Analysis of Probabilistic Basic Parallel Processes

Rémi Bonnet[1] *Stefan Kiefer*[1] Anthony W. Lin[1,2]

[1]University of Oxford, UK

[2]Academia Sinica, Taiwan

FoSSaCS, Grenoble
9 April 2014

Probabilistic Basic Parallel Processes (pBPP)

The rules of a probabilistic
Basic Parallel Process (pBPP):

$$X \xhookrightarrow{0.7} XY$$

$$X \xhookrightarrow{0.3} \varepsilon$$

$$Y \xhookrightarrow{0.6} YY$$

$$Y \xhookrightarrow{0.4} \varepsilon$$

A run: $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow Y \Rightarrow YY \Rightarrow Y \Rightarrow \varepsilon$

Probabilistic Basic Parallel Processes (pBPP)

A run as a growing tree:

The rules of a probabilistic Basic Parallel Process (pBPP):

$$X \overset{0.7}{\hookrightarrow} XY$$

$$X \overset{0.3}{\hookrightarrow} \varepsilon$$

$$Y \overset{0.6}{\hookrightarrow} YY$$

$$Y \overset{0.4}{\hookrightarrow} \varepsilon$$

A run: $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow Y \Rightarrow YY \Rightarrow Y \Rightarrow \varepsilon$

Probabilistic Basic Parallel Processes (pBPP)

A run as a growing tree:

X

The rules of a probabilistic Basic Parallel Process (pBPP):

$$X \overset{0.7}{\hookrightarrow} XY$$

$$X \overset{0.3}{\hookrightarrow} \varepsilon$$

$$Y \overset{0.6}{\hookrightarrow} YY$$

$$Y \overset{0.4}{\hookrightarrow} \varepsilon$$

A run: $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow Y \Rightarrow YY \Rightarrow Y \Rightarrow \varepsilon$

Probabilistic Basic Parallel Processes (pBPP)

The rules of a probabilistic Basic Parallel Process (pBPP):

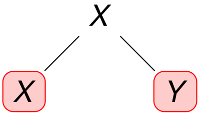
$$X \xhookrightarrow{0.7} XY$$

$$X \xhookrightarrow{0.3} \varepsilon$$

$$Y \xhookrightarrow{0.6} YY$$

$$Y \xhookrightarrow{0.4} \varepsilon$$

A run as a growing tree:



A run: $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow Y \Rightarrow YY \Rightarrow Y \Rightarrow \varepsilon$

Probabilistic Basic Parallel Processes (pBPP)

The rules of a probabilistic Basic Parallel Process (pBPP):

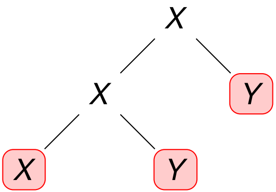
$$X \xhookrightarrow{0.7} XY$$

$$X \xhookrightarrow{0.3} \varepsilon$$

$$Y \xhookrightarrow{0.6} YY$$

$$Y \xhookrightarrow{0.4} \varepsilon$$

A run as a growing tree:



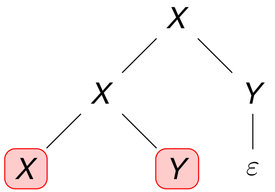
A run: $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow Y \Rightarrow YY \Rightarrow Y \Rightarrow \varepsilon$

Probabilistic Basic Parallel Processes (pBPP)

The rules of a probabilistic Basic Parallel Process (pBPP):

$$X \xhookrightarrow{0.7} XY$$
$$X \xhookrightarrow{0.3} \varepsilon$$
$$Y \xhookrightarrow{0.6} YY$$
$$Y \xhookrightarrow{0.4} \varepsilon$$

A run as a growing tree:



A run: $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow Y \Rightarrow YY \Rightarrow Y \Rightarrow \varepsilon$

Probabilistic Basic Parallel Processes (pBPP)

The rules of a probabilistic Basic Parallel Process (pBPP):

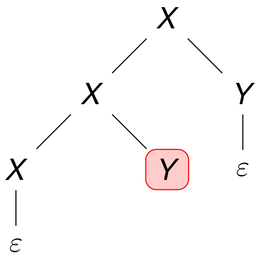
$$X \overset{0.7}{\hookrightarrow} XY$$

$$X \overset{0.3}{\hookrightarrow} \varepsilon$$

$$Y \overset{0.6}{\hookrightarrow} YY$$

$$Y \overset{0.4}{\hookrightarrow} \varepsilon$$

A run as a growing tree:



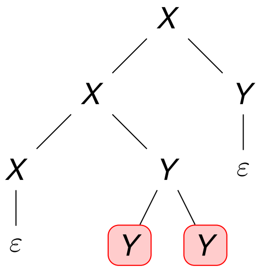
A run: $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow Y \Rightarrow YY \Rightarrow Y \Rightarrow \varepsilon$

Probabilistic Basic Parallel Processes (pBPP)

The rules of a probabilistic Basic Parallel Process (pBPP):

$$X \overset{0.7}{\hookrightarrow} XY$$
$$X \overset{0.3}{\hookrightarrow} \varepsilon$$
$$Y \overset{0.6}{\hookrightarrow} YY$$
$$Y \overset{0.4}{\hookrightarrow} \varepsilon$$

A run as a growing tree:



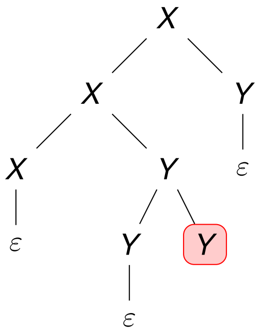
A run: $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow Y \Rightarrow YY \Rightarrow Y \Rightarrow \varepsilon$

Probabilistic Basic Parallel Processes (pBPP)

The rules of a probabilistic Basic Parallel Process (pBPP):

$$X \xhookrightarrow{0.7} XY$$
$$X \xhookrightarrow{0.3} \varepsilon$$
$$Y \xhookrightarrow{0.6} YY$$
$$Y \xhookrightarrow{0.4} \varepsilon$$

A run as a growing tree:



A run: $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow Y \Rightarrow YY \Rightarrow \color{red}{Y} \Rightarrow \varepsilon$

Probabilistic Basic Parallel Processes (pBPP)

The rules of a probabilistic Basic Parallel Process (pBPP):

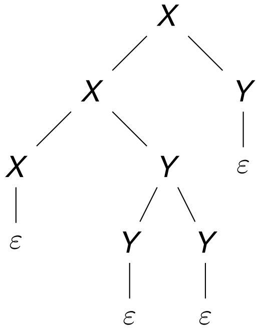
$$X \xhookrightarrow{0.7} XY$$

$$X \xhookrightarrow{0.3} \varepsilon$$

$$Y \xhookrightarrow{0.6} YY$$

$$Y \xhookrightarrow{0.4} \varepsilon$$

A run as a growing tree:



A run: $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow Y \Rightarrow YY \Rightarrow Y \Rightarrow \varepsilon$

Probabilistic Basic Parallel Processes (pBPP)

The rules of a probabilistic Basic Parallel Process (pBPP):

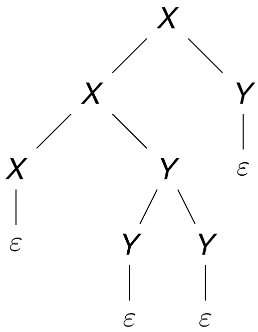
$$X \overset{0.7}{\hookrightarrow} XY$$

$$X \overset{0.3}{\hookrightarrow} \varepsilon$$

$$Y \overset{0.6}{\hookrightarrow} YY$$

$$Y \overset{0.4}{\hookrightarrow} \varepsilon$$

A run as a growing tree:



A run: $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow Y \Rightarrow YY \Rightarrow Y \Rightarrow \varepsilon$

Runs are random.

The order of the “nonterminals” is not important $\Rightarrow$ SPNs.

Which Nonterminals are Picked?

- Either randomly $\longrightarrow$ Markov chain
  - either uniformly with multiplicities
  - either uniformly without multiplicities

  Consider state XYX.
  With multiplicities: probability of scheduling X is $2/3$.
  Without multiplicities: probability of scheduling X is $1/2$.

  Both versions make sense. The same results hold.

- Or nondeterministically (by a scheduler) $\longrightarrow$ MDP

The set of states is $\mathbb{N}^{\Gamma}$, an infinite set,

where $\Gamma :=$ set of nonterminals.

Notation:

run: $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots$ with states $\alpha_i \in \mathbb{N}^{\Gamma}$

Coverability

Given a pBPP, a start state $\alpha_0 \in \mathbb{N}^\Gamma$ and a state $\phi \in \mathbb{N}^\Gamma$.

A run $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots$ covers ϕ
if there is $i \geq 0$ with $\alpha_i \geq \phi$ (componentwise).

Given also a finite "target" set $F = \{\phi_1, \ldots, \phi_k\} \subset \mathbb{N}^\Gamma$.

A run covers F if it covers some $\phi_j \in F$.

Coverability problem: Given a pBPP and α_0 and a target set F:

Starting from α_0, is F covered with probability 1 ?

= Reachability of an upward-closed set $F\uparrow$ with probability 1

Applications of Coverability

$$F := \{Producer\ Consumer\}$$

Covering F = Transaction between a Producer and a Consumer can take place

$$F := \{Grantrequest\}$$

Covering F = (At least) one request is granted

Examples for Coverability

$$X \overset{1}{\hookrightarrow} XY \text{ and } Y \overset{1}{\hookrightarrow} \varepsilon$$
$$\text{start state } \alpha_0 = X$$
$$\text{target set } F = \{YYY\}$$

F is covered with probability 1, i.e.,
runs like $X \Rightarrow XY \Rightarrow XYY \Rightarrow XY \Rightarrow XYY \Rightarrow XYYY \Rightarrow \ldots$
have (together) probability 1.

This is true even though there are runs that don't cover F,
like $X \Rightarrow XY \Rightarrow X \Rightarrow XY \Rightarrow X \Rightarrow \ldots$
(they have together probability 0)

Examples for Coverability

$$X \overset{0.7}{\hookrightarrow} XX \qquad\qquad Y \overset{1}{\hookrightarrow} Y$$

$$X \overset{0.3}{\hookrightarrow} Y$$

$$\alpha_0 = X$$
$$F = \{XXX\}$$

Runs of the form $X \Rightarrow Y \Rightarrow Y \Rightarrow \ldots$ have probability 0.3, so the probability of covering F is < 1 (but positive).

Reaching a Trap

Fix a pBPP, an initial state α_0, and a target set $F \subset \mathbb{N}^\Gamma$.

Write *Trap* $:=$ those states from which one cannot reach $F\uparrow$

Proposition (builds on [Abdulla, Ben Henda, Mayr, '07])

F is covered from α_0 with probability 1

$$\iff$$

From α_0 one cannot reach a trap while avoiding $F\uparrow$.

One direction is easy: if one can reach a trap while avoiding $F\uparrow$, then there is a finite path to do so. That path has a positive probability.

The other direction is less immediate.

Purely qualitative.

Karp-Miller-Style Algorithm for Coverability

Proposition (builds on [Abdulla, Ben Henda, Mayr, '07])

F is covered from α_0 with probability 1

$$\Longleftrightarrow$$

From α_0 one cannot reach a trap while avoiding $F\uparrow$.

Idea: decide coverability using the "trap" criterion.
$\longrightarrow$ Karp-Miller-Like Tree

Example: $\alpha_0 = X$ $\quad X \xhookrightarrow{0.7} XX \quad Y \xhookrightarrow{1} Y$

$F = \{XXX\}$ $\quad X \xhookrightarrow{0.3} Y$

X

Karp-Miller-Style Algorithm for Coverability

Proposition (builds on [Abdulla, Ben Henda, Mayr, '07])

F is covered from α_0 with probability 1

$$\Longleftrightarrow$$

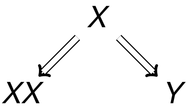
From α_0 one cannot reach a trap while avoiding $F\uparrow$.

Idea: decide coverability using the "trap" criterion.

$\longrightarrow$ Karp-Miller-Like Tree

Example: $\alpha_0 = X$ $\qquad$ $X \overset{0.7}{\hookrightarrow} XX$ $\qquad$ $Y \overset{1}{\hookrightarrow} Y$

$F = \{XXX\}$ $\qquad$ $X \overset{0.3}{\hookrightarrow} Y$



Karp-Miller-Style Algorithm for Coverability

Proposition (builds on [Abdulla, Ben Henda, Mayr, ’07])

F is covered from α_0 with probability 1

$$\Longleftrightarrow$$

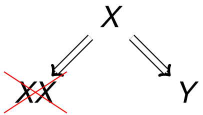
From α_0 one cannot reach a trap while avoiding $F\uparrow$.

Idea: decide coverability using the “trap” criterion.

$\longrightarrow$ Karp-Miller-Like Tree

Example: $\alpha_0 = X$ $\quad X \overset{0.7}{\hookrightarrow} XX \quad Y \overset{1}{\hookrightarrow} Y$

$F = \{XXX\}$ $\quad X \overset{0.3}{\hookrightarrow} Y$



Karp-Miller-Style Algorithm for Coverability

Proposition (builds on [Abdulla, Ben Henda, Mayr, '07])

F is covered from α_0 with probability 1

$$\iff$$

From α_0 one cannot reach a trap while avoiding $F\uparrow$.

Idea: decide coverability using the "trap" criterion.

$\longrightarrow$ Karp-Miller-Like Tree

Example: $\alpha_0 = X$ $\quad$ $X \xhookrightarrow{0.7} XX$ $\quad$ $Y \xhookrightarrow{1} Y$

$F = \{XXX\}$ $\quad$ $X \xhookrightarrow{0.3} Y$

X

$\swarrow$ ~~XX~~ $\quad$ $\searrow$ $Y \in$ *Trap*, so we've found a path to a trap

Karp-Miller-Style Algorithm for Coverability

Proposition (builds on [Abdulla, Ben Henda, Mayr, '07])

F is covered from α_0 with probability 1

$$\Longleftrightarrow$$

From α_0 one cannot reach a trap while avoiding $F\uparrow$.

The other example from before:

$$\alpha_0 = X \qquad F = \{YYY\} \qquad X \overset{1}{\hookrightarrow} XY \text{ and } Y \overset{1}{\hookrightarrow} \varepsilon$$

$$X$$
$$\Downarrow$$
$$XY$$

Karp-Miller-Style Algorithm for Coverability

Proposition (builds on [Abdulla, Ben Henda, Mayr, '07])

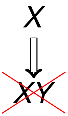
F is covered from α_0 with probability 1

$$\Longleftrightarrow$$

From α_0 one cannot reach a trap while avoiding $F\uparrow$.

The other example from before:

$$\alpha_0 = X \qquad F = \{YYY\} \qquad X \overset{1}{\hookrightarrow} XY \text{ and } Y \overset{1}{\hookrightarrow} \varepsilon$$



Karp-Miller-Style Algorithm for Coverability

Proposition (builds on [Abdulla, Ben Henda, Mayr, '07])

F is covered from α_0 with probability 1

$$\Longleftrightarrow$$

From α_0 one cannot reach a trap while avoiding $F{\uparrow}$.

The other example from before:

$$\alpha_0 = X \qquad F = \{YYY\} \qquad X \overset{1}{\hookrightarrow} XY \text{ and } Y \overset{1}{\hookrightarrow} \varepsilon$$

X

$\Downarrow$

~~XY~~

So one cannot reach a trap while avoiding $F{\uparrow}$.
(In fact, one cannot reach a trap at all.)

Karp-Miller Style

Try to find paths to a trap:

- Build a tree breadth-first starting from α_0:
  nodes = states branches = runs
- Never include states that cover F.
- If a configuration is larger than a predecessor, prune.
- If a leaf is a trap, return "coverability with prob < 1".
- If all leaves are pruned, return "coverability with prob 1".

Termination: Dickson's lemma, König's lemma
Correctness: "Smaller is better"

Results

That was a proof (sketch) of:

Theorem

The coverability problem is decidable:
Given a pBPP, an initial state α_0, a target set F.
One can decide whether F covered with probability 1
(when starting in α_0).

This is in contrast to general stochastic Petri nets
(as established in [Abdulla, Ben Henda, Mayr, '07]).

Results

That was a proof (sketch) of:

Theorem

The coverability problem is decidable:
Given a pBPP, an initial state α_0, a target set F.
One can decide whether F covered with probability 1 (when starting in α_0).

This is in contrast to general stochastic Petri nets (as established in [Abdulla, Ben Henda, Mayr, '07]).

However:

Theorem

The coverability problem has nonelementary complexity (is Tower-hard).

Proof. Long. By a reduction from a 2-counter machine with a Tower-budget.

Results for MDPs

Theorem

Given a pBPP, and α_0 and F as before.

One can decide whether there exists a scheduler such that F is covered with probability 1 (when starting in α_0).

If yes, then there is a memoryless deterministic scheduler, and one can compute such a scheduler.

Proof. By abstracting the state space (finite subset).
Relies on Petri-Net reachability
$\longrightarrow$ no upper complexity bound.

The MDP and Markov-Chain problems are rather different.

Results for MDPs

Theorem

Given a pBPP, and α_0 and F as before. And $k \in \mathbb{N}$.

One can decide whether for all k-fair schedulers F is covered with probability 1 (when starting in α_0).

k-fairness means: if an *X*-nonterminal is present, then an *X*-nonterminal must be scheduled within *k* steps.

Q-States Targets

Theorem

Given a pBPP, and α_0. Given also $F = \{X_1, \ldots, X_j\}$.

- *F is covered with probability 1 in the Markov-Chain model*
  $\iff$ F is covered with prob 1 for some scheduler
  $\iff$ F is covered with prob 1 for all k-fair schedulers.
  ($k \geq |\Gamma|$)
- *One can decide in polynomial time*
  whether F is covered with probability 1.

Proof. Simple.

An analogous restriction leads to decidability for general VASSs [Abdulla, Ben Henda, Mayr, '07] (no complexity bound).

Semilinear Targets

Theorem

Given a pBPP, and α_0.
Given also a semilinear target set $S \subseteq \mathbb{N}^\Gamma$.
The following problems are undecidable:

(a) *Is S reached with probability 1?*
(b) *Is S reached with probability 1 for some scheduler?*
(c) *Is S reached with probability 1 for all 7-fair schedulers?*

Proof. Reduction from 2-counter machines.

Concluding Remarks

- pBPP are simple and natural stochastic Petri nets.
- The considered problems are qualitative in two senses. Laws of probability impose a special but natural kind of fairness.
- Variations quickly lead to multi-dimensional random walks.