reachability games over higher-order pushdown systems.
- Applications to model-checking.

Future Work:

- Parity games.
- Implementation / alternative techniques.
  - Other notions of regularity [Carayol, 2004].
  - Grammars.
- Extensions of pushdown systems.