Model Checking Temporal Properties of Probabilistic Recursive Programs

Tobias Winkler, Christina Gehnen, Joost-Pieter Katoen

FoSSaCS 2022

UnRAVeL | RWTH AACHEN UNIVERSITY | Software Modeling and Verification Chair | RWTH AACHEN UNIVERSITY

LTL Model Checking for Transition Systems

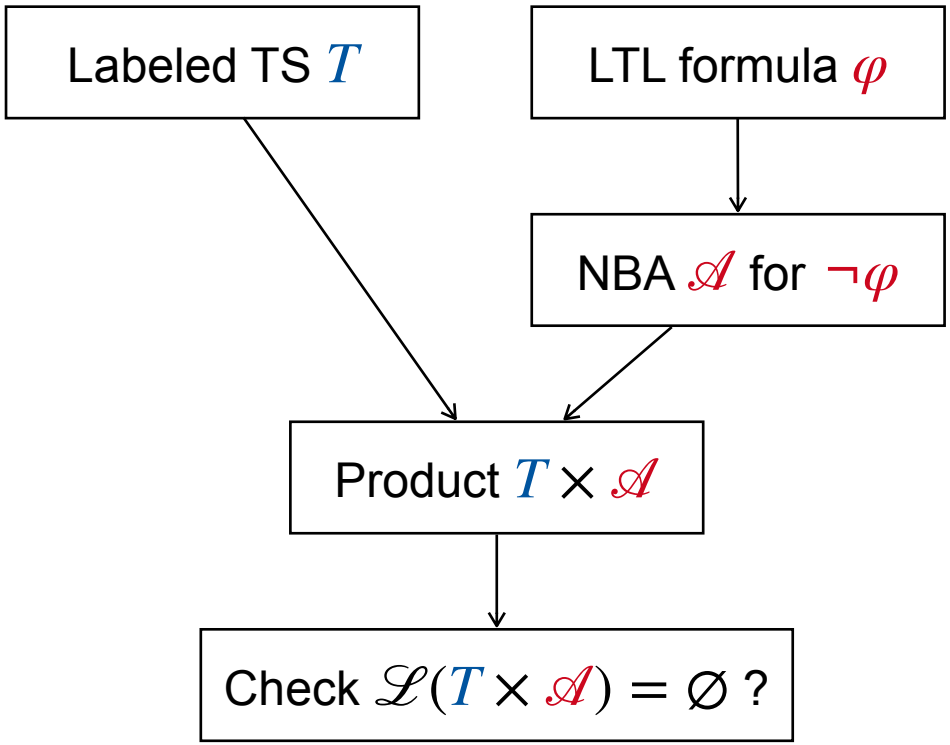
[Vardi & Wolper '86]

Labeled TS T

LTL formula φ

LTL Model Checking for Transition Systems

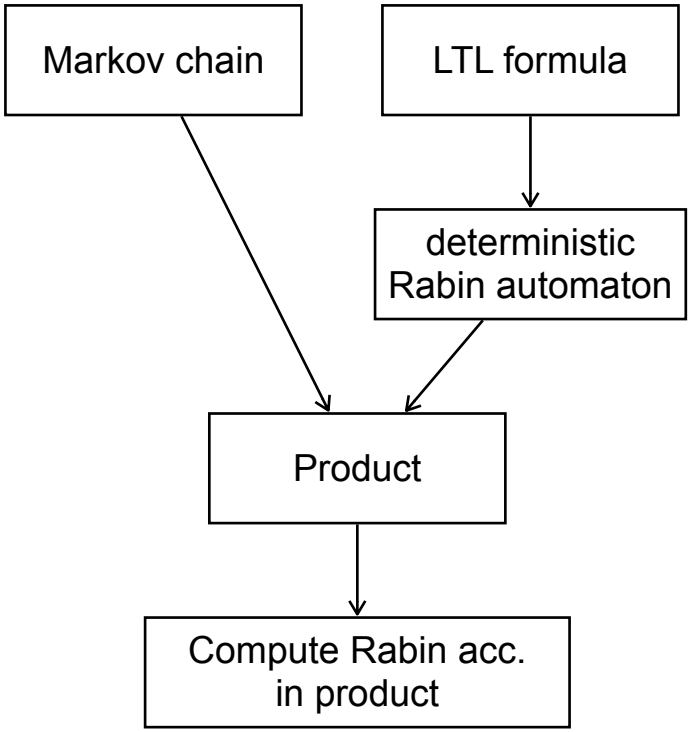
[Vardi & Wolper '86]



Extensions of LTL Model Checking

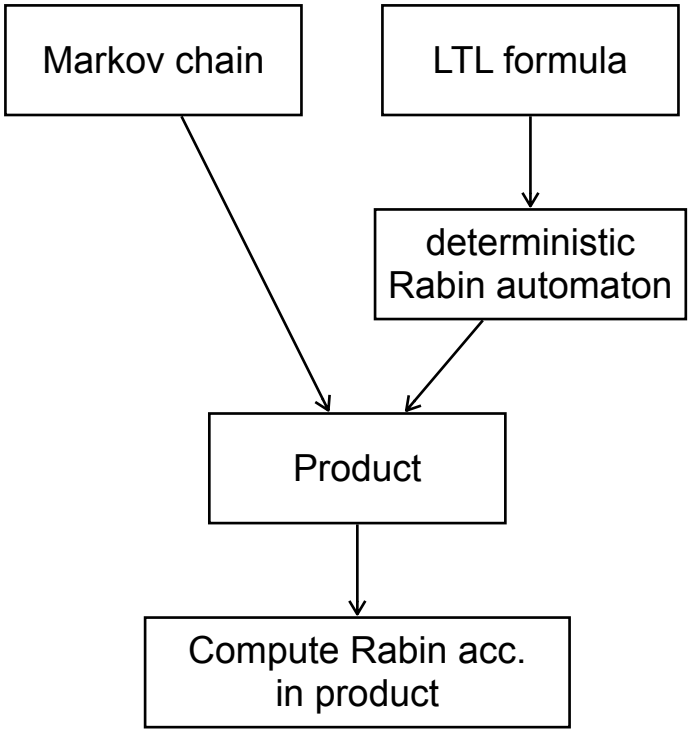
Extensions of LTL Model Checking

Finite-state probabilistic programs
[Vardi & Wolper, Courcoubetis & Yannakakis]

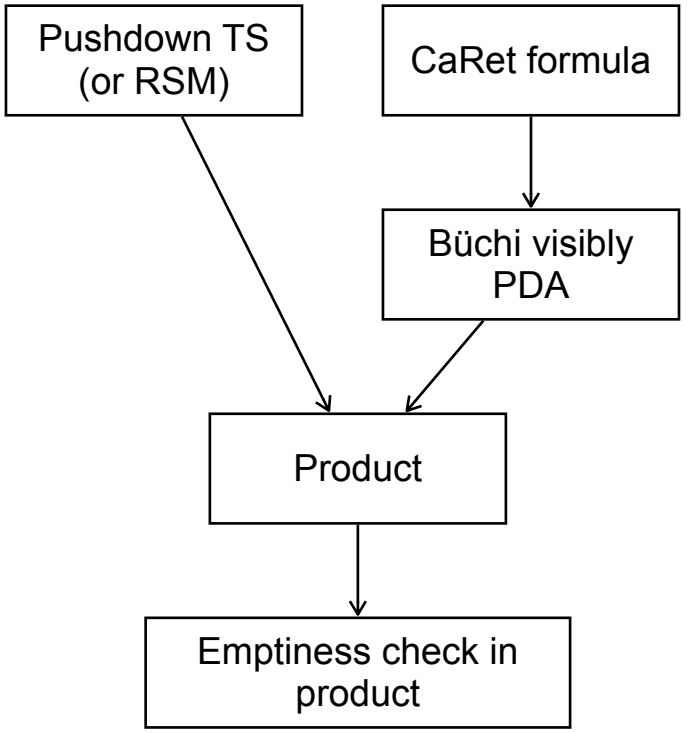


Extensions of LTL Model Checking

Finite-state probabilistic programs
[Vardi & Wolper, Courcoubetis & Yannakakis]

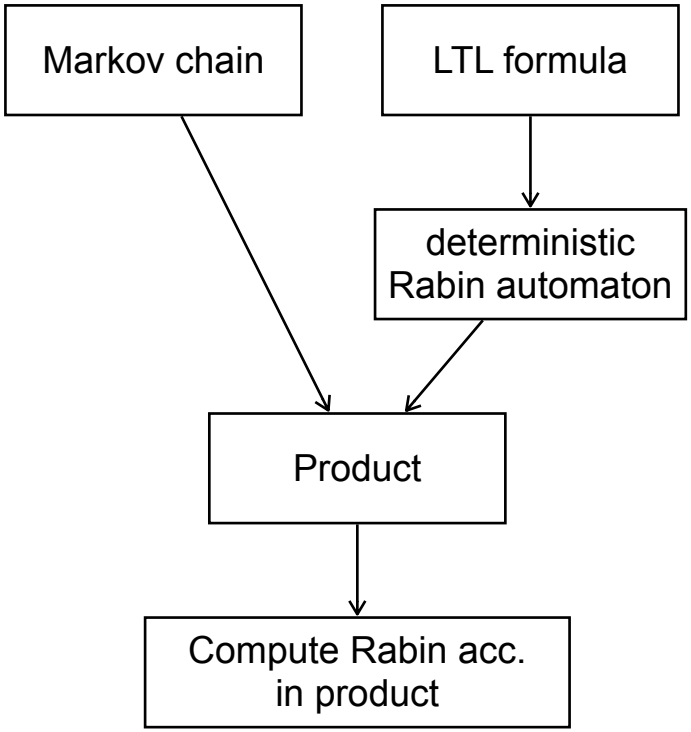


Finite-state procedural programs [Alur et al.]

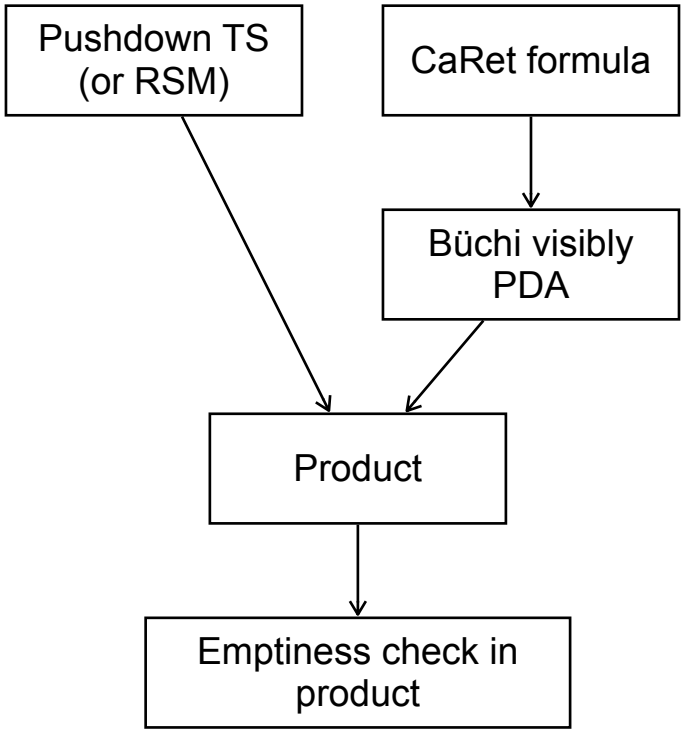


Extensions of LTL Model Checking

Finite-state probabilistic programs
[Vardi & Wolper, Courcoubetis & Yannakakis]

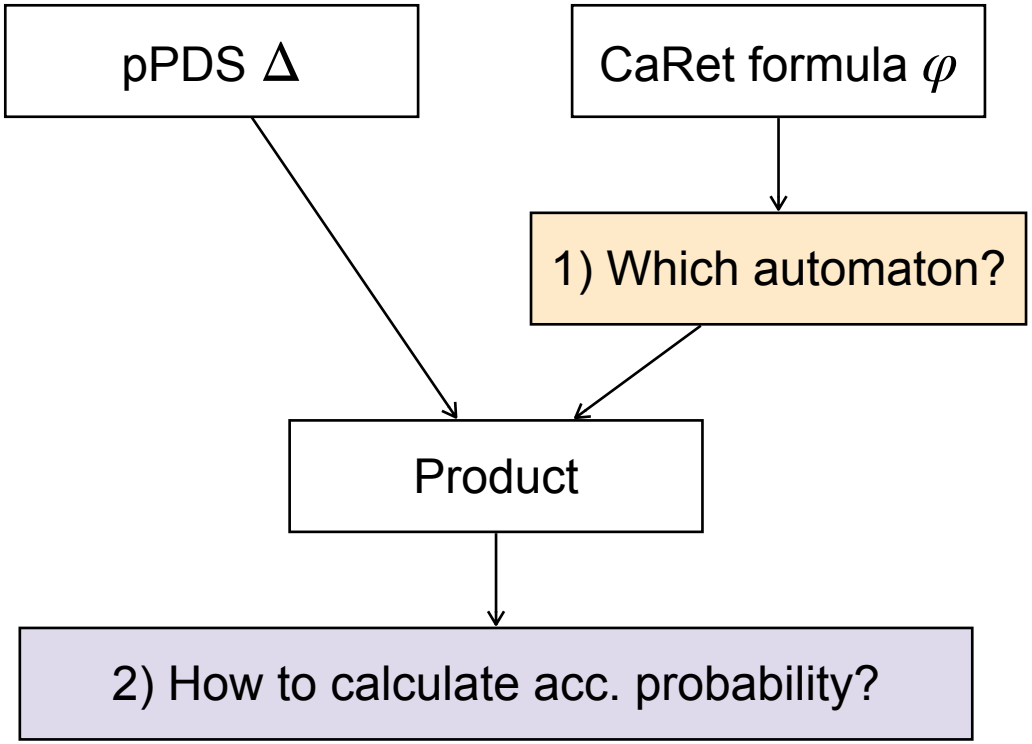


Finite-state procedural programs [Alur et al.]

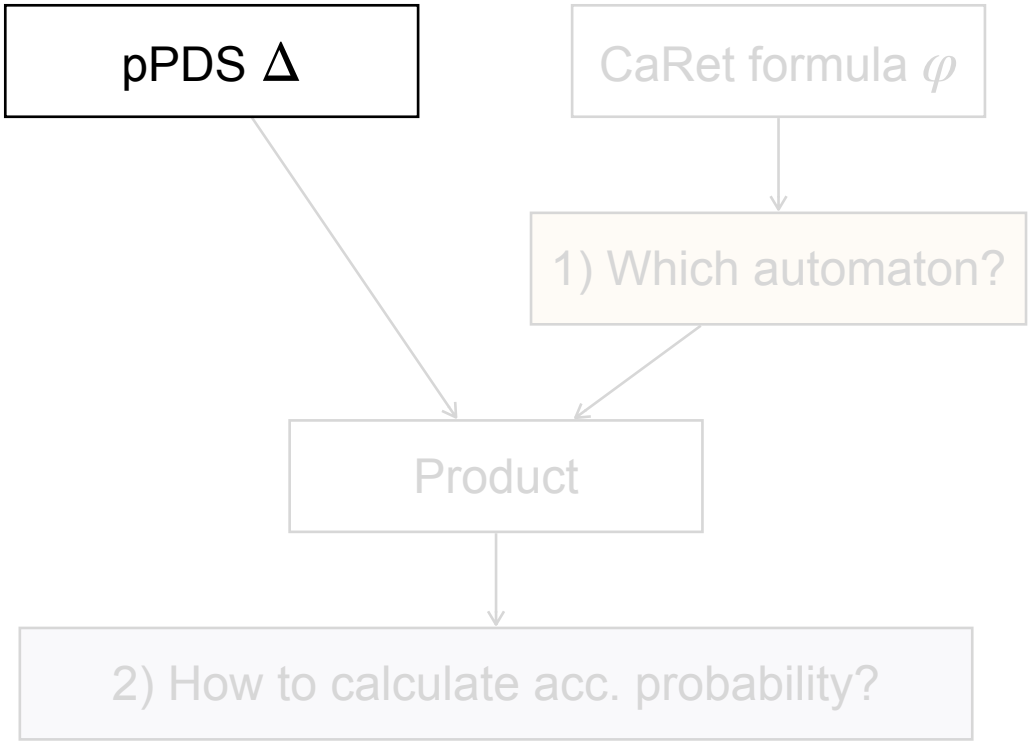


Our paper: Combine these two

Our Paper: CaRet Model Checking for Probabilistic Pushdown Systems

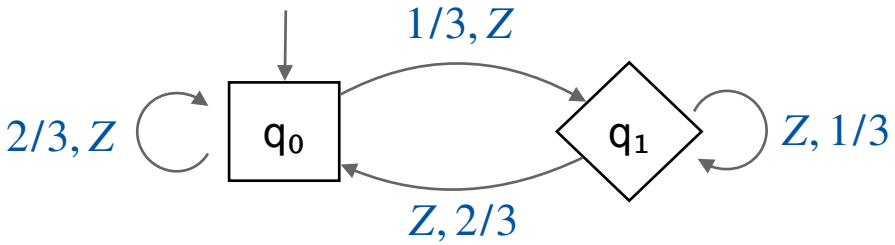


Our Paper: CaRet Model Checking for Probabilistic Pushdown Systems



Probabilistic Pushdown Systems

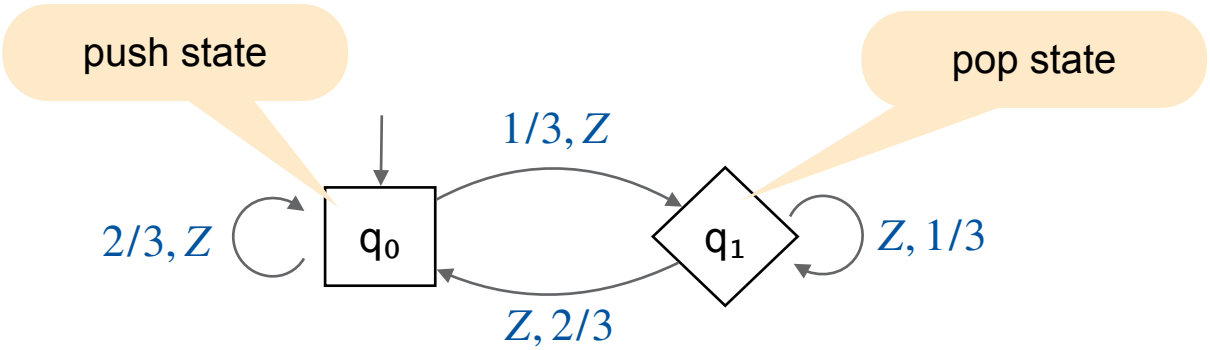
[Esparza, Kucera, Mayr '04]



Stack alphabet: $\Gamma = \{Z\}$

Probabilistic Pushdown Systems

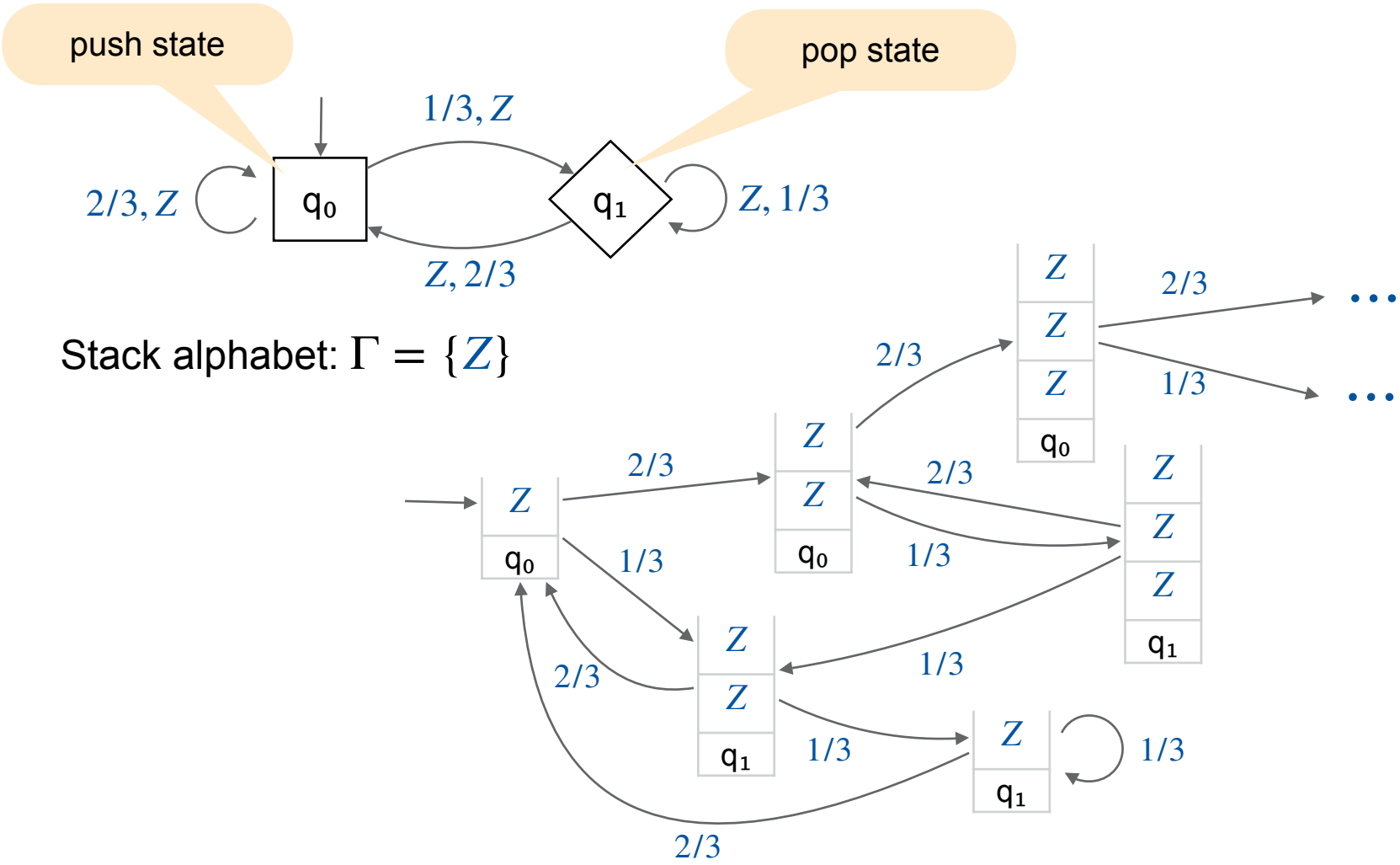
[Esparza, Kucera, Mayr '04]



Stack alphabet: $\Gamma = \{Z\}$

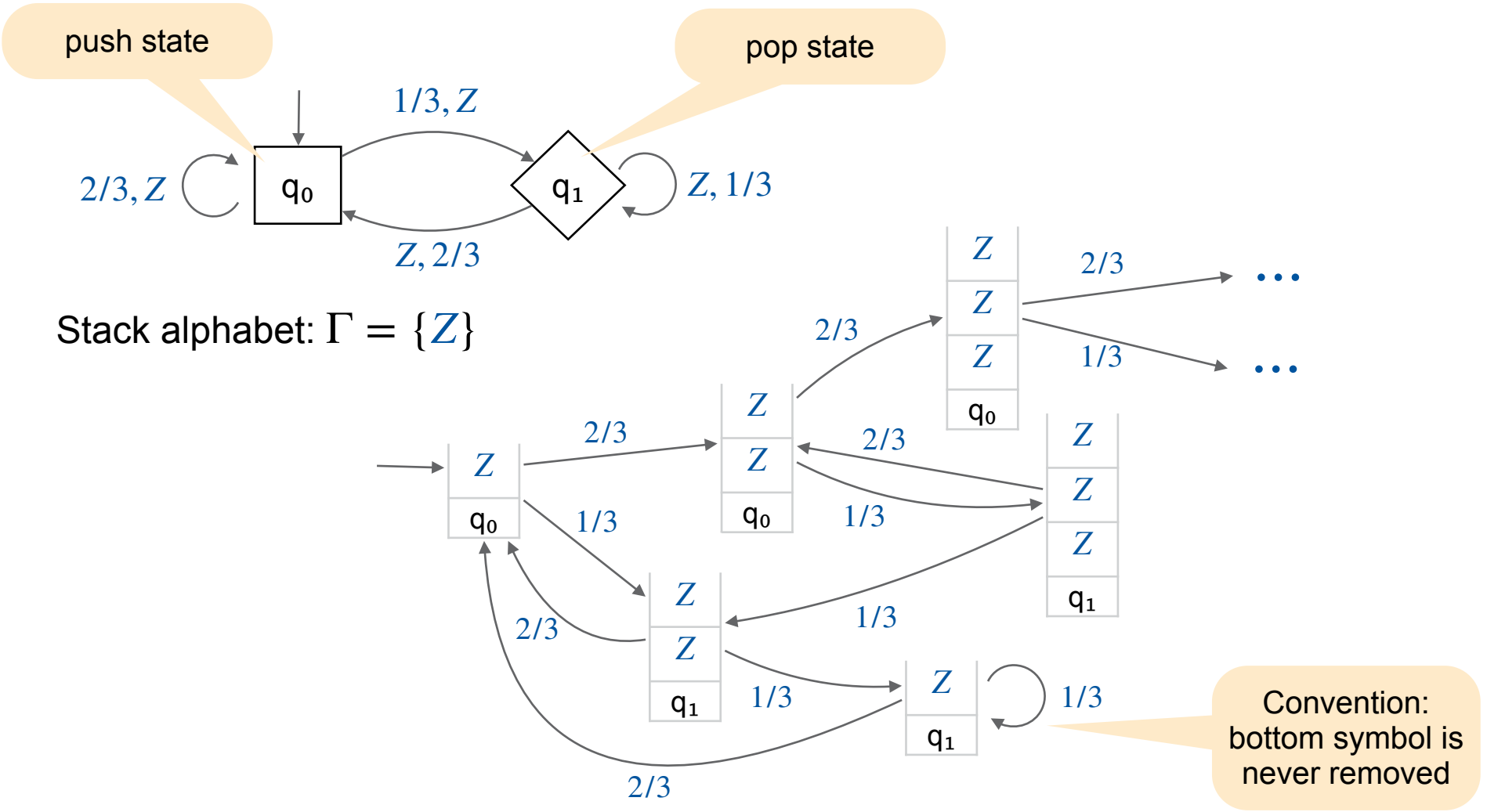
Probabilistic Pushdown Systems

[Esparza, Kucera, Mayr '04]



Probabilistic Pushdown Systems

[Esparza, Kucera, Mayr '04]



Probabilistic Pushdown Systems (pPDS)

Definition

A pPDS consists of

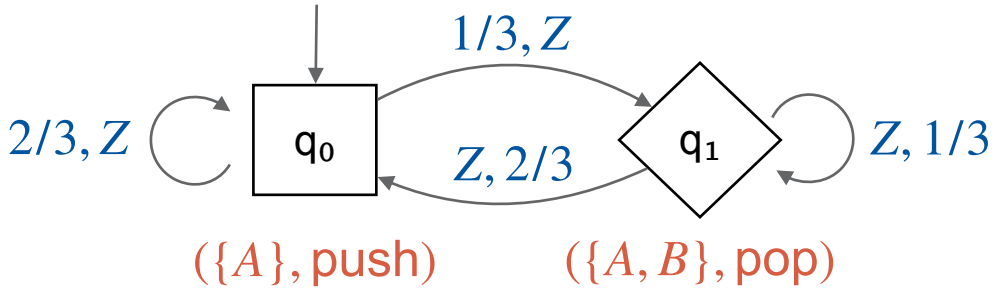
- *a finite set* $Q = (Q_{push}, Q_{int}, Q_{pop})$ *of states*
- *a finite stack alphabet* Γ
- *an initial configuration* $(q_0, Z_0) \in Q \times \Gamma$
- *(probabilistic) transitions*
  - $P_{push} \colon Q_{push} \to Dist(Q \times \Gamma)$
  - $P_{int} \colon \quad Q_{int} \to Dist(Q)$
  - $P_{pop} \colon \quad Q_{pop} \times \Gamma \to Dist(Q)$
- *a consistent labeling function* $Q \to 2^{AP} \times \{push, int, pop\}$

Probabilistic Pushdown Systems (pPDS)

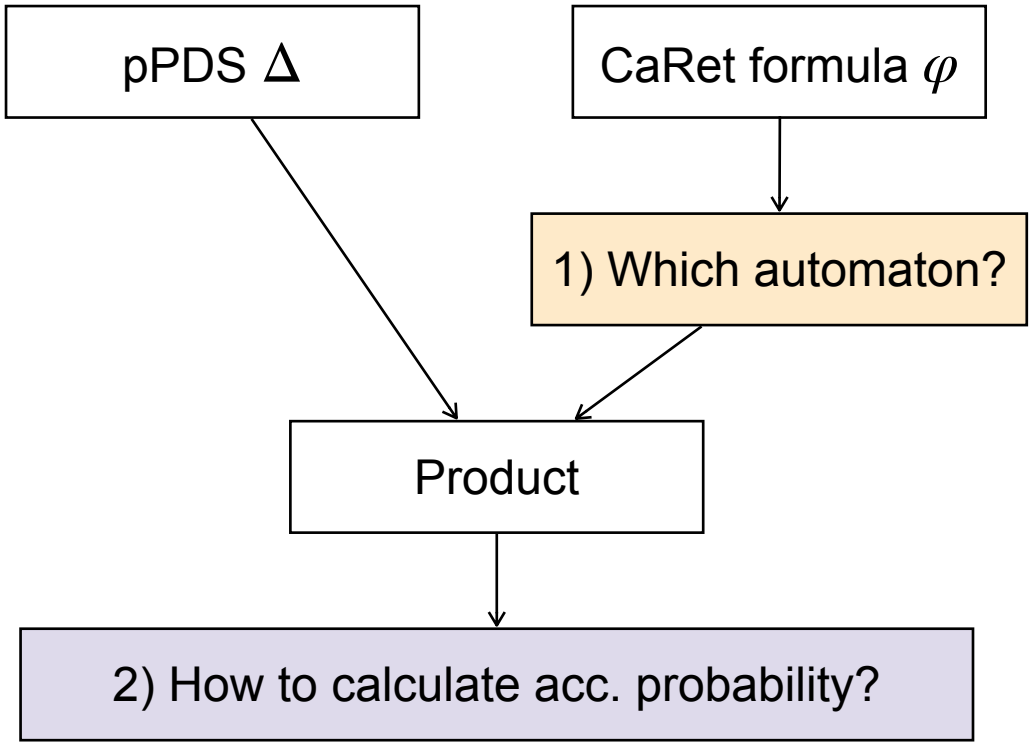
Definition

A pPDS consists of

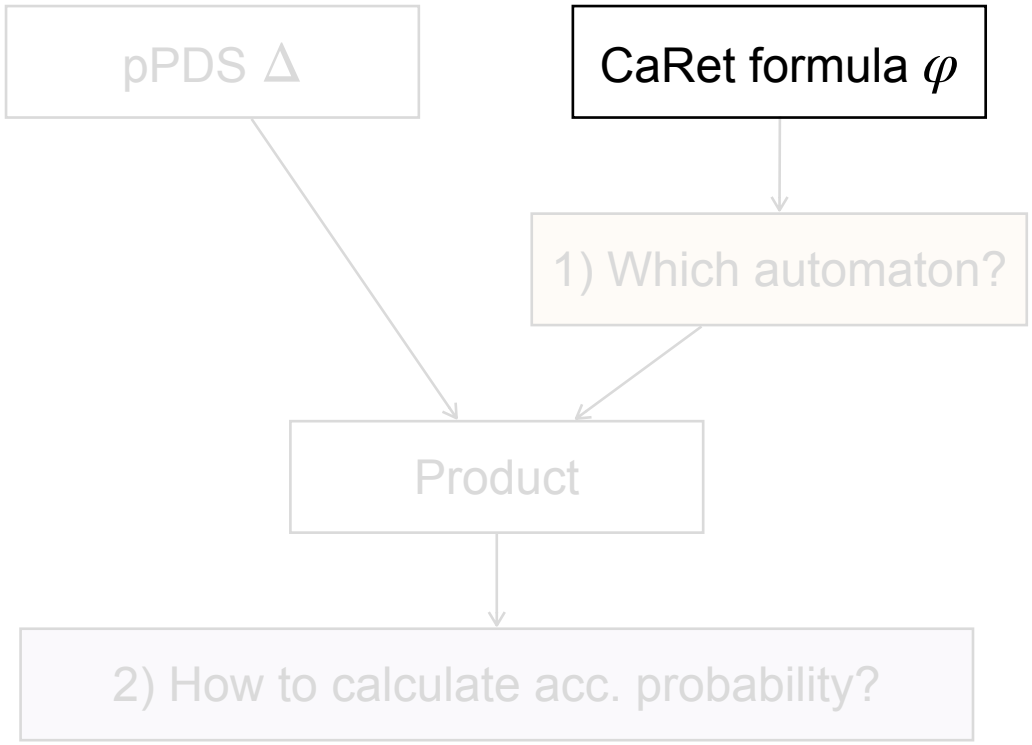
- *a finite set* $Q = (Q_{push}, Q_{int}, Q_{pop})$ *of states*
- *a finite stack alphabet* Γ
- *an initial configuration* $(q_0, Z_0) \in Q \times \Gamma$
- *(probabilistic) transitions*
  - $P_{push}: Q_{push} \to Dist(Q \times \Gamma)$
  - $P_{int}: \quad Q_{int} \to Dist(Q)$
  - $P_{pop}: \quad Q_{pop} \times \Gamma \to Dist(Q)$
- *a consistent labeling function* $Q \to 2^{AP} \times \{push, int, pop\}$



Overview


pPDS Δ
CaRet formula φ
1) Which automaton?
Product
2) How to calculate acc. probability?

Overview


pPDS Δ
CaRet formula φ
1) Which automaton?
Product
2) How to calculate acc. probability?

CaRet, a logic of Calls and Returns [Alur, Etessami, Madhusudan '04]

$$\varphi \;::=\; p \in AP \mid \varphi \vee \varphi \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi\, U\, \varphi \mid \bigcirc^{a}\varphi \mid \varphi\, U^{a}\, \varphi$$

CaRet, a logic of Calls and Returns [Alur, Etessami, Madhusudan '04]

+ special past modalities (omitted)

$$\varphi \;::=\; p \in AP \mid \varphi \vee \varphi \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi\, U\, \varphi \mid \bigcirc^{a}\varphi \mid \varphi\, U^{a}\, \varphi$$

CaRet, a logic of Calls and Returns [Alur, Etessami, Madhusudan '04]

+ special past modalities (omitted)

$$\varphi \;::=\; p \in AP \mid \varphi \vee \varphi \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi\, U\, \varphi \mid \bigcirc^{a}\varphi \mid \varphi\, U^{a}\, \varphi$$

- Interpreted over words $w \in (\ 2^{AP} \times \{\text{push, int, pop}\}\ )^{\omega}$

CaRet, a logic of Calls and Returns [Alur, Etessami, Madhusudan '04]

+ special past modalities (omitted)

$$\varphi \;::=\; p \in AP \mid \varphi \vee \varphi \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi\, U\, \varphi \mid \bigcirc^{a}\varphi \mid \varphi\, U^{a}\, \varphi$$

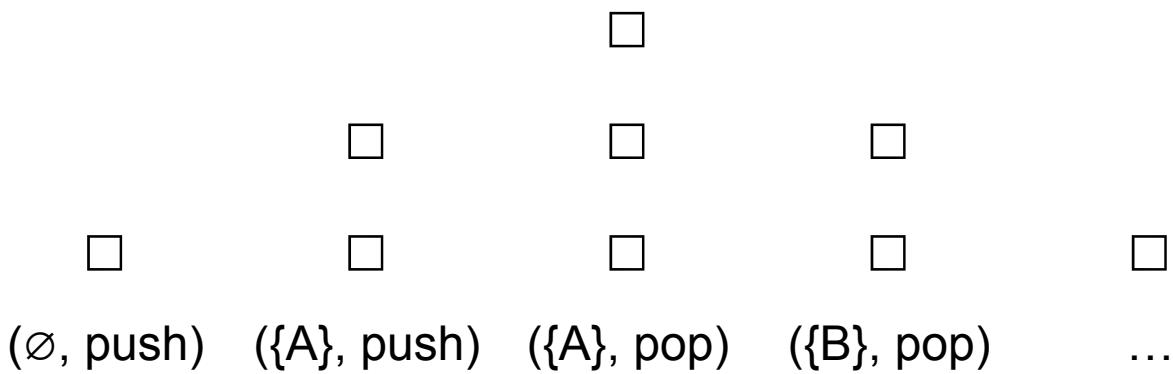
- Interpreted over words $w \in (\ 2^{AP} \times \{\text{push}, \text{int}, \text{pop}\}\ )^{\omega}$
- Example: $w = (\emptyset, \text{push}),\ (\{A\}, \text{push}),\ (\{A\}, \text{pop}),\ (\{B\}, \text{pop}),\ \ldots$

CaRet, a logic of Calls and Returns [Alur, Etessami, Madhusudan '04]

+ special past modalities (omitted)

$$\varphi \;::=\; p \in AP \mid \varphi \vee \varphi \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi\, U\, \varphi \mid \bigcirc^{a}\varphi \mid \varphi\, U^{a}\, \varphi$$

- Interpreted over words $w \in (\ 2^{AP} \times \{\text{push, int, pop}\}\ )^{\omega}$
- Example: $w = (\varnothing, \text{push}),\ (\{A\}, \text{push}),\ (\{A\}, \text{pop}),\ (\{B\}, \text{pop}),\ \ldots$

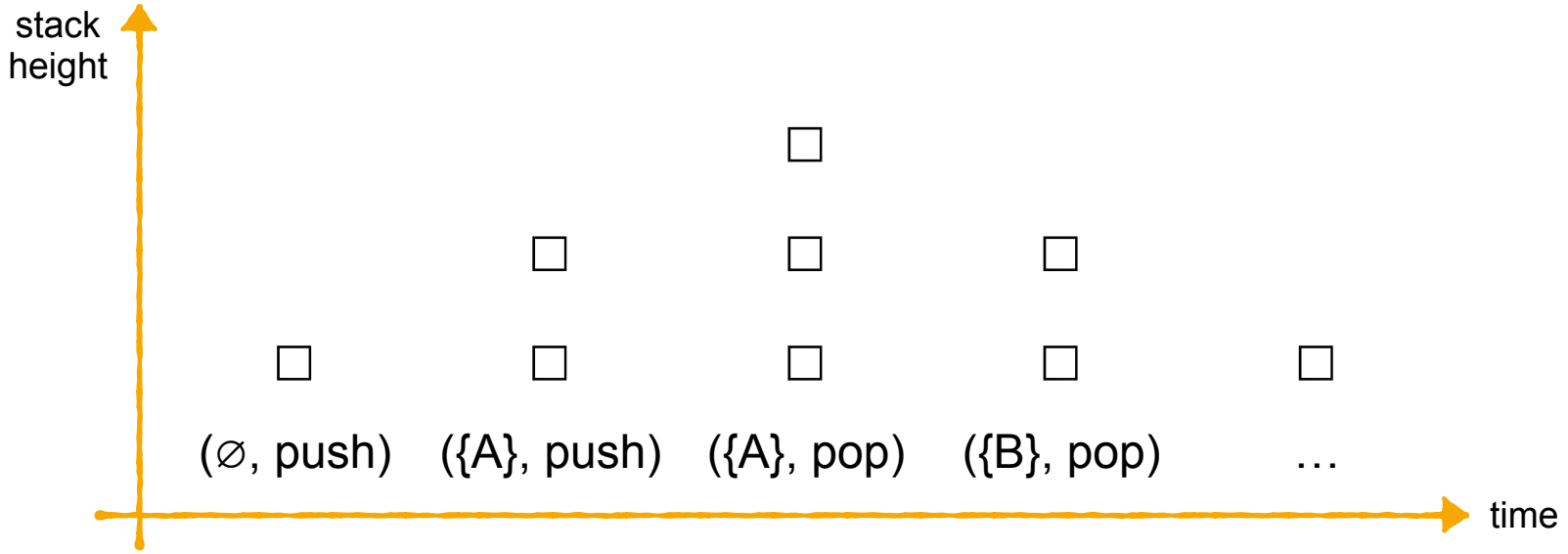


CaRet, a logic of Calls and Returns [Alur, Etessami, Madhusudan '04]

+ special past modalities (omitted)

$$\varphi \;::=\; p \in AP \mid \varphi \vee \varphi \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi\, U\, \varphi \mid \bigcirc^{a}\varphi \mid \varphi\, U^{a}\, \varphi$$

- Interpreted over words $w \in (\ 2^{AP} \times \{\text{push, int, pop}\}\ )^{\omega}$
- Example: $w = (\varnothing, \text{push}),\ (\{A\}, \text{push}),\ (\{A\}, \text{pop}),\ (\{B\}, \text{pop}),\ \ldots$

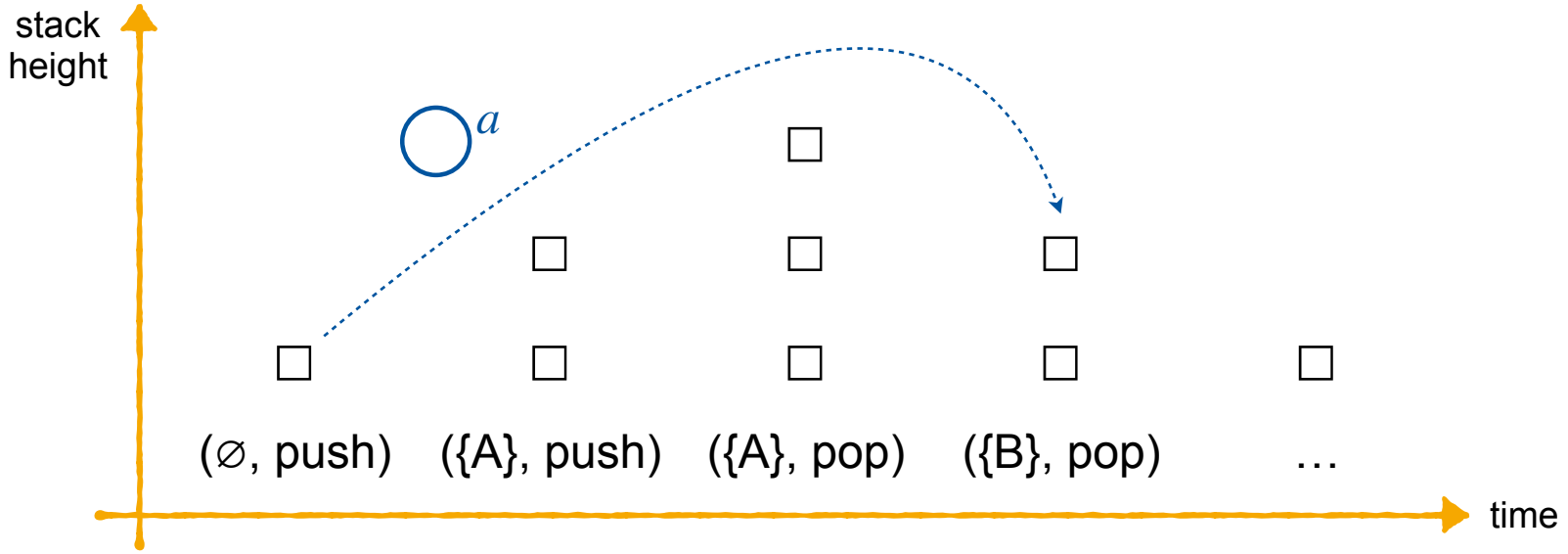


CaRet, a logic of Calls and Returns [Alur, Etessami, Madhusudan '04]

+ special past modalities (omitted)

$$\varphi \;::=\; p \in AP \mid \varphi \vee \varphi \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi\, U\, \varphi \mid \bigcirc^{a}\varphi \mid \varphi\, U^{a}\, \varphi$$

- Interpreted over words $w \in (\ 2^{AP} \times \{\text{push}, \text{int}, \text{pop}\}\ )^{\omega}$
- Example: $w = (\varnothing, \text{push}),\ (\{A\}, \text{push}),\ (\{A\}, \text{pop}),\ (\{B\}, \text{pop}),\ \ldots$



Example CaRet Formulas

[Alur, Etessami, Madhusudan ’04]

Example CaRet Formulas

[Alur, Etessami, Madhusudan '04]

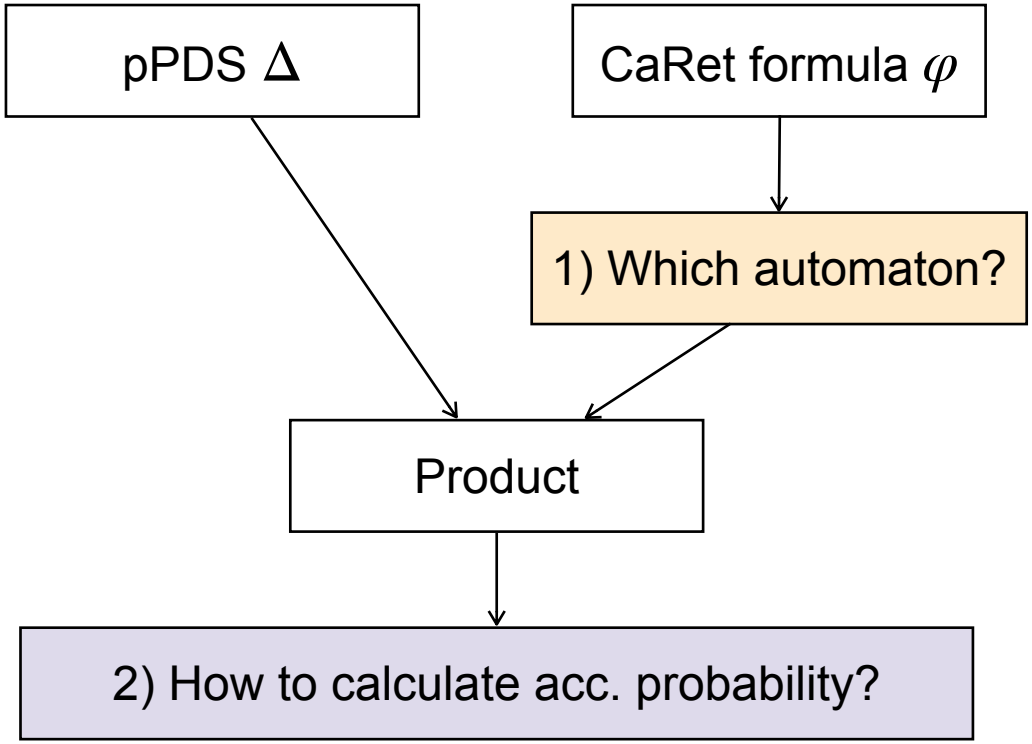
- (Hoare-like) total correctness: $\Box$ (push $\wedge\, P \wedge \varphi \longrightarrow \bigcirc^{a} \psi$)

Example CaRet Formulas

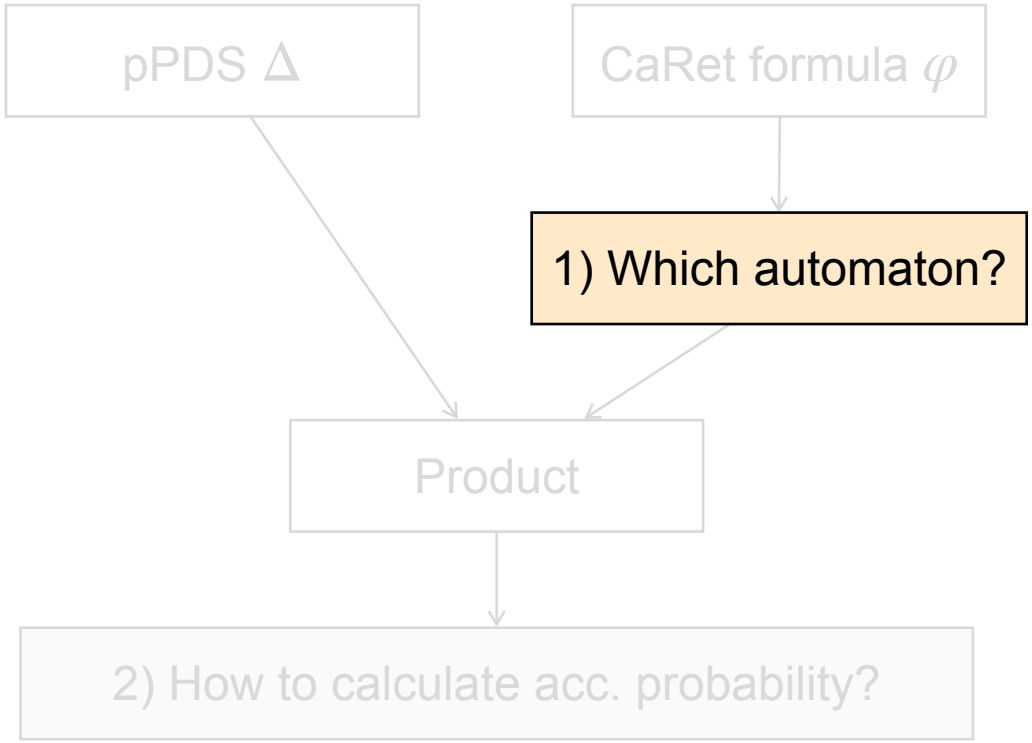
[Alur, Etessami, Madhusudan '04]

- (Hoare-like) total correctness: $\Box(\text{push} \wedge P \wedge \varphi \longrightarrow \bigcirc^{a}\psi)$
- Repeatedly bounded stack heigh: $\Diamond\Box(\text{push} \rightarrow \bigcirc^{a} \text{ pop})$

Overview


pPDS Δ
CaRet formula φ
1) Which automaton?
Product
2) How to calculate acc. probability?

Overview


pPDS Δ
CaRet formula φ
1) Which automaton?
Product
2) How to calculate acc. probability?

1) Automaton Model for Probabilistic CaRet

Requirements

(A) Capture all CaRet-definable languages

(B) Closed under taking products with pPDS

(C) Deterministic, due to the probabilistic setting

Visibly Pushdown Automata (VPA)

VPA = standard pushdown automaton with syntactic restriction:

- Input alphabet $\Sigma \times \{$ push, int, pop $\}$
- Reading a symbol tagged with
  - … push triggers a push transition
  - … int don't change stack height
  - … pop triggers a pop transition

Visibly Pushdown Automata (VPA)

VPA = standard pushdown automaton with syntactic restriction:

- Input alphabet Σ × { push, int, pop }
- Reading a symbol tagged with
  - … push triggers a push transition
  - … int don't change stack height
  - … pop triggers a pop transition

Theorem [Alur & Madhusudan '04]

Every CaRet formula φ can be transformed into a non-deterministic Büchi VPA $\mathcal{A}$ s.t. $L(\varphi) = L(\mathcal{A})$ and $|\mathcal{A}| \in O(2^{|\varphi|})$.

Determinizing VPA

Theorem [Löding, Madhusudan, Serre '04]

Every non-deterministic Büchi VPA $\mathscr{A}$ can be transformed into an equivalent deterministic VPA $\mathscr{D}$ with $|\mathscr{D}| \in O(2^{|\mathscr{A}|^2})$.

$\mathscr{D}$ uses a stair-parity acceptance condition.

Pushdown Automata with Stair-Parity Acceptance

Pushdown Automata with Stair-Parity Acceptance

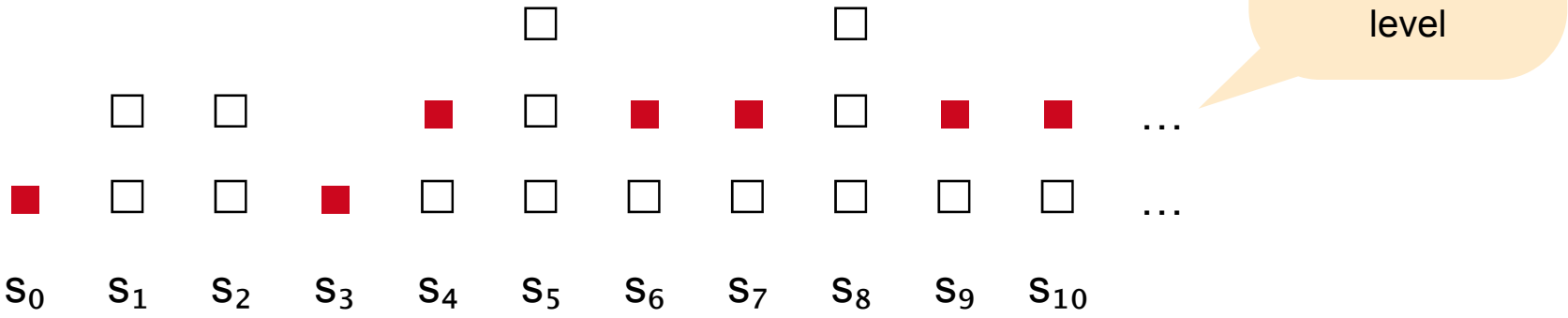
- Priority function $\Omega : S \to \mathbb{N}$ on states S of the VPA (like standard parity)

Pushdown Automata with Stair-Parity Acceptance

- Priority function $\Omega : S \to \mathbb{N}$ on states S of the VPA (like standard parity)
- Consider the **"steps"** of a run of the automaton:

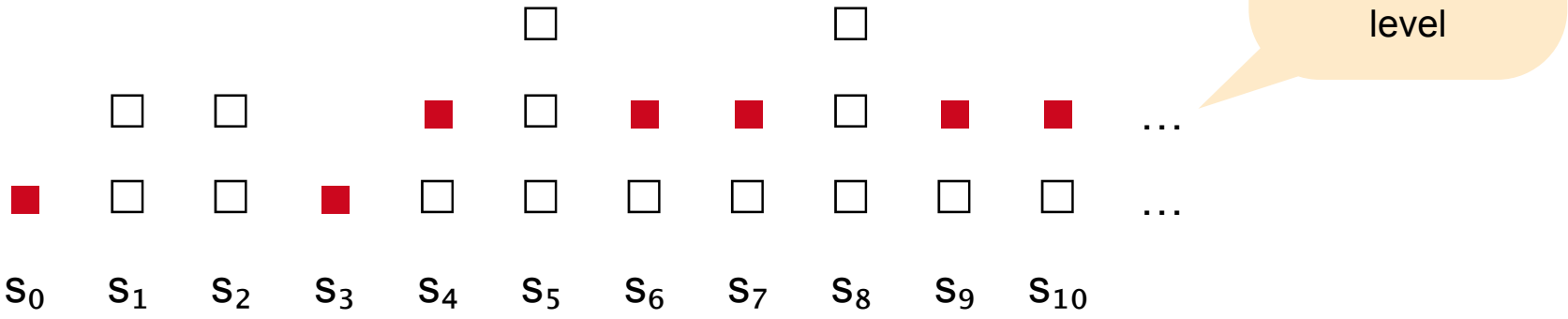
Pushdown Automata with Stair-Parity Acceptance

- Priority function $\Omega : S \to \mathbb{N}$ on states S of the VPA (like standard parity)
- Consider the **"steps"** of a run of the automaton:



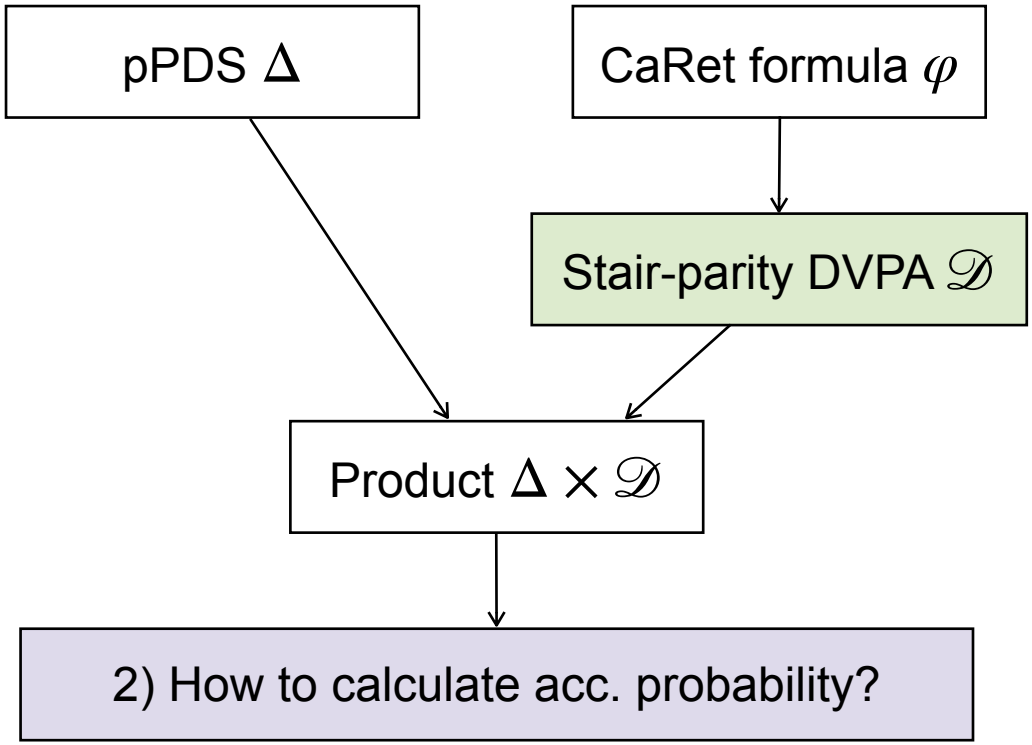
Pushdown Automata with Stair-Parity Acceptance

- Priority function $\Omega : S \to \mathbb{N}$ on states S of the VPA (like standard parity)
- Consider the **"steps"** of a run of the automaton:

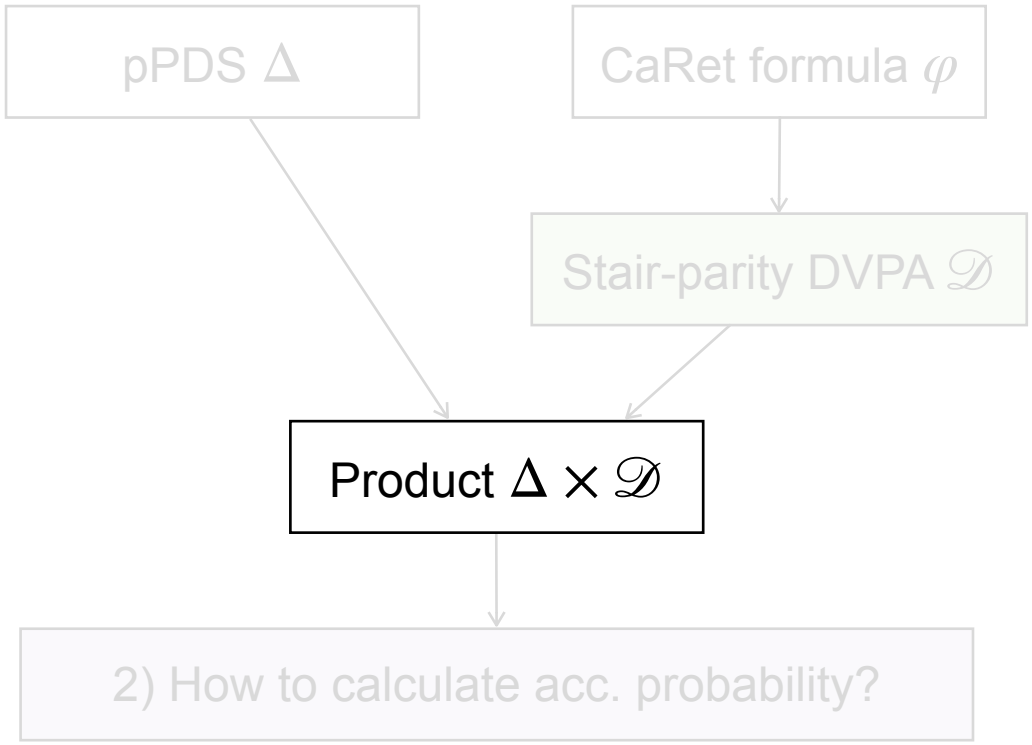


- Stair-parity = standard parity evaluated on sequence of steps
  Run is accepted iff the minimum priority seen inf. often at **steps** is even

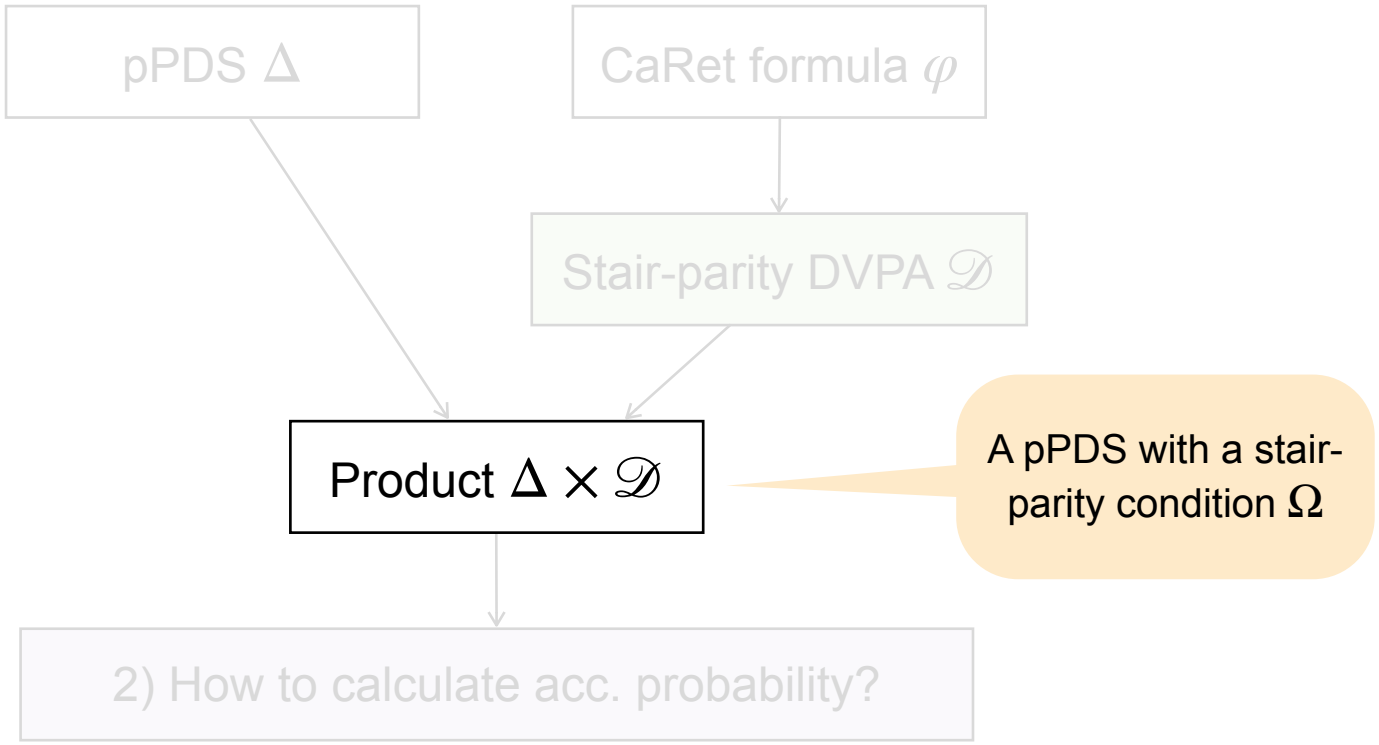
Overview


pPDS Δ
CaRet formula φ
Stair-parity DVPA $\mathscr{D}$
Product $\Delta \times \mathscr{D}$
2) How to calculate acc. probability?

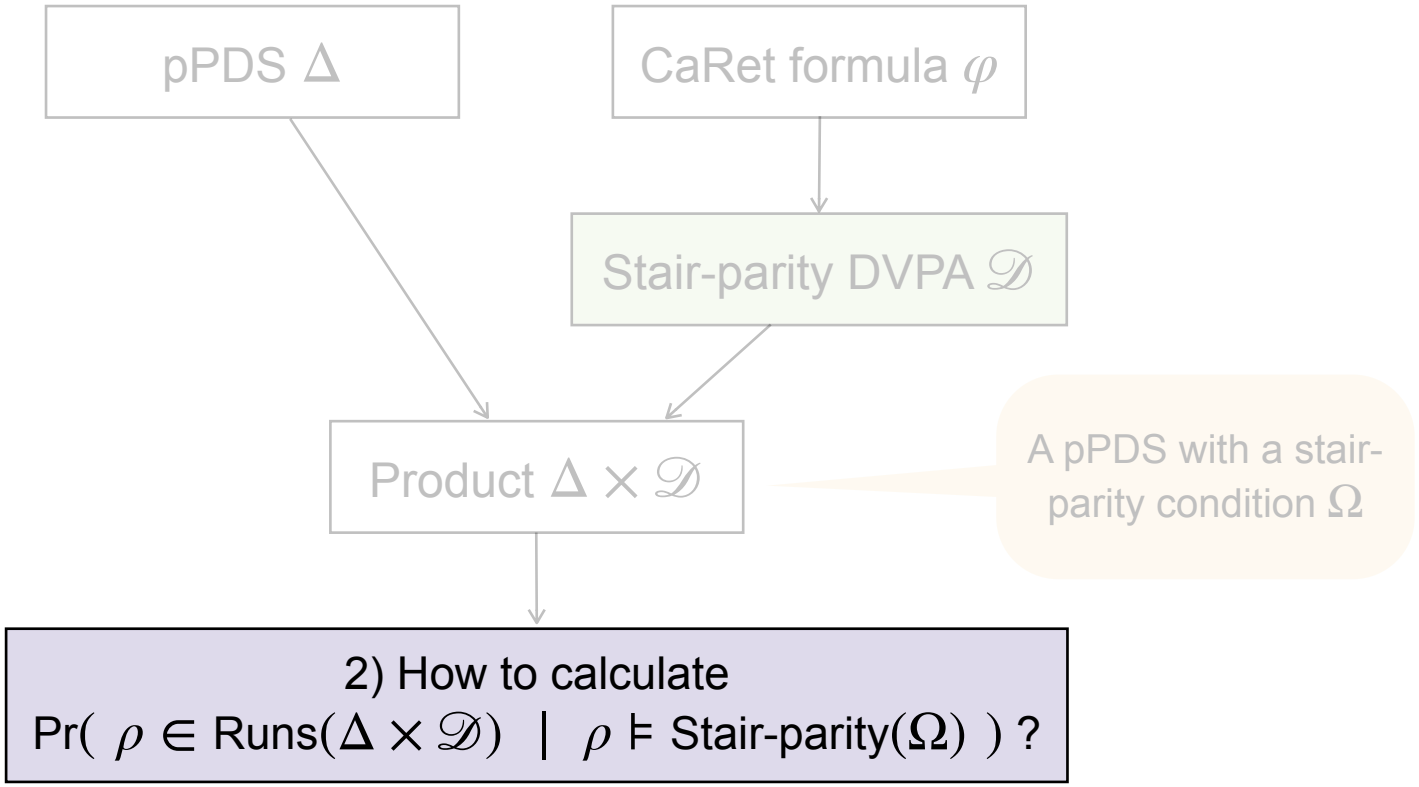
Overview


pPDS Δ
CaRet formula φ
Stair-parity DVPA $\mathscr{D}$
Product $\Delta \times \mathscr{D}$
2) How to calculate acc. probability?

Overview


pPDS Δ
CaRet formula φ
Stair-parity DVPA $\mathscr{D}$
Product $\Delta \times \mathscr{D}$
A pPDS with a stair-parity condition Ω
2) How to calculate acc. probability?

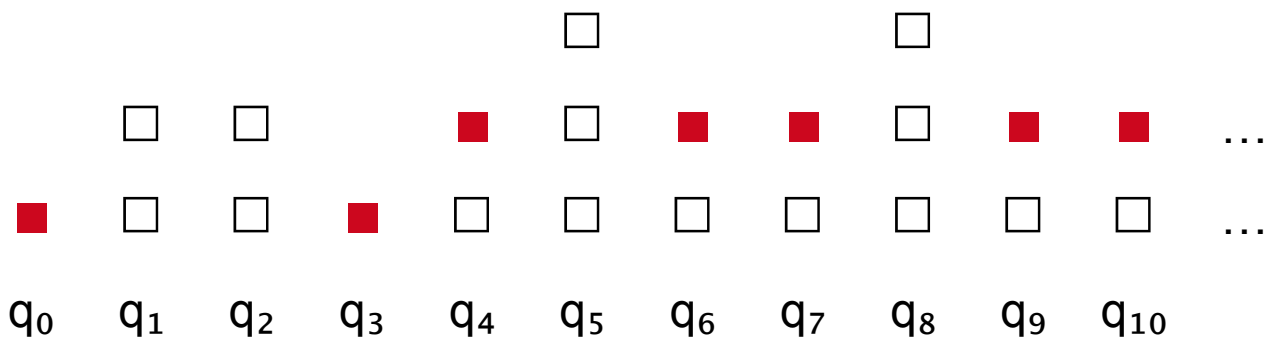
Overview


pPDS Δ
CaRet formula φ
Stair-parity DVPA $\mathscr{D}$
Product $\Delta \times \mathscr{D}$
A pPDS with a stair-parity condition Ω
2) How to calculate
Pr($\rho \in$ Runs($\Delta \times \mathscr{D}$) | $\rho \vDash$ Stair-parity(Ω)) ?

2) Stair-Parity Acceptance Probability in pPDS

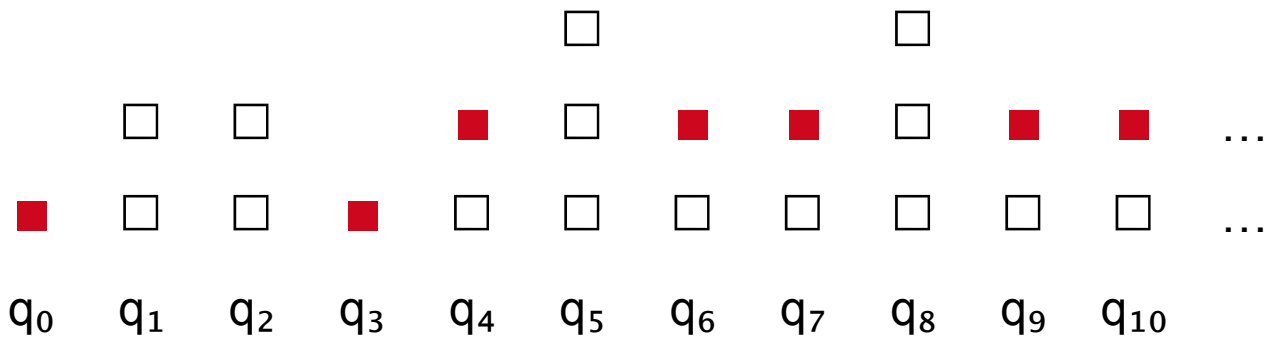
2) Stair-Parity Acceptance Probability in pPDS

- pPDS generate random runs such as this:



2) Stair-Parity Acceptance Probability in pPDS

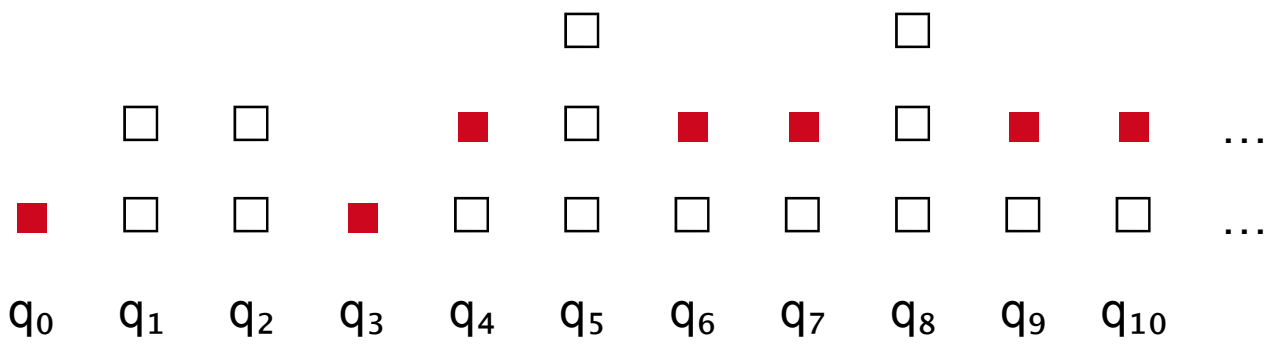
- pPDS generate random runs such as this:



- Idea [Esparza el al.]: Consider the i-th step $S^{(i)}$ as a random variable

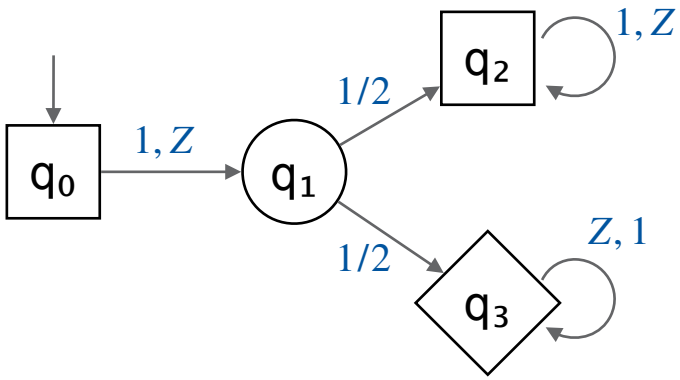
2) Stair-Parity Acceptance Probability in pPDS

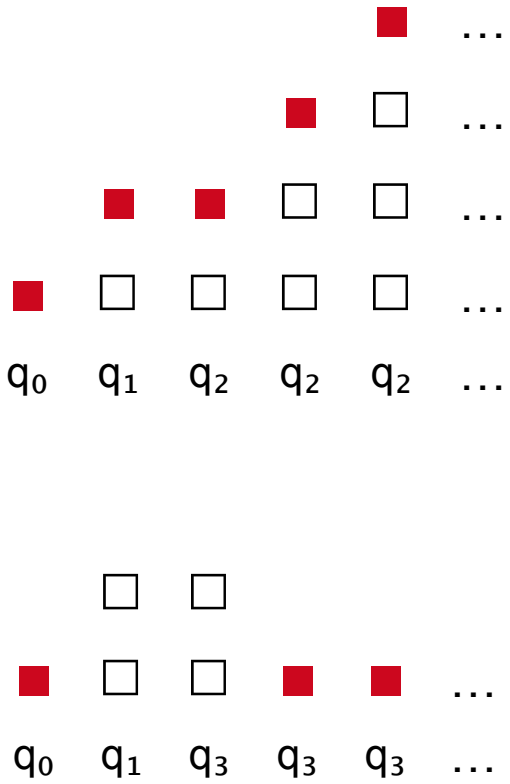
- pPDS generate random runs such as this:



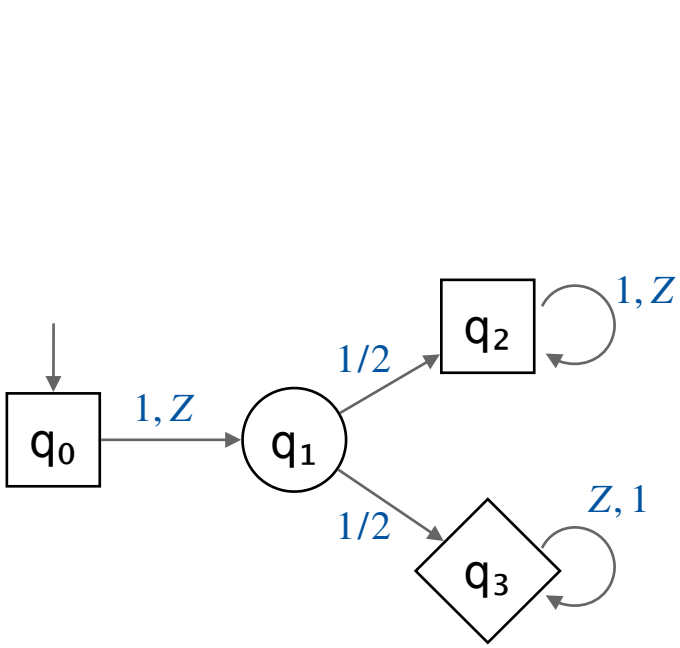
- Idea [Esparza el al.]: Consider the i-th step $S^{(i)}$ as a random variable
- Stochastic process $S^{(1)}, S^{(2)}, \ldots$ is a finite Markov chain

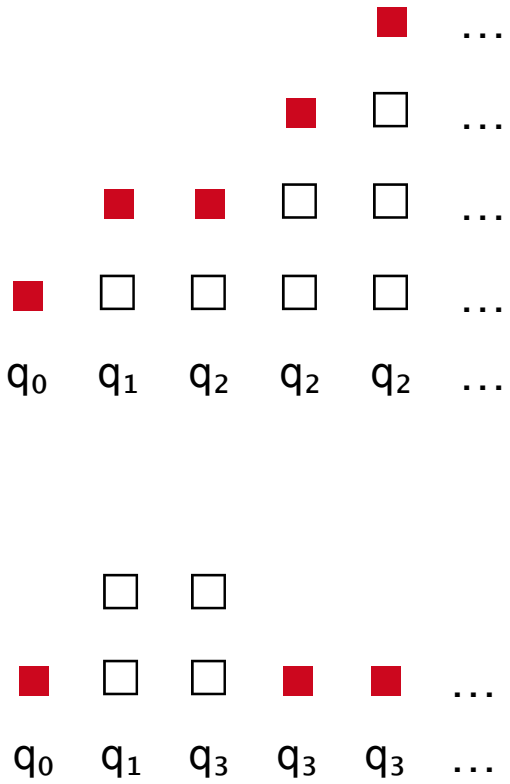
The Step Markov Chain

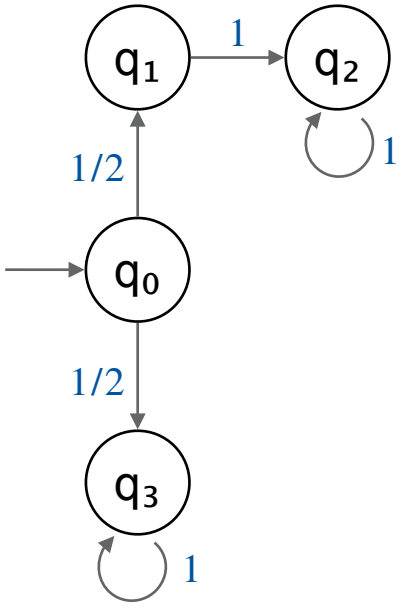




The Step Markov Chain





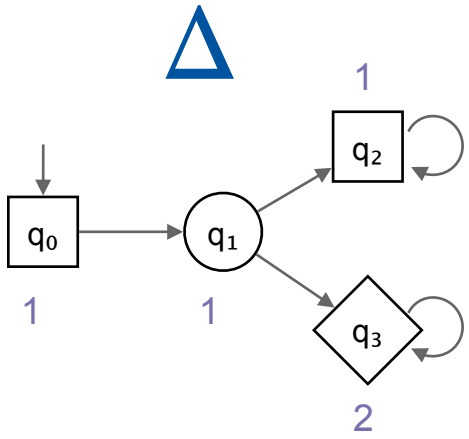


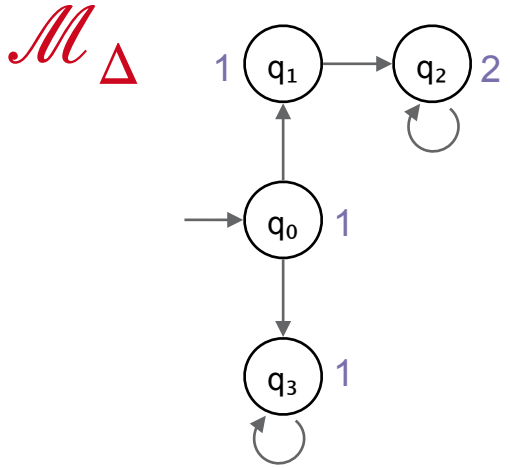
Killing two Birds with one Stone

Theorem

Let Δ be a PPDS and $\mathcal{M}_\Delta$ its step Markov chain (both with state space Q).
Let $\Omega : Q \to \mathbb{N}$ be a priority function. Then:

$$Pr(\ \rho \in Runs(\Delta) \mid \rho \models \textit{Stair-parity}(\Omega)\ ) \ = \ Pr(\ \rho \in Runs(\mathcal{M}_\Delta) \mid \rho \models \textit{Parity}(\Omega)\ )$$



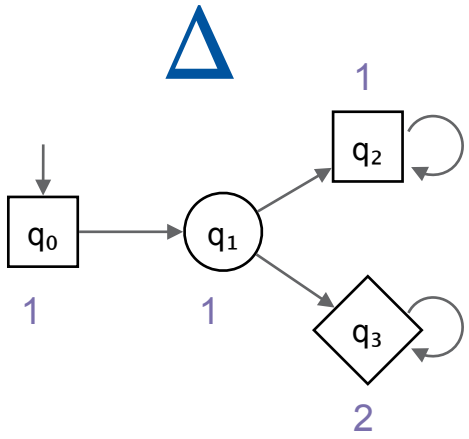


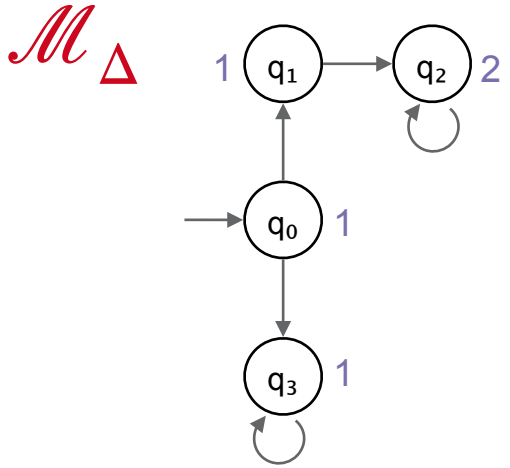
Killing two Birds with one Stone

Theorem

Let Δ be a PPDS and $\mathcal{M}_\Delta$ its step Markov chain (both with state space Q).
Let $\Omega : Q \to \mathbb{N}$ be a priority function. Then:

$$Pr(\ \rho \in Runs(\Delta) \mid \rho \models \textit{Stair-parity}(\Omega)\ ) \ = \ Pr(\ \rho \in Runs(\mathcal{M}_\Delta) \mid \rho \models \textit{Parity}(\Omega)\ )$$





stair-parity ---------- reduces to ----------▸ standard parity

Main Result: Decidability of Probabilistic CaRet Model Checking

Theorem

Let Δ *be a PPDS,* φ *a CaRet formula, and* $\lambda \in [0,1]$*. The problem*

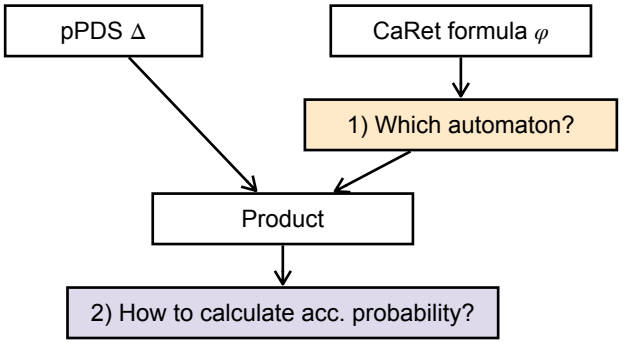
$$Pr(\ \rho \in Runs(\Delta) \mid \rho \vDash \varphi\ )\ \geq_?\ \lambda$$

is decidable in $PSPACE$ *in* $|\Delta|$ *and* $2EXPSPACE$ *in* $|\varphi|$*.*

Summary

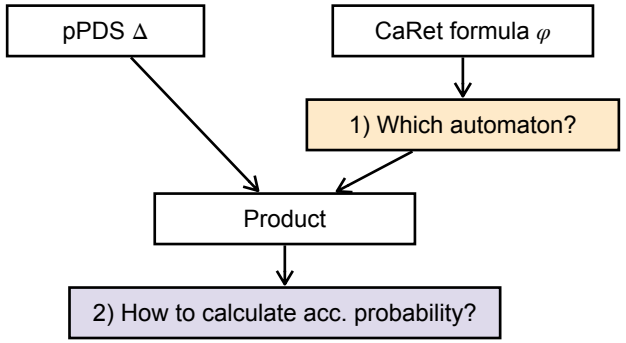
- Model checking finite-state probabilistic procedural & possibly recursive programs

- Automata-based CaRet Model Checking

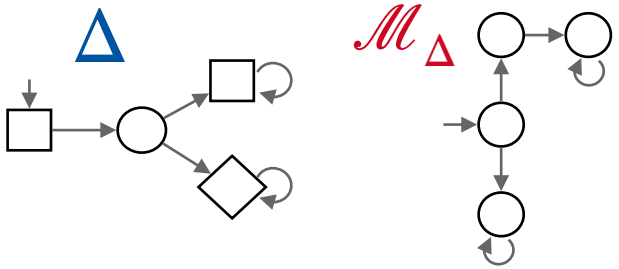


Summary

- Model checking finite-state probabilistic procedural & possibly recursive programs
- Automata-based CaRet Model Checking

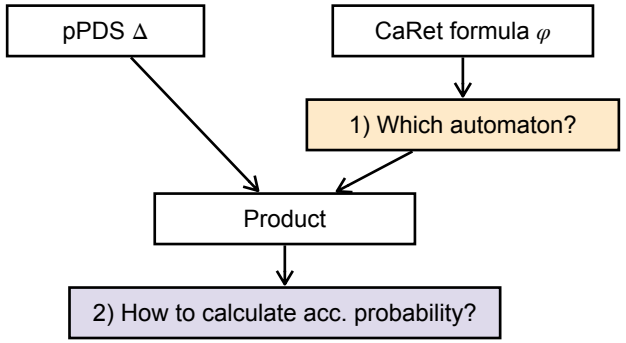


- 1) Deterministic stair-parity VPA are a suitable automaton model
- 2) Reduce stair-parity acceptance in pPDS to parity acceptance in step Markov chain

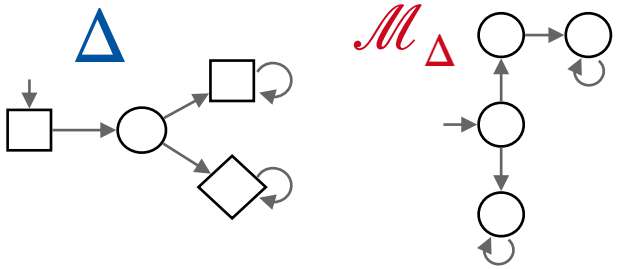


Summary

- Model checking finite-state probabilistic procedural & possibly recursive programs
- Automata-based CaRet Model Checking



- 1) Deterministic stair-parity VPA are a suitable automaton model
- 2) Reduce stair-parity acceptance in pPDS to parity acceptance in step Markov chain



Thank you!

Backup: Computing the Step Markov Chain

	$q \to r$	$q\bot \to r$	$q\bot \to r\bot$	$q \to r\bot$
$q \in Q_{\mathsf{call}}$	$\frac{[r\uparrow]}{[q\uparrow]}\Big(\sum_{r',Z} P_{\mathsf{call}}(q,r'Z)[r'Z{\downarrow}r] + \sum_{Z} P_{\mathsf{call}}(q,rZ)\Big)$	$\sum_{Z} P_{\mathsf{call}}(q,rZ)[r\uparrow]$	$\sum_{r',Z} P_{\mathsf{call}}(q,r'Z)[r'Z{\downarrow}r]$	0
$q \in Q_{\mathsf{int}}$	$\frac{[r\uparrow]}{[q\uparrow]} P_{\mathsf{int}}(q,r)$	0	$P_{\mathsf{int}}(q,r)$	0
$q \in Q_{\mathsf{ret}}$	n/a	0	$P_{\mathsf{ret}}(q\bot,r)$	n/a