Uniform random generation of executions in concurrent programs

Martin Pépin
under the supervision of Antoine Genitrini & Frédéric Peschanski

July 5, 2019

Sorbonne University – LIP6 – Paris

Outline

The project

Long term research project:
O. Bodini, M. Dien, A. Genitrini, F. Peschanski,...

Operators of concurrency $\rightarrow$ combinatorial **interpretation**

The project

Long term research project:
O. Bodini, M. Dien, A. Genitrini, F. Peschanski,...

Operators of concurrency $\rightarrow$ combinatorial **interpretation**

- Quantitative study
  $\rightarrow$ Combinatorial explosion
  $\rightarrow$ **Average** number of executions?

The project

Long term research project:
O. Bodini, M. Dien, A. Genitrini, F. Peschanski,...

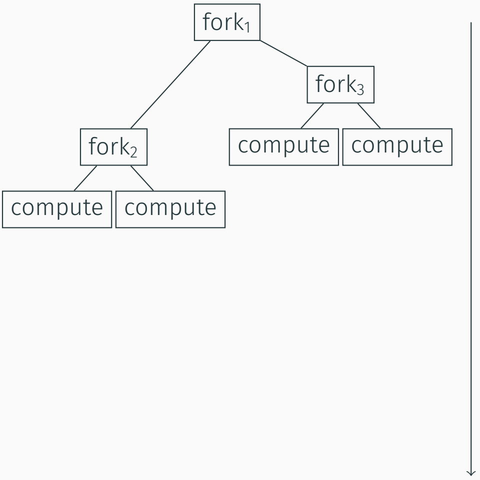
Operators of concurrency $\rightarrow$ combinatorial **interpretation**

- Quantitative study
  $\rightarrow$ Combinatorial explosion
  $\rightarrow$ **Average** number of executions?
- Algorithmic applications
  $\rightarrow$ Counting executions
  $\rightarrow$ **Uniform** random sampling of executions

Classical operators of concurrency (1)

$x.P \qquad P \parallel Q$

→ Trees [AofA'12]



Classical operators of concurrency (1)

$x.P$ $\qquad$ $P \parallel Q$

$\rightarrow$ Trees [AofA'12]

Synchronization

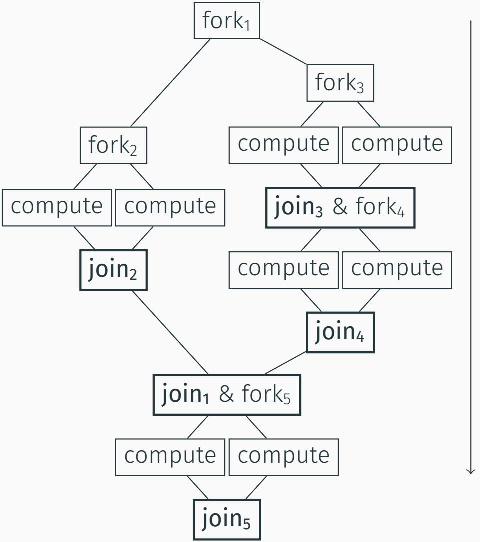
$\rightarrow$ DAGs

Counting is ♯P-complete

[Brightwell & Winkler'91]

(too difficult)

$\longrightarrow$ Fork-Join graphs [Analco'17]



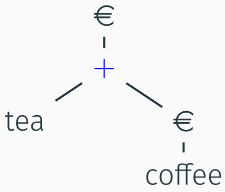
Classical operators of concurrency (2)

$+$ (choice operator)

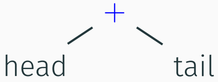
→ In the tree model [FSTTCS'13]

- Toy example
- Technical difficulties

Beverage vending machine



Tossing a coin



In today's talk

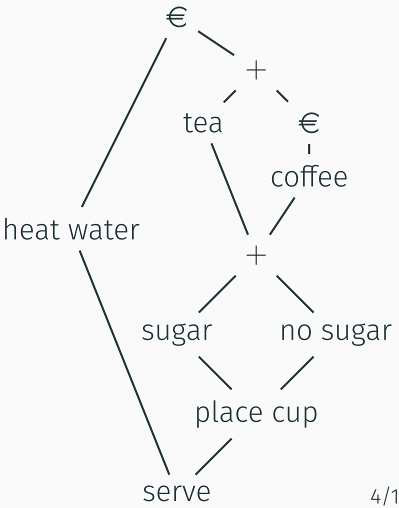
$+$. $\|$ sync

$\rightarrow$ General model: petri nets

$\rightarrow$ Tractable subclass: SP_{+}

$\rightarrow$ Uniform random generator of executions

Beverage vending machine



Outline

The project

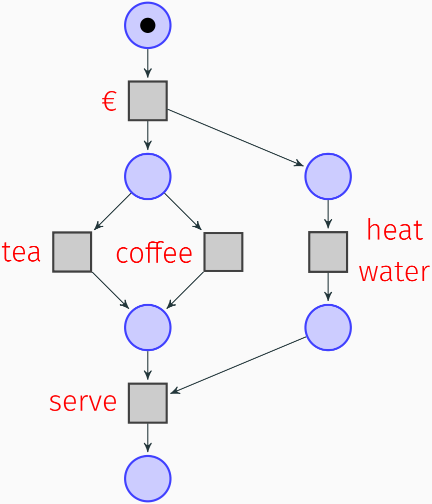
Petri nets

Combinatorial interpretation

Uniform sampling of executions

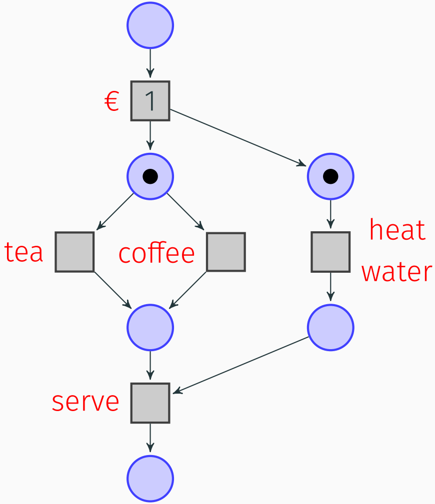
Petri nets

- Model for concurrent systems
- Places  symbolise states / resources
- Transitions  symbolise actions
- Connected by directed arcs
- Transitions consume and produce tokens



Petri nets

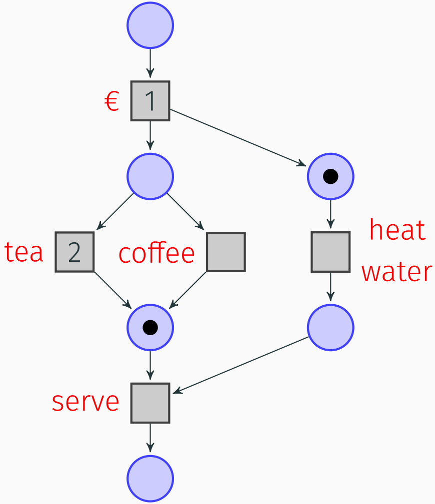
- Model for concurrent systems
- Places symbolise states / resources
- Transitions symbolise actions
- Connected by directed arcs
- Transitions consume and produce tokens



€;

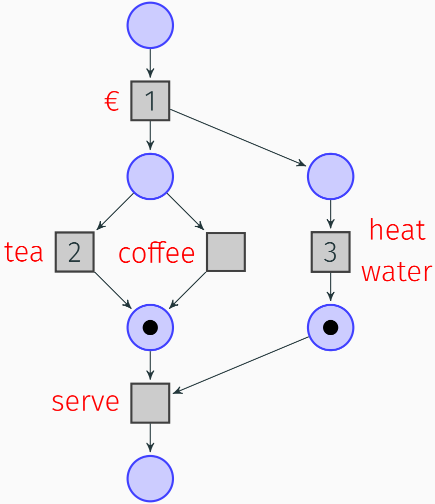
Petri nets

- Model for concurrent systems
- Places symbolise states / resources
- Transitions symbolise actions
- Connected by directed arcs
- Transitions consume and produce tokens



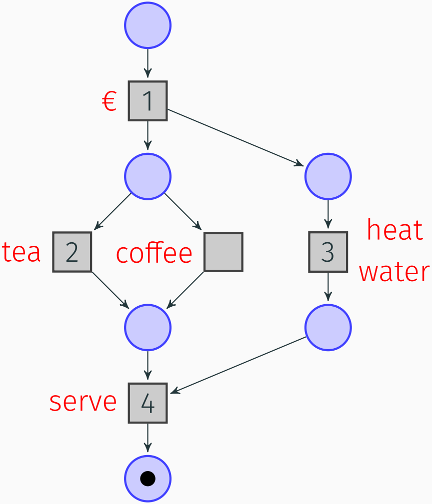
€; tea;

- Model for concurrent systems
- Places symbolise states / resources
- Transitions symbolise actions
- Connected by directed arcs
- Transitions consume and produce tokens



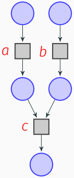
€; tea; heat;

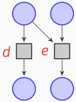
- Model for concurrent systems
- Places  symbolise states / resources
- Transitions  symbolise actions
- Connected by directed arcs
- Transitions consume and produce tokens



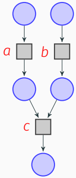
€; tea; heat; serve

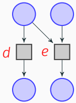
Construction rules (I)

$N_1 =$ 

$N_2 =$ 

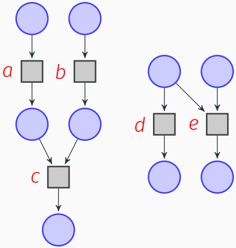
Construction rules (I)

$N_1 =$ 

$N_2 =$ 

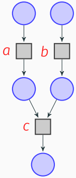
Parallel composition

$(N_1 \parallel N_2)$

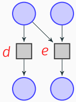


Construction rules (I)

$N_1 =$

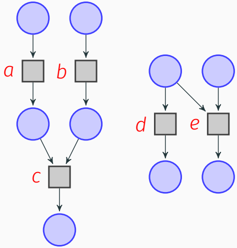


$N_2 =$



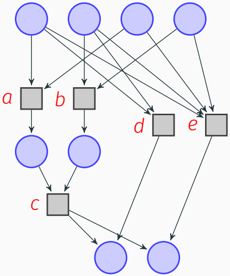
Parallel composition

$(N_1 \parallel N_2)$

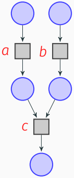


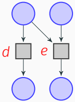
Choice

$(N_1 + N_2)$



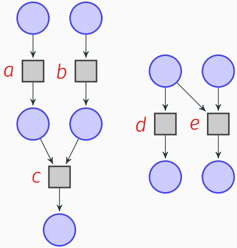
Construction rules (I)

$N_1 =$ 

$N_2 =$ 

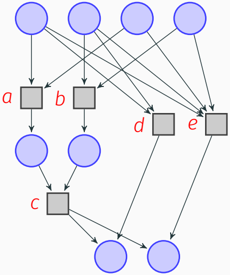
Parallel composition

$(N_1 \parallel N_2)$



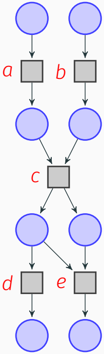
Choice

$(N_1 + N_2)$



Sequence

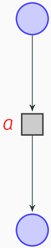
$(N_1 \,;\, N_2)$



Construction rules (II)

Atomic action

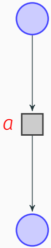
(a)



Construction rules (II)

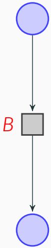
Atomic action

(a)



Barrier

(B)



Synchronisation

($[N : B]$)

Merge all the transitions labelled with B in N

Properties

We know

- $;$ is associative
- $\parallel$ is associative and commutative
- $+$ is associative and commutative
- Counting the executions is #P-hard

Properties

We know

- $;$ is associative
- $\parallel$ is associative and commutative
- $+$ is associative and commutative
- Counting the executions is #P-hard

We wish to prove

- Petri nets constructed using a, B, $;$, $+$, $\parallel$ and $[\cdot : B]$ are one-safe?
- All cycles are deadlocks?
- We can construct any cycle-free petri net that is one-safe???

Summary

$$
\begin{aligned}
N, M \;\; ::= \;& N \parallel M \\
& N + M \\
& N \;;\; M \\
& a \\
& B \\
& [N : B]
\end{aligned}
$$

Tractable subclass: SP_+

Summary

$$
\begin{aligned}
N, M \ ::= \ & N \parallel M \\
& N + M \\
& N \ ; \ M \\
& a \\
& B \\
& [N : B]
\end{aligned}
$$

$\rightsquigarrow$

Non-deterministic series parallel programs (SP_+)

- Simpler model
- Still expressive
- Tractable:
  - Specifiable
  - Efficient Uniform random generation of executions is possible

Outline

Analytic combinatorics

Definition: combinatorial class

A set $\mathcal{C}$ equipped with a size function $|\cdot| : \mathcal{C} \to \mathbb{N}$ such that $\forall n, \#\{c \in \mathcal{C}; |c| = n\} < \infty$

Specification $\quad\rightarrow\quad$ Generating Function $\quad\rightarrow\quad$ Asymptotics

Definition: combinatorial class

A set $\mathcal{C}$ equipped with a size function $|\cdot| : \mathcal{C} \to \mathbb{N}$ such that $\forall n, \#\{c \in \mathcal{C}; |c| = n\} < \infty$

Specification $\rightarrow$ Generating Function $\rightarrow$ Asymptotics

Spec	$\rightarrow$	GF
$\mathcal{A}$	$\rightarrow$	$\sum_n a_n z^n$
$\mathcal{A} + \mathcal{B}$	$\rightarrow$	$A(z) + B(z)$
$\mathcal{A} \times \mathcal{B}$	$\rightarrow$	$A(z)B(z)$
	$\dots$	

Analytic combinatorics

Definition: combinatorial class

A set $\mathcal{C}$ equipped with a size function $|\cdot| : \mathcal{C} \to \mathbb{N}$ such that $\forall n, \#\{c \in \mathcal{C}; |c| = n\} < \infty$

Specification $\rightarrow$ Generating Function $\rightarrow$ Asymptotics

Spec	$\rightarrow$	GF
$\mathcal{A}$	$\rightarrow$	$\sum_n a_n z^n$
$\mathcal{A}+\mathcal{B}$	$\rightarrow$	$A(z)+B(z)$
$\mathcal{A}\times\mathcal{B}$	$\rightarrow$	$A(z)B(z)$
	$\dots$	

$$A(z) \approx \tau - C\sqrt{1-\frac{z}{\rho}}$$

$$\downarrow$$

$$a_n \sim C' n^{-\frac{3}{2}} \rho^{-n}$$

SP$_+$ specification

Informal

$$
\begin{aligned}
\mathcal{S} &= \mathcal{S}_; \quad | \quad \mathcal{S}_\| \quad | \quad \mathcal{S}_+ \quad | \quad a \\
\mathcal{S}_; &= (\mathcal{S} \setminus \mathcal{S}_;) \;;\; (\mathcal{S} \setminus \mathcal{S}_;) \;;\; \cdots \;;\; (\mathcal{S} \setminus \mathcal{S}_;) \quad (\geq 2 \text{ terms}) \\
\mathcal{S}_\| &= (\mathcal{S} \setminus \mathcal{S}_\|) \;\|\; (\mathcal{S} \setminus \mathcal{S}_\|) \;\|\; \cdots \;\|\; (\mathcal{S} \setminus \mathcal{S}_\|) \quad (\geq 2 \text{ terms}) \\
\mathcal{S}_+ &= (\mathcal{S} \setminus \mathcal{S}_+) \;+\; (\mathcal{S} \setminus \mathcal{S}_+) \;+\; \cdots \;+\; (\mathcal{S} \setminus \mathcal{S}_+) \quad (\geq 2 \text{ terms})
\end{aligned}
$$

SP$_+$ specification

Informal

$$\mathcal{S} = \mathcal{S}_{;} \quad | \quad \mathcal{S}_{\|} \quad | \quad \mathcal{S}_{+} \quad | \quad a$$

$$\mathcal{S}_{;} = (\mathcal{S} \setminus \mathcal{S}_{;}) \ ; \ (\mathcal{S} \setminus \mathcal{S}_{;}) \ ; \ \cdots \ ; \ (\mathcal{S} \setminus \mathcal{S}_{;}) \quad (\geq 2 \text{ terms})$$

$$\mathcal{S}_{\|} = (\mathcal{S} \setminus \mathcal{S}_{\|}) \ \| \ (\mathcal{S} \setminus \mathcal{S}_{\|}) \ \| \ \cdots \ \| \ (\mathcal{S} \setminus \mathcal{S}_{\|}) \quad (\geq 2 \text{ terms})$$

$$\mathcal{S}_{+} = (\mathcal{S} \setminus \mathcal{S}_{+}) \ + \ (\mathcal{S} \setminus \mathcal{S}_{+}) \ + \ \cdots \ + \ (\mathcal{S} \setminus \mathcal{S}_{+}) \quad (\geq 2 \text{ terms})$$

Formal

$$\mathcal{S} = \mathcal{Z} + \mathcal{S}_{;} + \mathcal{S}_{\|} + \mathcal{S}_{+}$$

$$\mathcal{S}_{;} = \textsc{Seq}_{\geq 2}(\mathcal{S} \setminus \mathcal{S}_{;})$$

$$\mathcal{S}_{\|} = \textsc{MSet}_{\geq 2}(\mathcal{S} \setminus \mathcal{S}_{\|})$$

$$\mathcal{S}_{+} = \textsc{MSet}_{\geq 2}(\mathcal{S} \setminus \mathcal{S}_{+})$$

SP₊ specification

Informal

$$\mathcal{S} = \mathcal{S}_; \quad | \quad \mathcal{S}_\| \quad | \quad \mathcal{S}_+ \quad | \quad a$$

$$\mathcal{S}_; = (\mathcal{S} \setminus \mathcal{S}_;) \;;\; (\mathcal{S} \setminus \mathcal{S}_;) \;;\; \cdots \;;\; (\mathcal{S} \setminus \mathcal{S}_;) \quad (\geq 2 \text{ terms})$$

$$\mathcal{S}_\| = (\mathcal{S} \setminus \mathcal{S}_\|) \;\|\; (\mathcal{S} \setminus \mathcal{S}_\|) \;\|\; \cdots \;\|\; (\mathcal{S} \setminus \mathcal{S}_\|) \quad (\geq 2 \text{ terms})$$

$$\mathcal{S}_+ = (\mathcal{S} \setminus \mathcal{S}_+) \;+\; (\mathcal{S} \setminus \mathcal{S}_+) \;+\; \cdots \;+\; (\mathcal{S} \setminus \mathcal{S}_+) \quad (\geq 2 \text{ terms})$$

Formal

$$\mathcal{S} = \mathcal{Z} + \mathcal{S}_; + \mathcal{S}_\| + \mathcal{S}_+$$

$$\mathcal{S}_; = \textsc{Seq}_{\geq 2}(\mathcal{S} \setminus \mathcal{S}_;)$$

$$\mathcal{S}_\| = \textsc{MSet}_{\geq 2}(\mathcal{S} \setminus \mathcal{S}_\|)$$

$$\mathcal{S}_+ = \textsc{MSet}_{\geq 2}(\mathcal{S} \setminus \mathcal{S}_+)$$

$\xrightarrow{\text{symbolic method}}$ system of equations...

The symbolic method for executions

We can do the same for counting the possible executions

Specification	$\to$	Exponential generating function
$\mathcal{A}$	$\to$	$A(z) = \sum_n a_n \frac{z^n}{n!}$
$\mathcal{A} + \mathcal{B}$	$\to$	$A(z) + B(z)$
$\mathcal{A} \star \mathcal{B}$	$\to$	$A(z)B(z)$
$\mathcal{A} \boxed{\star} \mathcal{B}$ [1]	$\to$	$\int_0^z A'(z-u)B(u)du + A(0)B(z)$

[1] Ordered product [Analco'17]

Expected results

($\star$) Number of programs
Should be of the form $C \cdot n^{-\frac{3}{2}} \rho^{-n}$

Expected results

(⋆) Number of programs
Should be of the form $C \cdot n^{-\frac{3}{2}} \rho^{-n}$

(⋆) Average number of **global choices**
Should be of the form $A \cdot B^n$ for small B.
(Numerically, in the non commutative case: $B \approx 1.11678$)

Expected results

($\star$) Number of programs
Should be of the form $C \cdot n^{-\frac{3}{2}} \rho^{-n}$

($\star$) Average number of **global choices**
Should be of the form $A \cdot B^n$ for small B.
(Numerically, in the non commutative case: $B \approx 1.11678$)

($\star\star\star$) Average number of executions
(very ugly equations)

Outline

Goal

Sampling executions **uniformly at random** in a given SP_+ program.

Goal

Sampling executions **uniformly at random** in a given SP_+ program.

Idea 1: listing of all possible executions? **No**

Goal

Sampling executions **uniformly at random** in a given SP_{+} program.

Idea 1: listing of all possible executions? **No**

Idea 2: expand all the global choices and choose one? **No**
$\hookrightarrow$ [CSR'17] gives an algorithm for the choice-free case

Goal

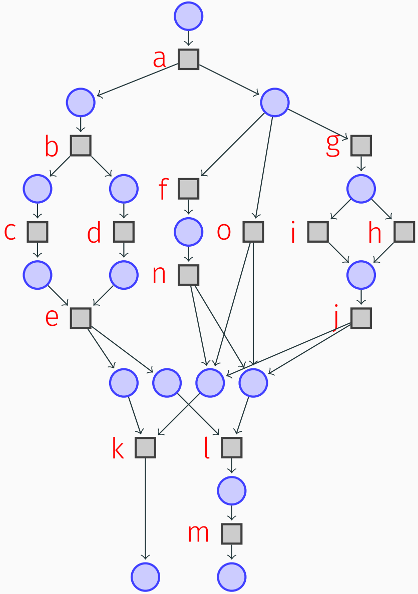
Sampling executions **uniformly at random** in a given SP_{+} program.

Idea 1: listing of all possible executions? **No**

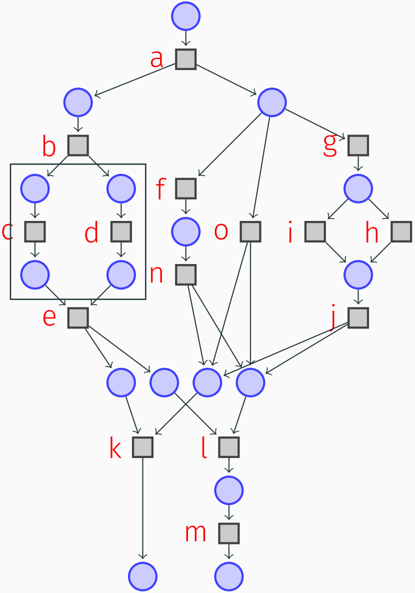
Idea 2: expand all the global choices and choose one? **No**
$\hookrightarrow$ [CSR'17] gives an algorithm for the choice-free case

Solution: use the symbolic method to select a global choice (next slides)

Algorithm (step 1): specify the executions of the program



Algorithm (step 1): specify the executions of the program



$$N \parallel M$$

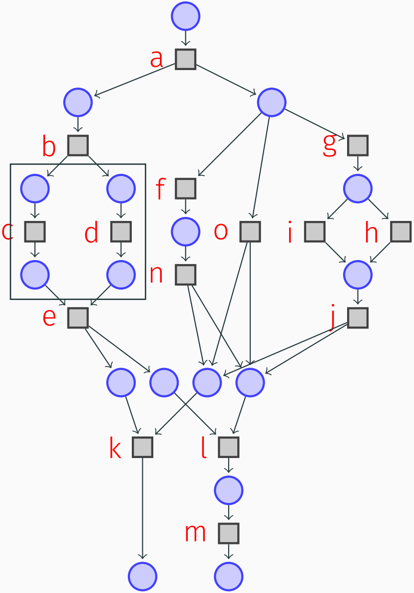
$$\downarrow$$

$$\mathcal{N} \star \mathcal{M}$$

$$\downarrow$$

$$N(z)M(z)$$

Algorithm (step 1): specify the executions of the program



$$c \parallel d$$

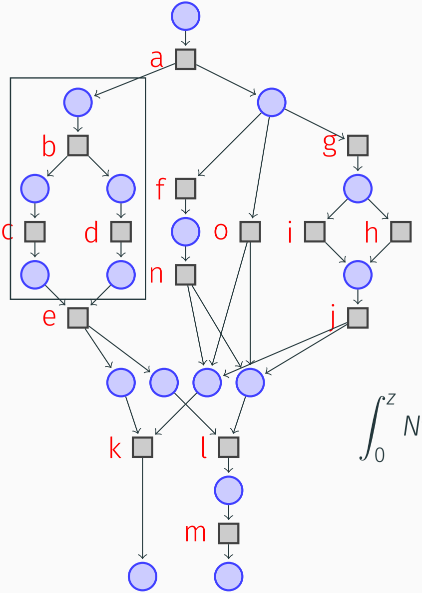
$$\downarrow$$

$$\mathcal{Z} \star \mathcal{Z}$$

$$\downarrow$$

$$z \cdot z = 2 \cdot \frac{z^2}{2!}$$

Algorithm (step 1): specify the executions of the program



$$N; M$$

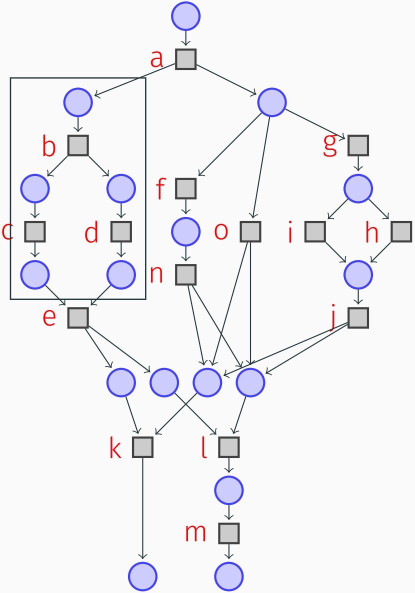
$$\downarrow$$

$$\mathcal{N} \boxed{\star} \mathcal{M}$$

$$\downarrow$$

$$\int_0^z N'(z-u)M(u)du + N(0)M(z)$$

Algorithm (step 1): specify the executions of the program



$$b; (c \parallel d)$$

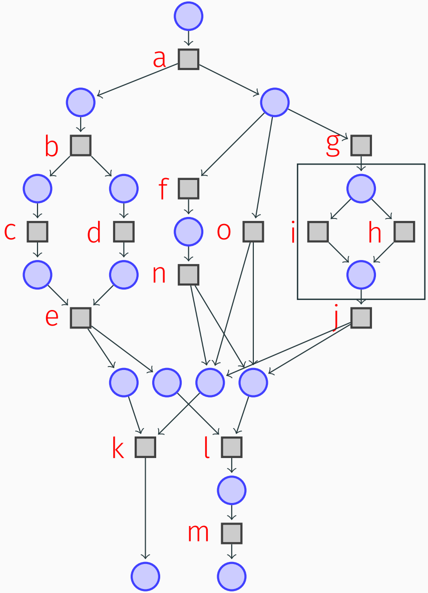
$$\downarrow$$

$$\mathcal{Z} \boxed{\star} (\mathcal{Z} \star \mathcal{Z})$$

$$\downarrow$$

$$\int_0^z u^2 du = 2 \cdot \frac{z^3}{3!}$$

Algorithm (step 1): specify the executions of the program



$$N + M$$

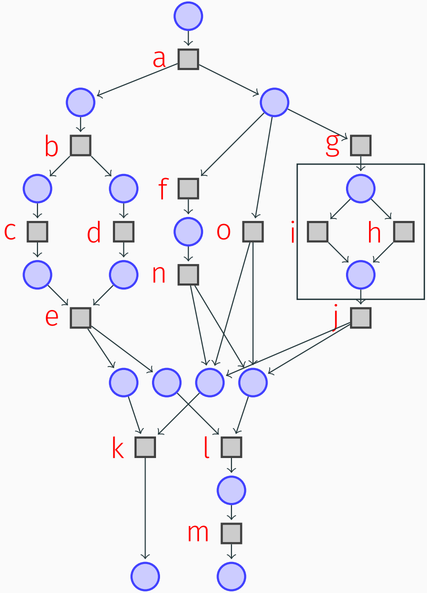
$$\downarrow$$

$$(\quad \mathcal{N} + \quad \mathcal{M})$$

$$\downarrow$$

$$N(z) + \quad M(z)$$

Algorithm (step 1): specify the executions of the program



$$N + M$$

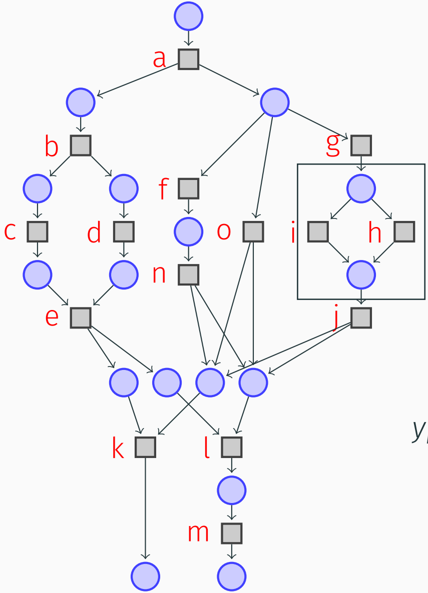
$$\downarrow$$

$$(\mathcal{Y}_i \mathcal{N} + \mathcal{Y}_h \mathcal{M})$$

$$\downarrow$$

$$y_i N(z) + y_h M(z)$$

Algorithm (step 1): specify the executions of the program



$$i + h$$

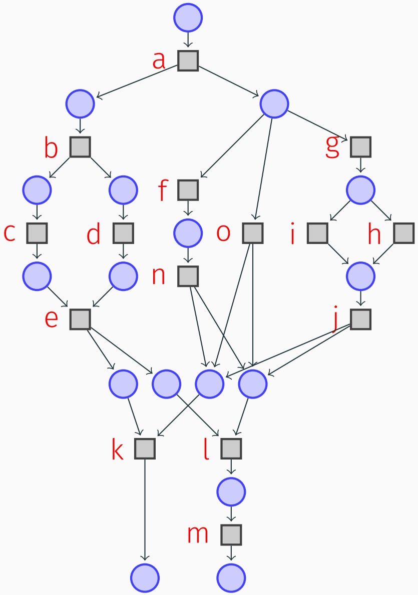
$$\downarrow$$

$$(\mathcal{Y}_i \mathcal{Z} + \mathcal{Y}_h \mathcal{Z})$$

$$\downarrow$$

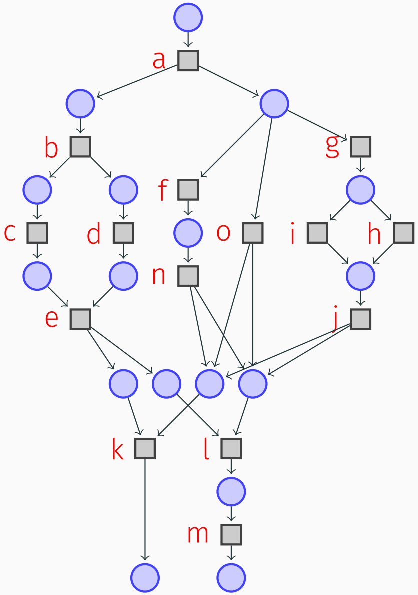
$$y_i z + y_h z = (1 \cdot y_i + 1 \cdot y_h) \frac{z^1}{1!}$$

Algorithm (step 2): backward symbolic method



$$30 y_o \frac{z^9}{9!} + 90 y_f \frac{z^{10}}{10!} + 210 y_g (y_h + y_i) \frac{z^{11}}{11!}$$

Algorithm (step 2): backward symbolic method

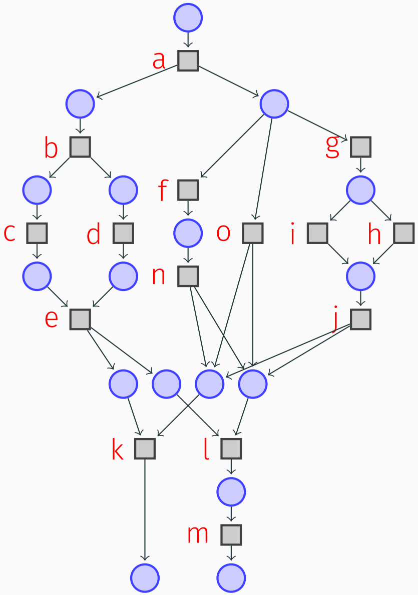


$$30y_o\frac{z^9}{9!}+90y_f\frac{z^{10}}{10!}+210y_g(y_h+y_i)\frac{z^{11}}{11!}$$

$$\Big\downarrow\, y_{\dots} \leftarrow 1$$

$$30\frac{z^9}{9!}+90\frac{z^{10}}{10!}+420\frac{z^{11}}{11!}$$

Algorithm (step 2): backward symbolic method



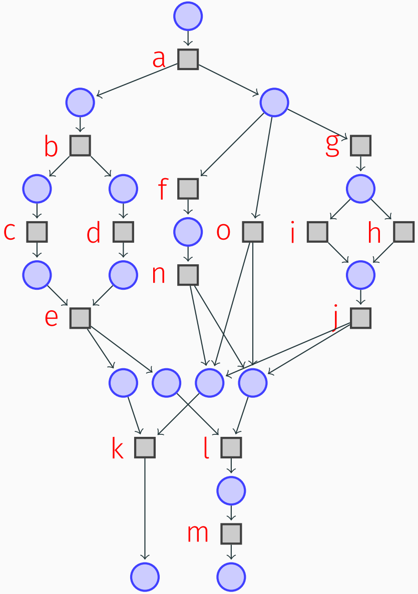
$$30y_o\frac{z^9}{9!}+90y_f\frac{z^{10}}{10!}+210y_g(y_h+y_i)\frac{z^{11}}{11!}$$

$$\Big\downarrow\, {\color{blue}y_{\dots}} \leftarrow 1$$

$$30\frac{z^9}{9!}+90\frac{z^{10}}{10!}+420\frac{z^{11}}{11!}$$

# of executions =

540



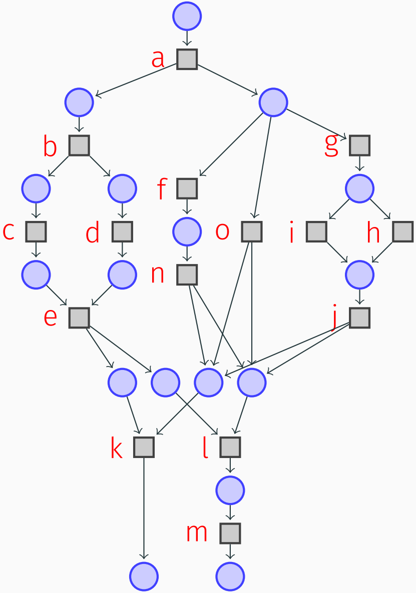
$$30y_o\frac{z^9}{9!}+90y_f\frac{z^{10}}{10!}+210y_g(y_h+y_i)\frac{z^{11}}{11!}$$

$$\Big\downarrow\, y_{\dots} \leftarrow 1$$

$$30\frac{z^9}{9!}+90\frac{z^{10}}{10!}+420\frac{z^{11}}{11!}$$

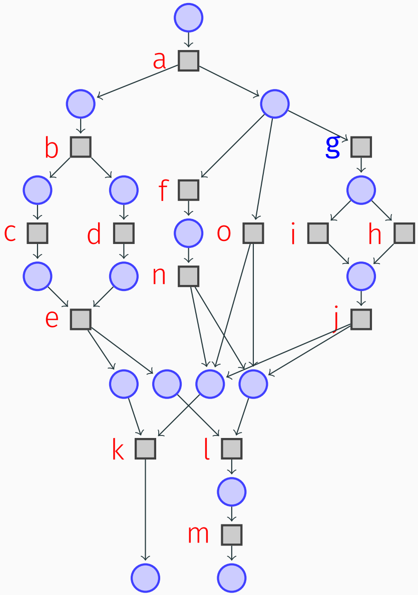
$$\mathbb{P}[\text{exec size}=11]=\frac{420}{540}$$

Algorithm (step 2): backward symbolic method



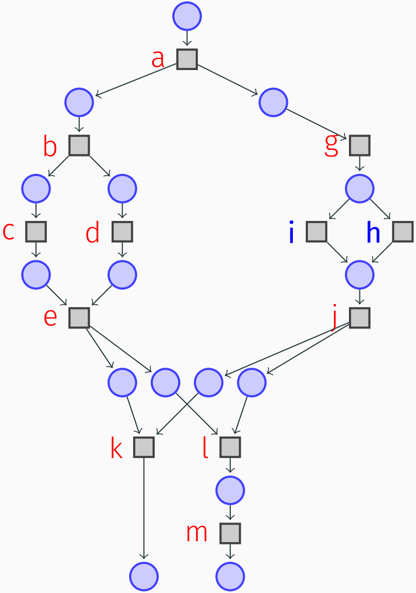
$$210 y_g (y_h + y_i) \frac{z^{11}}{11!}$$

Algorithm (step 2): backward symbolic method



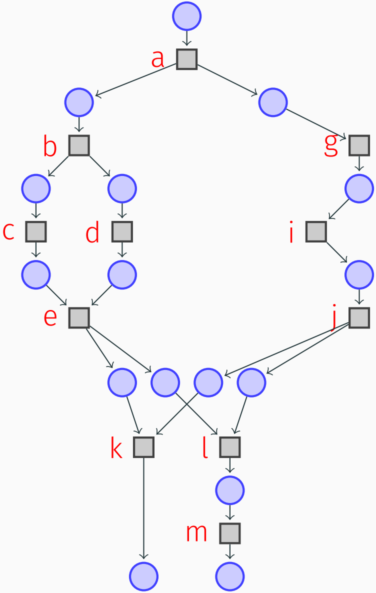
$$210\mathbf{y_g}(y_h + y_i)\frac{z^{11}}{11!}$$

Algorithm (step 2): backward symbolic method



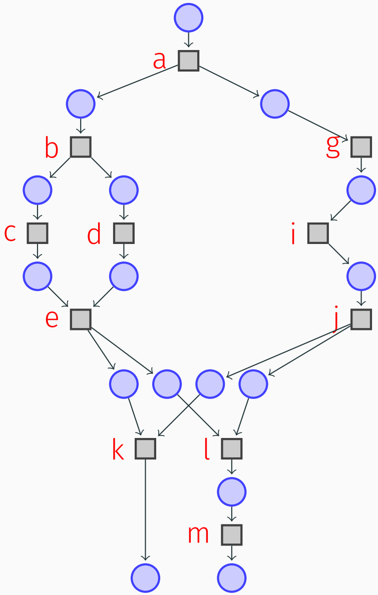
$$210 y_g (y_h + y_i) \frac{z^{11}}{11!}$$

Algorithm (step 2): backward symbolic method



$$210 y_g (y_h + y_i) \frac{z^{11}}{11!}$$

Algorithm (step 2): backward symbolic method



$$210 y_g (y_h + y_i) \frac{z^{11}}{11!}$$

There remains to sample an execution in the resulting choice-free graph

$\rightarrow$ [CSR'17]

Analysis of the algorithm

Correction: symbolic method

Worst case complexity ($n =$ size of the graph):

- Step 1: size of the polynomial: $O(n^2)$
- Step 2: number of arithmetic operations on big integers: $O(n^2)$
- Choice-free algorithm [CSR'17]: $O(n^2)$

Conclusion

- A class of petri nets that captures the features we want to study
- A realistic and tractable subclass: SP_+
- An efficient uniform random generator of executions

Conclusion

- A class of petri nets that captures the features we want to study
- A realistic and tractable subclass: SP_+
- An efficient uniform random generator of executions

In the future:

- Quantitative results
- Beyond SP_+: unfold **some** choices? Modelling loops?
- Recognize that a petri net is in SP_+
- Implement a statistical model checker based on our random generation techniques

Thank you!

Questions?

Sampling in a choice-free graph

Input: A choice-free SP_+
Output: A uniform execution of this program (list of actions)

function SAMPLE_CF(P)
  if $P = a$ **then**
    return [a]
  else if $P = P_1 \parallel P_2$ **then**
    return SHUFFLE(SAMPLE_CF(P_1), SAMPLE_CF(P_2))
  else if $P = P_1; P_2$ **then**
    return CONCAT(SAMPLE_CF(P_1), SAMPLE_CF(P_2))

References i

Olivier Bodini, Matthieu Dien, Antoine Genitrini, and Frédéric Peschanski.
The Ordered and Colored Products in Analytic Combinatorics: Application to the Quantitative Study of Synchronizations in Concurrent Processes.
In *14th SIAM Meeting on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 16–30, 2017.

Olivier Bodini, Antoine Genitrini, and Frédéric Peschanski.
The Combinatorics of Non-determinism.
In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2013)*, volume 24 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 425–436, Guwahati, India, 2013.

Olivier Bodini, Antoine Genitrini, and Frédéric Peschanski.
A Quantitative Study of Pure Parallel Processes.
Electronic Journal of Combinatorics, 23(1):P1.11, 39 pages, (electronic), 2016.

Graham Brightwell and Peter Winkler.
Counting linear extensions.
Order, 8(3):225–242, 1991.

Philippe Flajolet and Robert Sedgewick.
Analytic Combinatorics.
Cambridge University Press, 2009.