Parameterised Systems

Daniel Stan, Automated Reasoning group, TUK
Parameterised System

**Motivations**: analysis of a whole family of systems $(S_p)_p$.

$p \in Param$ is a parameter.

- **Synthesis**: “Does there exists $p \in Param$ such that $S_p \models \varphi$?”

- **Validity**: “Does $S_p \models \varphi$ for all $p$?”
Different types of parameters:

- Guard constants values (in timed automata):
  \( \text{Param} \subseteq \mathbb{R}^+ \);

- Probability values (in stochastic systems):
  \( \text{Param} \subseteq [0,1] \);

- ...  

- Number of interacting copies of processes:
  \( \text{Param} = \mathbb{N} \).
Parameterised System

**Different** types of parameters:

- Guard constants values (in timed automata): $\text{Param} \subseteq \mathbb{R}^+$;
- Probability values (in stochastic systems): $\text{Param} \subseteq [0, 1]$;
- ...  
- Number of interacting copies of processes: $\text{Param} = \mathbb{N}$.

**Today:** Validity of parameterised systems composed with **arbitrary many copies** using **well-quasi order** techniques.
Arbitrary many copies
Arbitrary many copies

**Definition**
A parameterised system $\mathcal{P} = (S, I, \Sigma, \rightarrow)$ is a LTS $(S, \Sigma, \rightarrow)$ equipped with

- A WQO $\preceq$ over $S$;
- An upward-closed set of initial states $I$;
- A norm function $|.| : S \rightarrow \mathbb{N}$ such that $\forall (s, t), s \leq t \Rightarrow |s| \leq |t|$

We write $S_k = \{s \in S \mid |s| = k\}$ and $\mathcal{P}_k$ the LTS restricted to initial states in $I \cap S_k$. 

\[ \leq \]
Arbitrary many copies

**Definition**
A parameterised system $\mathcal{P} = (S, I, \Sigma, \rightarrow)$ is a LTS $(S, \Sigma, \rightarrow)$ equipped with:

- A WQO $\preceq$ over $S$;
- An upward-closed set of initial states $I$;
- A norm function $|.| : S \rightarrow \mathbb{N}$ such that $\forall (s, t), s \preceq t \Rightarrow |s| \leq |t|$.

We write $S_k = \{ s \in S \mid |s| = k \}$ and $\mathcal{P}_k$ the LTS restricted to initial states in $I \cap S_k$.

Not necessarily a WSTS!
Some Petri Net example

\[ I = \{ k \cdot p_0 \mid k \in \mathbb{N} \} , \]
\[ |M| = \ldots , \]
\[ M \preceq M' \text{ if } \forall p \in P, M(p) \leq M'(p) \]

\[ |m| = m(p_0) + m(p_c) + m(p_d) \]

\[ I = \uparrow p_0 + p_e \]

\[ \text{Bad} = \uparrow 2p_c \]
Some Petri Net example

\[ I = \{ k \cdot p_0 \mid k \in \mathbb{N} \}, \]

\[ |M| = \ldots, \]

\[ M \preceq M' \text{ if } \forall p \in P, M(p) \leq M'(p) \text{ and } \]

\[ \text{support}(M) = \{ p \mid M(p) > 0 \} = \{ p \mid M'(p) > 0 \} = \text{support}(M'). \]

(still a WQO)

Remark:

\[ |M| \text{ is not necessarily the number of tokens in } M. \]
**Parameterised Systems (Our definition today)**

**Definition**
A parameterised system $\mathcal{P} = (S, I, \Sigma, \rightarrow)$ is a LTS $(S, \Sigma, \rightarrow)$ equipped with

- A WQO $\leq$ over $S$;
- An upward-closed set of initial states $I$;
- A norm function $|.| : S \rightarrow \mathbb{N}$ such that $\forall (s, t), s \leq t \Rightarrow |s| \leq |t|$.
- $\rightarrow$ preserves the norm
- $S_k$ is finite for any $k$.

We write $S_k = \{s \in S \mid |s| = k\}$ and $\mathcal{P}_k$ the LTS restricted to initial states in $I \cap S_k$. 
Cut-off property

**Definition**
A parameterised system $(\mathcal{L}_n)_n$ has a **cut-off property** for $\varphi$ if there exists a **cut-off bound** $n \in \mathbb{N}$, such that

$$\forall k > n, \mathcal{L}_k \models \varphi \iff \mathcal{L}_n \models \varphi$$

Namely, one of the two situations can happen:
Cut-off property

**Definition**
A parameterised system $(\mathcal{S}_n)_n$ has a cut-off property for $\varphi$ if there exists a cut-off bound $n \in \mathbb{N}$, such that

$$\forall k > n, \mathcal{S}_k \models \varphi \iff \mathcal{S}_n \models \varphi$$

Namely, one of the two situations can happen:

- $\forall n \geq k \mathcal{S}_n \models \varphi$: Positive cut-off;
**Definition**
A parameterised system $(\mathcal{L}_n)_n$ has a **cut-off property** for $\varphi$ if there exists a **cut-off bound** $n \in \mathbb{N}$, such that

$$\forall k > n, \mathcal{L}_k \models \varphi \iff \mathcal{L}_n \models \varphi$$

Namely, one of the two situations can happen:

- $\forall n \geq k \mathcal{L}_n \models \varphi$: **Positive** cut-off;

![Diagram showing positive cut-off](image)
Cut-off property

**Definition**
A parameterised system \((\mathcal{S}_n)_n\) has a **cut-off property** for \(\varphi\) if there exists a **cut-off bound** \(n \in \mathbb{N}\), such that

\[
\forall k > n, \mathcal{S}_k \models \varphi \iff \mathcal{S}_n \models \varphi
\]

Namely, one of the two situations can happen:

- \(\forall n \geq k \mathcal{S}_n \models \varphi\): **Positive** cut-off;

- \(\forall n \geq k \mathcal{S}_n \not\models \varphi\): **Negative** cut-off;
Cut-off property

**Definition**
A parameterised system \((\mathcal{I}_n)_n\) has a **cut-off property** for \(\varphi\) if there exists a **cut-off bound** \(n \in \mathbb{N}\), such that

\[\forall k > n, \mathcal{I}_k \models \varphi \iff \mathcal{I}_n \models \varphi\]

Namely, one of the two situations can happen:

- \(\forall n \geq k \mathcal{I}_n \models \varphi\): **Positive** cut-off;
  
  \[
  \begin{array}{cccccccc}
  & & & & & & & \\
  & X & \checkmark & X & \checkmark & \checkmark & \checkmark & \checkmark & \rightarrow n \\
  1 & 2 & 3 & 11 & 12 & 13 & 14 & \\
  \end{array}
  \]

- \(\forall n \geq k \mathcal{I}_n \not\models \varphi\): **Negative** cut-off;
  
  \[
  \begin{array}{cccccccc}
  & & & & & & & \\
  & \checkmark & X & X & \checkmark & X & X & X & \rightarrow n \\
  1 & 2 & 3 & 4 & 15 & 16 & 17 & \\
  \end{array}
  \]
Cut-off point for coverability Safety in Petri Net
Cut-off point for coverability Safety in Petri Net: general procedure
Cut-off point for coverability Safety in Petri Net: general procedure

**Problem**: Given a Petri $N$, with a set of upward-closed initial marking $I = \uparrow p_0$ and an upward closed set $Bad$. Is there a marking $M \in I$ that can reach $Bad$?
Cut-off point for coverability Safety in Petri Net: general procedure

**Problem**: Given a Petri Net $N$, with a set of upward-closed initial marking $I = \uparrow p_0$ and an upward closed set $Bad$. Is there a marking $M \in I$ that can reach $Bad$?

- **Karp-Miller tree** from $M_0 = \omega \cdot p_0$. 
Cut-off point for coverability Safety in Petri Net: general procedure

**Problem**: Given a Petri $N$, with a set of upward-closed initial marking $I = \uparrow p_0$ and an upward closed set $Bad$. Is there a marking $M \in I$ that can reach $Bad$?

- **Karp-Miller tree** from $M_0 = \omega \cdot p_0$.
- (Recall week 3): for any $M$, if there is $M'$ in $\text{KarpMillerTree}(N)$ with $M \leq M'$, then $N$ can cover $M$. 
**Problem**: Given a Petri Net $N$, with a set of upward-closed initial marking $I = \uparrow p_0$ and an upward closed set $Bad$. Is there a marking $M \in I$ that can reach $Bad$?

- **Karp-Miller tree** from $M_0 = \omega \cdot p_0$.
- (Recall week 3): for any $M$, if there is $M'$ in $KarpMillerTree(N)$ with $M \leq M'$, then $N$ can cover $M$.

**Conclusion**: the problem is EXPSPACE-complete.
Recall: Backward coverability for WSTS

\[
\text{function } \text{BackCoverFormal}(N, p_0, \text{Bad}, \leq) \\
\text{ } U \leftarrow \text{Bad} \\
\text{ } U' \leftarrow \emptyset \\
\text{while } U \neq U' \text{ do} \\
\text{ } U' \leftarrow U \\
\text{ } U \leftarrow U \cup \text{Pre}(U) \\
\text{return } \exists k : k \cdot p_0 \in U
\]
Non-atomic operations on shared variables

Inspired by:

- *Model checking parameterized asynchronous shared-memory systems.*
  Antoine Durand-Gasselin, Javier Esparza, Pierre Ganty, Rupak Majumdar, 2017

- *Reachability in Networks of Register Protocols under Stochastic Schedulers*
  Patricia Bouyer, Nicolas Markey, Mickael Randour, Arnaud Sangnier, Daniel Stan, 2016
The model

Register protocol with $D = \{0, 1, 2\}$.

**Definition: register protocol**

$\mathcal{P} = \langle Q, D, q_0, d_0, q_f, T \rangle$

- $\langle Q, q_0, q_f, T \rangle$ is a finite state automaton;
- $D$ finite dataset for the shared register;
- $d_0$ an initial value;
- $T \subseteq Q \times \{R, W\} \times D \times Q$ set of transitions, labelled by **read** and **write** operations over $D$. 
The model

\[ R(0) \xrightarrow{W(1)} q_1 \xrightarrow{R(1)} q_2 \xrightarrow{W(2)} R(2) \]
The model

Network for two processes (self-loops omitted).
The model

Network for two processes (self-loops omitted).
The model

Network for two processes (self-loops omitted).
The model

Network for two processes (self-loops omitted).

- There exist paths from there, the processes in $q_0$ are trapped;
The model

Network for two processes (self-loops omitted).

(Non-exhaustive construction)

- There exist paths from there, the processes in $q_0$ are trapped;
The model

Network for two processes (self-loops omitted).

(Non-exhaustive construction)

- There exist paths from there, the processes in $q_0$ are trapped;
There exist paths from there, the processes in $q_0$ are trapped;
There exist paths from there, the processes in $q_0$ are trapped;
The model

Network for two processes (self-loops omitted).

(Non-exhaustive construction)

- There exist paths from there, the processes in $q_0$ are trapped;
Network for two processes (self-loops omitted).

There exist paths from there, the processes in $q_0$ are trapped;
There exist paths that reach $q_f$ . . .
. . . and they require at least two processes.
A **configuration** \( s \in S \) is a multiset of states \( + \) a register value \( d \in D \);
- A **configuration** $s \in S$ is a multiset of states $+$ a register value $d \in D$;

- **Parameter**: once the system is started, the configuration has a fixed size;

- **Interleaving** semantics;

- **Non-atomic** operations (read or write at a time);

- Goal: reach a configuration which **covers** $q_f$. 

![Configuration Diagram]

In this diagram, $q_0$, $q_1$, and $q_2$ represent states, and $d$ represents a register value.
A configuration $s \in S$ is a multiset of states $+$ a register value $d \in D$.

Parameter: once the system is started, the configuration has a fixed size;

Interleaving semantics;

Non-atomic operations (read or write at a time);

Goal: reach a configuration which covers $q_f$.

Semantics:
A configuration $s \in S$ is a multiset of states + a register value $d \in D$;

Parameter: once the system is started, the configuration has a fixed size;

Interleaving semantics;

Non-atomic operations (read or write at a time);

Goal: reach a configuration which covers $q_f$.

Semantics:

How does the scheduler work?
Non-deterministic Scheduler Case: Reachability/Safety

- The scheduler is helpful;
- **Monotonicity**: if $q_f$ is reachable with $n$ initial processes, it is with $n + 1$;
Non-deterministic Scheduler Case: Reachability/Safety

- The scheduler is **helpful**;
- **Monotonicity**: if $q_f$ is reachable with $n$ initial processes, it is with $n + 1$;
Non-deterministic Scheduler Case: Reachability/Safety

- The scheduler is **helpful**;
- **Monotonicity**: if $q_f$ is reachable with $n$ initial processes, it is with $n+1$;

![Diagram](image)

4 5 n
Non-deterministic Scheduler Case: Reachability/Safety

- The scheduler is **helpful**;
- **Monotonicity**: if $q_f$ is reachable with $n$ initial processes, it is with $n+1$;

![Diagram showing reachability for values of n from 4 to 8]
Non-deterministic Scheduler Case: Reachability/Safety

- The scheduler is **helpful**;
- **Monotonicity:** if $q_f$ is reachable with $n$ initial processes, it is with $n+1$;

![Diagram showing reachability and safety for different values of $n$.]
Non-deterministic Scheduler Case: Reachability/Safety

- The scheduler is **helpful**;
- **Monotonicity:** if $q_f$ is reachable with $n$ initial processes, it is with $n + 1$;
- …and there exists a witness of polynomial size.

![Diagram showing x's and checks for different values of n]

Theorem (EGM13)

*Given a protocol, if any, there exists a polynomial path that covers $q_f$.***
Non-deterministic Scheduler Case: Reachability/Safety

- The scheduler is **helpful**;
- **Monotonicity:** if $q_f$ is reachable with $n$ initial processes, it is with $n+1$;
- ... and there exists a witness of polynomial size.

![Diagram showing the reachability of states]

**Theorem (EGM13)**

Given a protocol, if any, there exists a polynomial path that covers $q_f$.

**Proof:** check last appearance and dis-appearance of each control state...
Non-deterministic Scheduler Case:  
Reachability/Safety

- The scheduler is **helpful**;
- **Monotonicity:** if $q_f$ is reachable with $n$ initial processes, it is with $n+1$;
- ... and there exists a witness of polynomial size.

---

Theorem (EGM13)

*Given a protocol, if any, there exists a *polynomial* path that covers $q_f$.*

**Proof:** check last appearance and dis-appearance of each control state...  
Gives a bound on the runtime of the backward coverability algorithm.
Fair, Probabilistic scheduler

- We don’t control the scheduler anymore;
- **Stochastic** behaviour (environment);
- Finite patterns cannot be repeated infinitely often;
- We consider **almost-sure** reachability: $\mathbb{P}_n(\diamond \uparrow q_f) \equiv 1$
Fair, Probabilistic scheduler

- We don’t control the scheduler anymore;
- **Stochastic** behaviour (environment);
- Finite patterns cannot be repeated infinitely often;
- We consider *almost-sure* reachability: $\mathbb{P}_n(\Diamond \uparrow q_f) = 1$
Fair, Probabilistic scheduler

- We don’t control the scheduler anymore;
- **Stochastic** behaviour (environment);
- Finite patterns cannot be repeated infinitely often;
- We consider **almost-sure** reachability: \( \mathbb{P}_n(\Diamond \uparrow q_f) = 1 \)
Fair, Probabilistic scheduler

- We don’t control the scheduler anymore;
- **Stochastic** behaviour (environment);
- Finite patterns cannot be repeated infinitely often;
- We consider **almost-sure** reachability: $\mathbb{P}_n(\Diamond \uparrow q_f) = 1$

Qualitative property + Finite configuration space $\implies$ **Exact** probability values are **not** relevant.
Lack of monotonicity
Lack of monotonicity

\[
q_0 \xrightarrow{W(1)} q_1 \xrightarrow{R(1)} q_2 \xrightarrow{W(2)} q_3 \xrightarrow{R(2)} q_f
\]

⇒ Additional processes can create new deadlocks!
Lack of monotonicity

Additional processes can create new deadlocks!
Probabilistic Cut-off and WQO relation

We study the cut-off for the following $\varphi$ property:

$$\varphi := \mathbb{P}(\diamond \uparrow q_f) = 1$$
Probabilistic Cut-off and WQO relation

We study the cut-off for the following $\varphi$ property:

$$\varphi := \mathbb{P}(\lozenge \uparrow q_f) = 1$$

**Theorem (Admitted)**

$$\mathbb{P}_k(\lozenge \uparrow q_f) = 1$$

if, and only if,

$$\text{Post}^*(I \cap S_{=k}) \subseteq \text{Pre}^*(\uparrow q_f)$$
Probabilistic Cut-off and WQO relation

We study the cut-off for the following $\varphi$ property:

$$\varphi := \mathbb{P}(\diamond \uparrow q_f) = 1$$

Theorem (Admitted)

$$\mathbb{P}_k(\diamond \uparrow q_f) = 1$$

if, and only if,

$$Post^*(I \cap S_{=k}) \subseteq Pre^*(\uparrow q_f)$$

Proof (sketch):
Probabilistic Cut-off and WQO relation

We study the cut-off for the following \( \varphi \) property:

\[
\varphi := \mathbb{P}(\lozenge \uparrow q_f) = 1
\]

**Theorem (Admitted)**

\[
\mathbb{P}_k(\lozenge \uparrow q_f) = 1
\]

if, and only if,

\[
\text{Post}^*(I \cap S_{=k}) \subseteq \text{Pre}^*(\uparrow q_f)
\]

**Proof (sketch):** If \( \mathbb{P}_k(\lozenge \uparrow q_f) = 1 \), then any reachable state \( s \in S_k \) is reached with positive probability so should still reach \( \uparrow q_f \) with probability 1, in particular \( s \in \text{Pre}^*(\uparrow q_f) \). Let’s prove the reverse implication: Assume \( \text{Post}^*(I \cap S_{=k}) \subseteq \text{Pre}^*(\uparrow q_f) \), then:

- For any reachable state \( s \), there exists a path to reach \( \uparrow q_f \) so the probability for this to happen is some positive number \( f(s) > 0 \).
Probabilistic Cut-off and WQO relation

We study the cut-off for the following \( \varphi \) property:

\[
\varphi := \mathbb{P}(\Diamond \uparrow q_f) = 1
\]

**Theorem (Admitted)**

\[
\mathbb{P}_k(\Diamond \uparrow q_f) = 1
\]

if, and only if,

\[
\text{Post}^*(I \cap S_{=k}) \subseteq \text{Pre}^*(\uparrow q_f)
\]

**Proof (sketch):** If \( \mathbb{P}_k(\Diamond \uparrow q_f) = 1 \), then any reachable state \( s \in S_k \) is reached with positive probability so should still reach \( \uparrow q_f \) with probability 1, in particular \( s \in \text{Pre}^*(\uparrow q_f) \). Let’s prove the reverse implication:

Assume \( \text{Post}^*(I \cap S_{=k}) \subseteq \text{Pre}^*(\uparrow q_f) \), then:

- For any reachable state \( s \), there exists a path to reach \( \uparrow q_f \) so the probability for this to happen is some positive number \( f(s) > 0 \).
- At any time point, the probability of eventually reaching \( \uparrow q_f \) is at least \( \max_{s \in S_k} f(s) \) which is **positive** since \( S_k \) is **finite**.
Examples

"Filter" protocol $\mathcal{F}_n$ for $n > 0$. 
Examples

For protocol $\mathcal{F}_n$, 

- networks of size $\geq n$ cover $s_n$ with probability 1,
- networks of size $< n$ cannot cover $s_n$.

No deadlock can ever occur as all processes can always go back to the initial state.
“Filter” protocol $\mathcal{F}_n$ for $n > 0$.

For protocol $\mathcal{F}_n$,

- networks of size $\geq n$ cover $s_n$ with probability 1,
- networks of size $< n$ cannot cover $s_n$.

No deadlock can ever occur as all processes can always go back to the initial state.

$\implies$ **Tight positive cut-off equal to $n$, i.e., linear in the protocol size.**
Existence of a cut-off

**Theorem**
For any register protocol \(\mathcal{P}\) there always exists a cut-off for almost-sure reachability
Existence of a cut-off

**Theorem**
For any register protocol $\mathcal{P}$ **there always exists a cut-off for almost-sure reachability**

⚠️ This result strongly relies on the fact that both $(S, \leq, \rightarrow)$ and $(S, \leq, \rightarrow^{-1})$ are WSTS
Existence of a cut-off

Theorem
For any register protocol $\mathcal{P}$ there always exists a cut-off for almost-sure reachability

⚠️ This result strongly relies on the fact that both $(S, \leq, \rightarrow)$ and $(S, \leq, \rightarrow^{-1})$ are WSTS

The non-atomicity guarantees that when a process takes a transition, all processes in the same state can also take the same transition (with a non-zero probability).

$\implies$ a.k.a. copycat lemma.
Existence: quick sketch (1/2)

- Write $\text{Pre}^*(\uparrow q_f) \subseteq S$ the set of configurations that can reach $q_f$;
- Write $\text{Post}^*(I \cap S_{\geq n}) \subseteq S$ the set of reachable configurations of size $\geq n$;
Existence: quick sketch (1/2)

- Write $\text{Pre}^*([q_f]) \subseteq S$ the set of configurations that can reach $q_f$;
- Write $\text{Post}^*(I \cap S_{\geq n}) \subseteq S$ the set of reachable configurations of size $\geq n$;

Positive cut-off:
$\exists n \text{ Post}^*(I \cap S_{\geq n}) \subseteq \text{Pre}^*([q_f])$

Negative cut-off:
$\forall n \text{ Post}^*(I \cap S_{\leq n}) \not\subseteq \text{Pre}^*([q_f])$
Existence: quick sketch (1/2)

- Write $\text{Pre}^* (\uparrow q_f ) \subseteq S$ the set of configurations that can reach $q_f$;
- Write $\text{Post}^* (I \cap S_{\geq n}) \subseteq S$ the set of reachable configurations of size $\geq n$;
  
  **Positive cut-off:**
  \[ \exists n \text{ Post}^* (I \cap S_{\geq n}) \subseteq \text{Pre}^* (\uparrow q_f ) \]

  **Negative cut-off:**
  \[ \forall n \text{ Post}^* (I \cap S_{\geq n}) \not\subseteq \text{Pre}^* (\uparrow q_f ) \]

- Consider $\leq$ point-wise order over configurations, with state-value support equality.
Existence: quick sketch (1/2)

- Write $\text{Pre}^*(\uparrow q_f) \subseteq S$ the set of configurations that can reach $q_f$;
- Write $\text{Post}^*(I \cap S_{\geq n}) \subseteq S$ the set of reachable configurations of size $\geq n$;

Positive cut-off: $\exists n \; \text{Post}^*(I \cap S_{\geq n}) \subseteq \text{Pre}^*(\uparrow q_f)$

Negative cut-off: $\forall n \; \text{Post}^*(I \cap S_{= n}) \not\subseteq \text{Pre}^*(\uparrow q_f)$

Consider $\leq$ point-wise order over configurations, with state-value support equality.
Existence: quick sketch (1/2)

- Write $\text{Pre}^* (\uparrow q_f) \subseteq S$ the set of configurations that can reach $q_f$;
- Write $\text{Post}^* (I \cap S_{\geq n}) \subseteq S$ the set of reachable configurations of size $\geq n$;

  **Positive** cut-off:
  $\exists n \text{ Post}^* (I \cap S_{\geq n}) \subseteq \text{Pre}^* (\uparrow q_f)$

  **Negative** cut-off:
  $\forall n \text{ Post}^* (I \cap S_{\geq n}) \not\subseteq \text{Pre}^* (\uparrow q_f)$

- Consider $\leq$ point-wise order over configurations, **with state-value support** equality.

  $\begin{array}{cccc}
  0 & q_0 & q_1 & \not\leq \quad 1 & q_0 & q_1 & \leq \quad 1 & q_0 & q_1 & q_1 & \not\leq \\
  1 & q_0 & q_1 & q_1 & \not\leq \\
  \end{array}$

- $(S, \leq)$ is a well-quasi-ordered set;
- $\text{Pre}^*$ and $\text{Post}^* (S_{\geq n})$ are upward-closed for any $n$;
Existence: quick sketch (2/2)

1. $\text{Post}^* (I \cap S_{\geq 1}) = \uparrow \{\theta_1, \ldots, \theta_l\}$ and $\text{Pre}^* = \uparrow \{\eta_1, \ldots, \eta_m\}$. 
Existence: quick sketch (2/2)

1. $\text{Post}^* (I \cap S_{\geq 1}) = \uparrow \{ \theta_1, \ldots, \theta_l \}$ and $\text{Pre}^* = \uparrow \{ \eta_1, \ldots, \eta_m \}$.

2. *Is $\text{Post}^* (I \cap S_{\geq n})$ eventually included in $\text{Pre}^*$?*

    $\implies$ A bit technical...
Existence: quick sketch (2/2)

1. \( \text{Post}^* (I \cap S_{\geq 1}) = \uparrow \{\theta_1, \ldots, \theta_l\} \) and \( \text{Pre}^* = \uparrow \{\eta_1, \ldots, \eta_m\} \).

2. \text{Is Post}^* (I \cap S_{\geq n}) \text{ eventually included in Pre}^* ?

\( \Rightarrow \) A bit technical... A more general results applies on systems that are WSTS in both directions.
Existence: quick sketch (2/2)

1. Post\(^*(I \cap S_{\geq 1}) = \uparrow \{\theta_1, \ldots, \theta_l\}\) and Pre\(^* = \uparrow \{\eta_1, \ldots, \eta_m\}.

2. Is Post\(^*(I \cap S_{\geq n})\) eventually included in Pre\(^*\)?

\[\implies \text{A bit technical... A more general results applies on systems that are WSTS in both directions}\]

\[\ldots\text{intuitively, the goal is to check if elements of Post} \quad (I \cap S_{\geq n})\quad \text{can enter Pre}^*\quad \text{by adding sufficiently many processes in a given state.}\]
Lemma 1. Let $(S, I, \preceq, |.|)$ a parameterised system such that

1. $(S, \preceq, \rightarrow^{-1})$ is a WSTS;

2. For any $s, s' \in S$ with $s \preceq s'$ and for any $k \in [|s|, |s'|]$, there exists $t$ of size $k$ such that $s \preceq t \preceq s'$.

Then for any upward closed set $U$, there exists $N$ such that, either

$$\forall k \geq N, \text{Post}^* (I \cap S = k) \subseteq U$$

either,

$$\forall k \geq N, \text{Post}^* (I \cap S = k) \not\subseteq U$$

Proof. Let $K = \{k \mid \text{Post}^* (I \cap S = k) \not\subseteq U\}$.

- If $K$ is finite, take $N = 1 + \max K$. Then $\forall k \geq N, \text{Post}^* (I \cap S = k) \subseteq U$.

- If $K$ is infinite, for any $k \in K$, let $i_k \in I \cap S_{=k}$ and $x_k \in \text{Post}^* (i_k) \setminus U$. Since $(S, \preceq)$ is a WQO, we can extract an infinite subset $K' \subset K$ such that for all $k_1, k_2 \in K'$ with $k_1 \leq k_2$, $i_{k_1} \preceq i_{k_2}$ but also (by extracting another subsequence), $x_{k_1} \preceq x_{k_2}$. We take $n = \min K$, for any $k \geq N$, since $K'$ is infinite, there exists $k' \in K'$ such that $n \leq k \leq k'$. Since $x_n \preceq x_{k'}$ there exists $x_k$ such that $x_n \preceq x_k \preceq x_{k'}$ with $|x_k| = k$. Since $i_n \rightarrow^* x_n \preceq x_k$ and $(S, \preceq, \rightarrow^{-1})$ is a WQO, there exists $i_k$ such that $i_n \preceq i_k \rightarrow^* x_k$ hence $x_k \in \text{Post}^* (I \cap S = k)$. Moreover, $\overline{U}$ is downward closed so $x_k \not\in U$. 

\[\Box\]