Handling Infinite Branching WSTS

Michael Blondin\textsuperscript{1, 2}, Alain Finkel\textsuperscript{1} & Pierre McKenzie\textsuperscript{1, 2}

\textsuperscript{1}LSV, ENS Cachan
\textsuperscript{2}DIRO, Université de Montréal

March 31, 2014
Well-structured transition systems (WSTS) are known to encompass a large number of infinite state systems.
Well-structured transition systems (WSTS) are known to encompass a large number of infinite state systems. Moreover, multiple decidability results are known on WSTS.
Well-structured transition systems (WSTS) are known to encompass a large number of infinite state systems. Moreover, multiple decidability results are known on WSTS. However, most results and techniques known suppose finite branching.
Well-structured transition systems (WSTS) are known to encompass a large number of infinite state systems. Moreover, multiple decidability results are known on WSTS. However, most results and techniques known suppose finite branching.

We propose a tool, the WSTS completion, based on work of Finkel and Goubault-Larrecq, to handle infinitely branching WSTS.
Ordered transition system

\[ S = (X, \rightarrow, \leq) \] where

- \( X \) set,
- \( \rightarrow \subseteq X \times X \),
- \( \leq \) quasi-ordering \( X \).
Ordered transition system

\[ S = (X, \rightarrow, \leq) \text{ where} \]

- \( X \) set: recursively enumerable,
- \( \rightarrow \subseteq X \times X \): decidable,
- \( \leq \) quasi-ordering \( X \): decidable.
A WSTS is an ordered transition system \((X, \rightarrow, \leq)\) with

- well-quasi-ordering: \(\forall x_0, x_1, \ldots \exists i < j\) s.t. \(x_i \leq x_j\),
- monotony:

\[
\forall x \quad x' \rightarrow y' \quad \exists x' \rightarrow y
\]
Well-ordered transition system (WSTS)

A WSTS is an ordered transition system \((X, \rightarrow, \leq)\) with

- well-quasi-ordering: \(\forall x_0, x_1, \ldots \ \exists i < j \text{ s.t. } x_i \leq x_j\),

- transitive monotony:

\[
\forall x \quad \rightarrow \quad y
\]

\[
\land
\]

\[
\land
\]

\[
x' \quad \rightarrow \quad y'
\]
Branching

A WSTS \((X, \rightarrow, \leq)\) is finitely branching if \(\text{Post}(x)\) is finite for every \(x \in X\).
Branching

A WSTS \((X, \rightarrow, \leq)\) is finitely branching if Post\((x)\) is finite for every \(x \in X\).

Some infinitely branching WSTS

- Inserting FIFO automata (Cécé, Finkel, Iyer 1996),
Branching

A WSTS \((X, \rightarrow, \leq)\) is finitely branching if \(\text{Post}(x)\) is finite for every \(x \in X\).

Some infinitely branching WSTS

- Inserting FIFO automata (Cécé, Finkel, Iyer 1996),
- Inserting automata (Bouyer, Markey, Ouaknine, Schnoebelen, Worrell 2012),
### Branching

A WSTS \((X, \rightarrow, \preceq)\) is finitely branching if \(\text{Post}(x)\) is finite for every \(x \in X\).

### Some infinitely branching WSTS

- Inserting FIFO automata (Cécé, Finkel, Iyer 1996),
- Inserting automata (Bouyer, Markey, Ouaknine, Schnoebelen, Worrell 2012),
- \(\omega\)-Petri nets (Geeraerts, Heussner, Praveen & Raskin 2013),
Branching

A WSTS \((X, \rightarrow, \leq)\) is finitely branching if \(\text{Post}(x)\) is finite for every \(x \in X\).

Some infinitely branching WSTS

- Inserting FIFO automata (Cécé, Finkel, Iyer 1996),
- Inserting automata (Bouyer, Markey, Ouaknine, Schnoebelen, Worrell 2012),
- \(\omega\)-Petri nets (Geeraerts, Heussner, Praveen & Raskin 2013),
- Essentially finite WSTS (Abdulla, Cerans, Jonsson & Tsay 2000),
A WSTS \((X, \rightarrow, \leq)\) is finitely branching if \(\text{Post}(x)\) is finite for every \(x \in X\).

Some infinitely branching WSTS

- Inserting FIFO automata (Cécé, Finkel, Iyer 1996),
- Inserting automata (Bouyer, Markey, Ouaknine, Schnoebelen, Worrell 2012),
- \(\omega\)-Petri nets (Geeraerts, Heussner, Praveen & Raskin 2013),
- Essentially finite WSTS (Abdulla, Cerans, Jonsson & Tsay 2000),
- Do you know other ones?
Problematic

Some decidability results for WSTS based on finite reachability trees; impossible for infinite branching.

Some rely on upward closed sets; what about downward closed, in particular with infinite branching?
Some decidability results for WSTS based on finite reachability trees; impossible for infinite branching.

Some rely on upward closed sets; what about downward closed, in particular with infinite branching?

A tool

Develop from the WSTS completion introduced by Finkel & Goubault-Larrecq 2009.
$I \subseteq X$ is an *ideal* if it is

- downward closed: $I = \downarrow I$,
- directed: $a, b \in I \implies \exists c \in I$ s.t. $a \leq c$ and $b \leq c$. 

---

**Ideals**

$I \subseteq X$ is an *ideal* if it is

- downward closed: $I = \downarrow I$,
- directed: $a, b \in I \implies \exists c \in I$ s.t. $a \leq c$ and $b \leq c$. 

Theorem (Finkel & Goubault-Larrecq 2009; GL 2014)

\[ D \text{ downward closed} \implies D = \bigcup_{\text{finite}} \text{Ideals} \]
Theorem (Finkel & Goubault-Larrecq 2009; GL 2014)

\[ D \text{ downward closed} \implies D = \bigcup_{\text{finite}} \text{Ideals} \]

Theorem (Finkel & Goubault-Larrecq 2009; GL 2014; BFM 2014)

Every downward closed subset decomposes canonically as the union of its maximal ideals.
Completion (Finkel & Goubault-Larrecq 2009; BFM 2014)

The *completion* of $S = (X, \rightarrow_S, \leq)$ is $\hat{S} = (\hat{X}, \rightarrow_{\hat{S}}, \subseteq)$ such that
Completion (Finkel & Goubault-Larrecq 2009; BFM 2014)

The *completion* of $S = (X, \rightarrow_S, \leq)$ is $\hat{S} = (\hat{X}, \rightarrow_{\hat{S}}, \subseteq)$ such that

$\hat{X} = \text{Ideals}(X)$,
Completion (Finkel & Goubault-Larrecq 2009; BFM 2014)

The *completion* of $S = (X, \rightarrow_S, \leq)$ is $\hat{S} = (\hat{X}, \rightarrow_{\hat{S}}, \subseteq)$ such that

- $\hat{X} = \text{Ideals}(X)$,
- $I \rightarrow_{\hat{S}} J$ if $\downarrow \text{Post}(I) = \ldots \cup J \cup \ldots$.

canonical
Theorem (Finkel & Goubault-Larrecq 2009; BFM 2014)

Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, then

- $\hat{S}$ is finitely branching.
Theorem (Finkel & Goubault-Larrecq 2009; BFM 2014)

Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, then

- $\hat{S}$ is finitely branching,
- $\hat{S}$ has (strong) monotony,
Theorem (Finkel & Goubault-Larrecq 2009; BFM 2014)

Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, then

- $\hat{S}$ is finitely branching,
- $\hat{S}$ has (strong) monotony,
- $\hat{S}$ is not always a WSTS
Theorem (Finkel & Goubault-Larrecq 2009; BFM 2014)

Let \( S = (X, \rightarrow_S, \leq) \) be a WSTS, then

- \( \hat{S} \) is finitely branching,
- \( \hat{S} \) has (strong) monotony,
- \( \hat{S} \) is \textit{not always} a WSTS
Relating executions of $S$ and $\hat{S}$

Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, then

- if $x \xrightarrow[k]{} y$,
Relating executions of $S$ and $\hat{S}$

Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, then

- if $x \stackrel{k}{\rightarrow}_S y$, then for every ideal $I \supseteq \downarrow x$
Relating executions of $S$ and $\hat{S}$

Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, then

- if $x \xrightarrow{k}_S y$, then for every ideal $I \supseteq \downarrow x$ there exists an ideal $J \supseteq \downarrow y$.  


Relating executions of $S$ and $\hat{S}$

Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, then

- if $x \xrightarrow{k}_S y$, then for every ideal $I \supseteq \downarrow x$ there exists an ideal $J \supseteq \downarrow y$ such that $I \xrightarrow{k}_{\hat{S}} J$, 


Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, then

- if $x \xrightarrow{k} S y$, then for every ideal $I \supseteq \downarrow x$ there exists an ideal $J \supseteq \downarrow y$ such that $I \xrightarrow{k} \hat{S} J,$

- if $I \xrightarrow{k} \hat{S} J,$
Relating executions of $S$ and $\hat{S}$

Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, then

- if $x \xrightarrow{k}_S y$, then for every ideal $I \supseteq \downarrow x$ there exists an ideal $J \supseteq \downarrow y$ such that $I \xrightarrow{k}_{\hat{S}} J$,

- if $I \xrightarrow{k}_{\hat{S}} J$, then for every $y \in J$
Relating executions of $S$ and $\hat{S}$

Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, then

- if $x \xrightarrow{k}_S y$, then for every ideal $I \supseteq \downarrow x$ there exists an ideal $J \supseteq \downarrow y$ such that $I \xrightarrow{k}_{\hat{S}} J$,

- if $I \xrightarrow{k}_{\hat{S}} J$, then for every $y \in J$ there exists $x \in I$
### Relating executions of $S$ and $\hat{S}$

Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, then

- if $x \xrightarrow{S} y$, then for every ideal $I \supseteq \downarrow x$ there exists an ideal $J \supseteq \downarrow y$ such that $I \xrightarrow{k} \hat{S} J$,

- if $I \xrightarrow{k} \hat{S} J$, then for every $y \in J$ there exists $x \in I$ such that $x \xrightarrow{*} S y' \geq y$. 
Relating executions of $S$ and $\tilde{S}$

Let $S = (X, \rightarrow_S, \leq)$ be a WSTS with transitive monotony, then

- if $x \xrightarrow{k}_S y$, then for every ideal $I \supseteq \downarrow x$ there exists an ideal $J \supseteq \downarrow y$ such that $I \xrightarrow{k}_{\tilde{S}} J$,

- if $I \xrightarrow{k}_{\tilde{S}} J$, then for every $y \in J$ there exists $x \in I$ such that $x \xrightarrow{\geq k}_S y' \geq y$. 
## Termination

\textit{Input:} \((X, \to, \leq)\) a WSTS, \(x_0 \in X\).

\textit{Question:} \(\not\exists x_0 \to x_1 \to x_2 \to \ldots\)?
Termination

Input: \((X, \rightarrow, \leq)\) a WSTS, \(x_0 \in X\).

Question: \(\not\exists x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \ldots\) ?

Proposition (Dufourd, Jančar & Schnoebelen 1999)

Termination is undecidable for infinitely branching WSTS.
Strong termination

**Input:** \((X, \rightarrow, \leq)\) a WSTS, \(x_0 \in X\).

**Question:** \(\exists k\) bounding length of executions from \(x_0\)?
Strong termination

**Input:** \((X, \rightarrow, \leq)\) a WSTS, \(x_0 \in X\).

**Question:** \(\exists k\) bounding length of executions from \(x_0\)?

**Remark**

Strong termination and termination are the same in finitely branching WSTS.
Theorem (Blondin, Finkel & McKenzie 2014)

Strong termination is decidable for WSTS with transitive monotony and such that $\hat{S}$ is a post-effective WSTS.
Theorem (Blondin, Finkel & McKenzie 2014)

Strong termination is decidable for WSTS with transitive monotony and such that $\hat{S}$ is a post-effective WSTS.
Theorem (Blondin, Finkel & McKenzie 2014)

Strong termination is decidable for WSTS with transitive monotony and such that \( \hat{S} \) is a post-effective WSTS.

Proof

Executions bounded in \( S \) iff bounded in \( \hat{S} \).
Theorem (Blondin, Finkel & McKenzie 2014)

Strong termination is decidable for WSTS with transitive monotony and such that \( \hat{S} \) is a post-effective WSTS.

Proof

Executions bounded in \( S \) iff bounded in \( \hat{S} \). Since \( \hat{S} \) finitely branching, we can decide termination in \( \hat{S} \) by Finkel & Schnoebelen 2001.
**Coverability**

**Input:** \((X, \rightarrow, \leq)\) a WSTS, \(x_0, x \in X\).

**Question:** \(x_0 \overset{*}{\rightarrow} x' \geq x\)?
Coverability

**Input:** \((X, \to, \leq)\) a WSTS, \(x_0, x \in X\).

**Question:** \(x_0 \in \uparrow \text{Pre}^*(\uparrow x)\)?
## Coverability

**Input:** \((X, \rightarrow, \leq)\) a WSTS, \(x_0, x \in X\).

**Question:** \(x_0 \in \uparrow \text{Pre}^*(\uparrow x)\)?

## Backward method (Abdulla, Cerans, Jonsson & Tsay 2000)

Compute \(Y_0, \ldots, Y_n\) converging to \(\uparrow \text{Pre}^*(\uparrow x)\) and verify if \(x_0 \in Y_n\).
Coverability

*Input:* $(X, \rightarrow, \leq)$ a WSTS, $x_0, x \in X$.

*Question:* $x_0 \in \uparrow \text{Pre}^*(\uparrow x)$?

Backward method (Abdulla, Cerans, Jonsson & Tsay 2000)

Compute $Y_0, \ldots, Y_n$ converging to $\uparrow \text{Pre}^*(\uparrow x)$ and verify if $x_0 \in Y_n$. 
<table>
<thead>
<tr>
<th>Coverability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $(X, \rightarrow, \leq)$ a WSTS, $x_0, x \in X$.</td>
</tr>
<tr>
<td><strong>Question:</strong> $x \in \downarrow \text{Post}^*(x_0)$?</td>
</tr>
</tbody>
</table>
Coverability

**Input:** \((X, \rightarrow, \leq)\) a WSTS, \(x_0, x \in X\).

**Question:** \(x \in \downarrow \text{Post}^*(x_0)\)?

**Forward method**

Coverability:

- Enumerate executions \(\downarrow x_0 \xrightarrow{\ast}^\hat{S} I\),
- Accept if \(x \in I\).
### Coverability

**Input:** $(X, →, ≤)$ a WSTS, $x_0, x ∈ X$.

**Question:** $x ∈ ↓\text{Post}^*(x_0)$?

### Forward method

**Coverability:**
- Enumerate executions $↓x_0 \xrightarrow{∗} S I$.
- Accept if $x ∈ I$.

**Non coverability:**
- Enumerate $D ⊆ X$ downward closed, $x_0 ∈ D$ and $↓\text{Post}_S(D) ⊆ D$.
- Reject if $x ∉ D$. 
Coverability

Input: \((X, \to, \leq)\) a WSTS, \(x_0, x \in X\).

Question: \(x \in \downarrow \text{Post}^*(x_0)\)?

**Forward method**

Coverability:
- Enumerate executions \(\downarrow x_0 \xrightarrow{\ast} S I\),
- Accept if \(x \in I\).

Non coverability:
- Enumerate \(D \subseteq X\) downward closed, \(x_0 \in D\) and \(\downarrow \text{Post}_S(D) \subseteq D\),
- Reject if \(x \notin D\).  
  **Witness:** \(D = \downarrow \text{Post}_S^*(x_0)\)
### Coverability

**Input:** \((X, \rightarrow, \leq)\) a WSTS, \(x_0, x \in X\).

**Question:** \(x \in \downarrow Post^*(x_0)\)?

### Forward method

**Coverability:**
- Enumerate executions \(\downarrow x_0 \xrightarrow{*} \hat{S} I\).
- Accept if \(x \in I\).

**Non coverability:**
- Enumerate \(D = I_1 \cup \ldots \cup I_k\)
- Reject if \(x \notin D\).
## Coverability

**Input:** $(X, \rightarrow, \leq)$ a WSTS, $x_0, x \in X$.

**Question:** $x \in \downarrow \text{Post}^*(x_0)$?

### Forward method

**Coverability:**
- Enumerate executions $\downarrow x_0 \overset{*}{\rightarrow}^\downarrow I$.
- Accept if $x \in I$.

**Non coverability:**
- Enumerate $D \subseteq X$ downward closed.
- Reject if $x \notin D$. 
Coverability

Input: \((X, \rightarrow, \leq)\) a WSTS, \(x_0, x \in X\).

Question: \(x \in \downarrow \text{Post}^*(x_0)\)?

**Forward method**

**Coverability:**
- Enumerate executions \(\downarrow x_0 \xrightarrow[*]{} I\).
- Accept if \(x \in I\).

**Non coverability:**
- Enumerate \(D \subseteq X\) downward closed, \(x_0 \in D\).
- Reject if \(x \notin D\).
### Coverability

**Input:** $(X, \to, \leq)$ a WSTS, $x_0, x \in X$.

**Question:** $x \in \downarrow \text{Post}^*(x_0)$?

### Forward method

**Coverability:**
- Enumerate executions $\downarrow x_0 \xrightarrow{\star} \hat{S} I$,
- Accept if $x \in I$.

**Non coverability:**
- Enumerate $D \subseteq X$ downward closed, $\downarrow x_0 \subseteq I_1 \cup \ldots \cup I_k$
- Reject if $x \notin D$. 

Input: $(X, \to, \leq)$ a WSTS, $x_0, x \in X$.  
Question: $x \in \downarrow \text{Post}^*(x_0)$?
### Coverability

**Input:** \((X, \rightarrow, \leq)\) a WSTS, \(x_0, x \in X\).

**Question:** \(x \in \downarrow \text{Post}^* (x_0)\)?

### Forward method

**Coverability:**
- Enumerate executions \(\downarrow x_0 \xrightarrow{*} S I\),
- Accept if \(x \in I\).

**Non coverability:**
- Enumerate \(D \subseteq X\) downward closed, \(\exists j\) s.t. \(\downarrow x_0 \subseteq I_j\)
- Reject if \(x \notin D\).
Coverability

Input: \((X, \rightarrow, \leq)\) a WSTS, \(x_0, x \in X\).

Question: \(x \in \downarrow \text{Post}^*(x_0)\)?

Forward method

Coverability:

- Enumerate executions \(\downarrow x_0 \xrightarrow{\ast} \hat{S} I\),
- Accept if \(x \in I\).

Non coverability:

- Enumerate \(D \subseteq X\) downward closed, \(x_0 \in D\)
- Reject if \(x \notin D\).
Coverability

*Input:* \((X, \rightarrow, \leq)\) a WSTS, \(x_0, x \in X\).

*Question:* \(x \in \downarrow \text{Post}^*(x_0)?\)

**Forward method**

**Coverability:**
- Enumerate executions \(\downarrow x_0 \xrightarrow{\ast} I\),
- Accept if \(x \in I\).

**Non coverability:**
- Enumerate \(D \subseteq X\) downward closed, \(x_0 \in D\) and \(\downarrow \text{Post}_S(D) \subseteq D\),
- Reject if \(x \notin D\).
Open questions

- What other applications has the completion?
Open questions

- What other applications has the completion?
- Boundness and strong control-state maintainability also decidable for infinitely branching WSTS. Other problems decidable?
Open questions

- What other applications has the completion?
- Boundness and strong control-state maintainability also decidable for infinitely branching WSTS. Other problems decidable?
- Algorithms working on the completion more efficient for what WSTS/problems?
Thank you!