WBTS: the new class of WSTS without WQO

Alain Finkel

LSV, ENS Paris-Saclay (ex ENS Cachan)

SynCoP + PV
22-23 April 2017 Blasenhus, Uppsala, Sweden

- Based on joint works with Michael Blondin & Pierre McKenzie, to appear in LMCS.

```
one_task(int k) {
 // some work...
}
main(int P) {
 for i := 1 to P step 1
  fork(one_task(i))
}
```

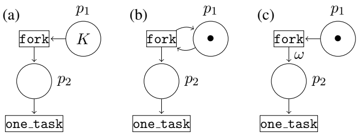


Figure 1. An example of a parametric system with three possible models

From parameterized systems

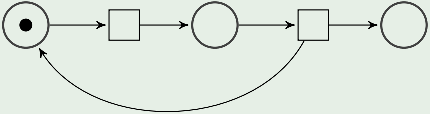
- Parameterized protocols: mutual exclusion, broadcast protocols
- Parametric concurrent systems with dynamic thread creation (ex: Geeraerts & al. FI, 2015)
- Parameterized systems in BIP (ex: Konnov & al. Concur'2016)

To infinite-state models

- Counter (abstraction) machines with reset-transfer-affine extensions
- ω-Petri nets (infinite branching)
- WSTS

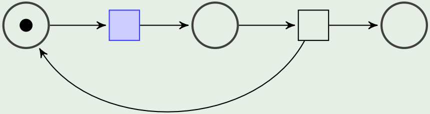
Well Structured Transition Systems (WSTS) encompass a large number of infinite state systems (PN and reset-transfer-affine-ω extensions, lossy fifo systems, broadcast protocols, CFG, graph rewriting, depth bounded processes, fragments of the π-calculus,....)

Example of WSTS: Petri nets



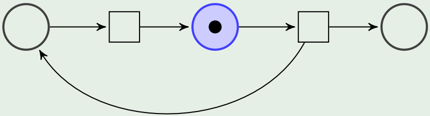
Well Structured Transition Systems (WSTS) encompass a large number of infinite state systems (PN and reset-transfer-affine-ω extensions, lossy fifo systems, broadcast protocols, CFG, graph rewriting, depth bounded processes, fragments of the π-calculus,....)

Example of WSTS: Petri nets



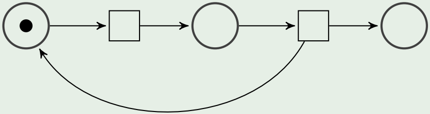
Well Structured Transition Systems (WSTS) encompass a large number of infinite state systems (PN and reset-transfer-affine-ω extensions, lossy fifo systems, broadcast protocols, CFG, graph rewriting, depth bounded processes, fragments of the π-calculus,....)

Example of WSTS: Petri nets



Multiple decidability results are known for (finitely branching) WSTS.

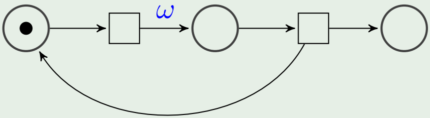
Example of WSTS: Petri nets



Post(⊙ ○ ○) = ○ ⊙ ○

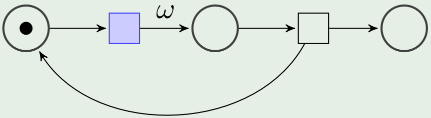
And also for (infinitely branching) WSTS such as systems with infinitely many initial states and parametric systems

Example of WSTS: ω–Petri nets (Geeraerts, Heußner, Praveen & Raskin PN'13, FU'15)



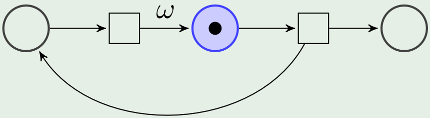
And also for (infinitely branching) WSTS such as systems with infinitely many initial states and parametric systems

Example of WSTS: ω–Petri nets (Geeraerts, Heußner, Praveen & Raskin PN'13, FU'15)



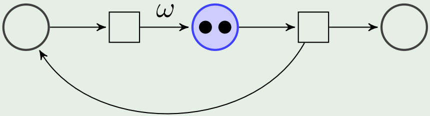
And also for (infinitely branching) WSTS such as systems with infinitely many initial states and parametric systems

Example of WSTS: ω–Petri nets (Geeraerts, Heußner, Praveen & Raskin PN'13, FU'15)



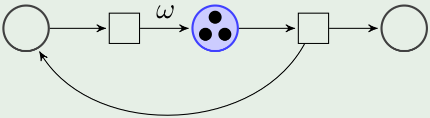
And also for (infinitely branching) WSTS such as systems with infinitely many initial states and parametric systems

Example of WSTS: ω–Petri nets (Geeraerts, Heußner, Praveen & Raskin PN'13, FU'15)



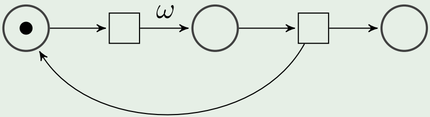
And also for (infinitely branching) WSTS such as systems with infinitely many initial states and parametric systems

Example of WSTS: ω–Petri nets (Geeraerts, Heußner, Praveen & Raskin PN'13, FU'15)



And also for (infinitely branching) WSTS such as systems with infinitely many initial states and parametric systems

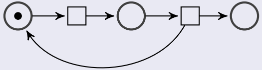
Example of WSTS: ω–Petri nets (Geeraerts, Heußner, Praveen & Raskin PN'13, FU'15)



Well structured transition system (F, ICALP'87)

$S = (X, \rightarrow, \leq)$ where

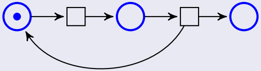
- X set,
- $\rightarrow \subseteq X \times X$,
- monotony,
- well-quasi-ordered.



Well structured transition system (F, ICALP'87)

$S = (X, \rightarrow, \leq)$ where

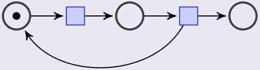
- $\mathbb{N}^3$,
- $\rightarrow \subseteq X \times X$,
- monotony,
- well-quasi-ordered.



Well structured transition system (F, ICALP'87)

$S = (X, \rightarrow, \leq)$ where

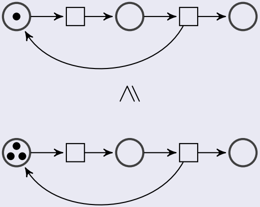
- X set,
- $\rightarrow \subseteq \mathbb{N}^3 \times \mathbb{N}^3$,
- monotony,
- well-quasi-ordered.



Well structured transition system (F, ICALP'87)

$S = (X, \rightarrow, \leq)$ where

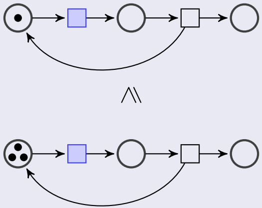
- X set,
- $\rightarrow \subseteq X \times X$,
- monotony,
- well-quasi-ordered.



Well structured transition system (F, ICALP'87)

$S = (X, \to, \leq)$ where

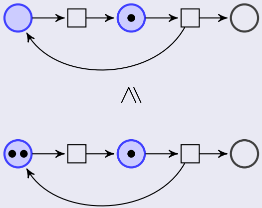
- X set,
- $\to \subseteq X \times X$,
- monotony,
- well-quasi-ordered.



Well structured transition system (F, ICALP'87)

$S = (X, \rightarrow, \leq)$ where

- X set,
- $\rightarrow \subseteq X \times X$,
- monotony,
- well-quasi-ordered.



Well structured transition system (F, ICALP'87)

$S = (X, \rightarrow, \leq)$ where

- X set,
- $\rightarrow \subseteq X \times X$,
- monotony,
- well-quasi-ordered.

$$
\begin{array}{cccc}
\forall\, x & \rightarrow & y & \\
\leq & & \leq & \\
x' & \xrightarrow{*} & y' & \exists
\end{array}
$$

Well structured transition system (F, ICALP'87)

$S = (X, \rightarrow, \leq)$ where

- X set,
- $\rightarrow \subseteq X \times X$,
- transitive monotony,
- well-quasi-ordered.

$$\begin{array}{cccc} \forall\; x & \rightarrow & y & \\ \leq & & \leq & \\ x' & \boxed{\xrightarrow{+} \;\; y'} & & \exists \end{array}$$

Well structured transition system (F, ICALP'87)

$S = (X, \rightarrow, \leq)$ where

- X set,
- $\rightarrow \subseteq X \times X$,
- strong monotony,
- well-quasi-ordered.

$$\begin{array}{cccc} \forall\; x & \rightarrow & y & \\ \wedge\!\!\backslash & & \wedge\!\!\backslash & \\ x' & \rightarrow & y' & \exists \end{array}$$

Well structured transition system (F, ICALP'87)

$S = (X, \rightarrow, \leq)$ where

- X set,
- $\rightarrow \subseteq X \times X$,
- monotony,
- well-quasi-ordered:
  $\forall x_0, x_1, \ldots \; \exists i < j$ s.t. $x_i \leq x_j$.

WSTS Everywhere! (F, Schnoebelen LATIN'98, TCS'01)

- $T(w)$ = length of a longest computation starting from $w \in \Sigma^*$.
- $T(w) \in \mathbb{N}_\omega$.
- $w \leq_T w'$ if $T(w) \leq T(w')$.
- $\leq_T$ is a wqo on Σ^*.

WSTS Everywhere! (F, Schnoebelen LATIN'98, TCS'01)

- $T(w)$ = length of a longest computation starting from $w \in \Sigma^*$.
- $T(w) \in \mathbb{N}_\omega$.
- $w \leq_T w'$ if $T(w) \leq T(w')$.
- $\leq_T$ is a wqo on Σ^*.

Theorem

Turing machines are WSTS with strict and strong monotony wrt $\leq_T$.

WSTS Everywhere!

- $\leq_T$ is not decidable.
- Hence TM are non-effective WSTS.
- This also proves that there is no (non-trivial) decidability result for non-effective WSTS (not surprising !).

Effective WSTS

A WSTS $S = (X, \rightarrow, \leq)$, given as a tuple $(M_X, M_{\rightarrow}, M_{\leq})$ of Turing machines, is *effective* if:

(1) M_X decides X

(2) $M_{\rightarrow}$ decides $\rightarrow$.

(3) $M_{\leq}$ decides $\leq$.

(Effective) WSTS Everywhere!

- $S = (\mathbb{N}^k, \leq)$.
  - Petri nets: WSTS with strict and strong monotony.
  - Positive Affine nets, Reset/Transfer Petri nets: WSTS with strong (but not strict) monotony.

(Effective) WSTS Everywhere!

- $S = (\mathbb{N}^k, \leq)$.
  - Petri nets: WSTS with strict and strong monotony.
  - Positive Affine nets, Reset/Transfer Petri nets: WSTS with strong (but not strict) monotony.

- $S = (Q \times \Sigma^{*k}, = \times \sqsubseteq^k)$.
  - LCS: WSTS with non-strict monotony.

WSTS still verywhere!

- Data nets: $S = (Q \times \mathbb{N}^k)^*$
  - Lazic, Newcomb, Ouaknine, Roscoe, Worrell (PN'07)
  - Hofman, Lasota, Lazić, Leroux, Schmitz, Totzke (FOSSACS'16).
  - Lasota (PN'16)
- ν-Petri nets: $S = (Q \times \mathbb{N}^k)^{\oplus}$.
  - Rosa-Velardo, de Frutos-Escrig (PN'07)
  - Lazić and Schmitz (LICS'16).
- Pi-calculus: Depth-Bounded Processes (trees).
  - Wies, Zufferey, Henzinger (FOSSACS'10, VMCAI'12).
- Timed Petri nets: $Regions = ((Q \times \mathbb{N}^k)^{\oplus})^*$
  - Bonnet, F, Haddad, Rosa-Velardo (FOSSACS'10)
  - Haddad, Schmitz, Schnoebelen (LICS'12).
- Process algebra (BPP,...).

Objective

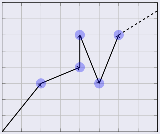
We want to study the usual reachability problems, e.g.,

- Reachability...but it is undecidable for general WSTS :((

Objective

We want to study the usual reachability problems, e.g.,

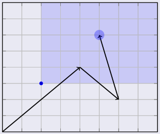
- Reachability...but it is undecidable for general WSTS :((
- Termination



Objective

We want to study the usual reachability problems, e.g.,

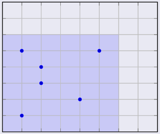
- Reachability...but it is undecidable for general WSTS :((
- Termination
- Coverability (the most used property)



Objective

We want to study the usual reachability problems, e.g.,

- Reachability...but it is undecidable for general WSTS :((
- Termination
- Coverability (the most used property)
- Boundedness



Objective

We want to study the usual reachability problems, e.g.,

- Reachability...but it is undecidable for general WSTS :((
- Termination
- Coverability (the most used property)
- Boundedness
- And other properties like eventuality, simulation by finite automaton...

- ICALP'87 (F)
  - WSTS definitions
  - decidability of termination
  - decidability of boundedness
  - computation of the coverability set hence decidability of coverability (under stronger hyp.)

- ICALP'87 (F)
  - WSTS definitions
  - decidability of termination
  - decidability of boundedness
  - computation of the coverability set hence decidability of coverability (under stronger hyp.)
- LICS'96 (Abdulla, Cerans, Jonsson, Tsay)
  - decidability of coverability with a backward algorithm
  - decidability of simulation with finite-state systems
  - undecidability of repeated control-state (for LCS).

- ICALP'87 (F)
  - WSTS definitions
  - decidability of termination
  - decidability of boundedness
  - computation of the coverability set hence decidability of coverability (under stronger hyp.)
- LICS'96 (Abdulla, Cerans, Jonsson, Tsay)
  - decidability of coverability with a backward algorithm
  - decidability of simulation with finite-state systems
  - undecidability of repeated control-state (for LCS).
- LICS'98 (Emerson, Namjoshi), LICS'99 (Esparza, F, Mayr)
  - broadcast protocols are WSTS
  - model checking of WSTS (with procedures)
- WSTS everywhere, TCS'01 (F, Schnoebelen)

- FSTTCS'04 (Geeraerts, Raskin and Van Begin):
  - The first forward coverability algorithm for WSTS (with ADL).
- STACS'09, ICALP'09 (F, Goubault-Larrecq), ICALP'14 (Blondin, F, McKenzie)
  - ADL is not an hypothesis.
  - Ideal completion of any WSTS
  - Computation of the clover for flattable WSTS
  - ω^2-WSTS are completable and robust....

- FSTTCS'04 (Geeraerts, Raskin and Van Begin):
  - The first forward coverability algorithm for WSTS (with ADL).
- STACS'09, ICALP'09 (F, Goubault-Larrecq), ICALP'14 (Blondin, F, McKenzie)
  - ADL is not an hypothesis.
  - Ideal completion of any WSTS
  - Computation of the clover for flattable WSTS
  - ω^2-WSTS are completable and robust....
- 2015-2016: Use of ideals decomposition in:
  - RP'15: The Ideal View on Rackoff's Coverability Technique (Lazić, Schmitz)
  - LICS'15: Demystifying Reachability in Vector Addition Systems (Leroux, Schmitz).
  - FOSSACS'16: Coverability Trees for Petri Nets with Unordered Data (Schmitz and a lot of authors...)
  - LICS'16: ν-Petri nets (Lazić, Schmitz).
  - ...

The survey/story of coverability for WSTS

Year	Authors	Mathematical hyp.	Effectivity hyp.	back/forwar
1978	Arnold & Latteux	reset VAS	reset VAS are effective	backward
1987	F.	(very) WSTS+strong+strict+ω^2-wqo+...	effective very WSTS	forward
1996	Abdulla & CJT	strong monotony	$\text{Pre}_S(\uparrow x)$ comp.	backward
1998	F. Schnoebelen	monotony	$\uparrow \text{Pre}_S(\uparrow x)$ comp.	backward
2004	Geeraerts & RV	strong monotony, ADL	effective ADL	forward
2006	Geeraerts & RV	monotony, ADL	effective ADL	forward
2009	F. & Goubault-Larrecq	strong monotony, weak ADL, flattable	effective WADL	forward
2009	F. & Goubault-Larrecq	strong monotony, flattable	ideally effective	forward
2014	Blondin & FM	monotony,	ideally effective	forward
2016	Blondin & FM	monotony, no wqo but FAC	ideally effective	forward

Coverability

For monotone transition systems, y is coverable from x if

- $\exists x' \mid x \xrightarrow{*} x' \geq y$ (this is the definition !) iff

Coverability

For monotone transition systems, y is coverable from x if

- $\exists x' \mid x \xrightarrow{*} x' \geq y$ (this is the definition !) iff
- $x \in \mathrm{Pre}^*(\uparrow y)$ (this could be the definition !) iff

Coverability

For monotone transition systems, y is coverable from x if

- $\exists x' \mid x \xrightarrow{*} x' \geq y$ (this is the definition !) iff
- $x \in \mathrm{Pre}^*(\uparrow y)$ (this could be the definition !) iff
- $y \in \downarrow \mathrm{Post}^*(x)$ (this could be the definition !).

Remark

- $\mathrm{Pre}^*(\uparrow y) = \uparrow \mathrm{Pre}^*(\uparrow y)$

Coverability

For monotone transition systems, y is coverable from x if

- $\exists x' \mid x \xrightarrow{*} x' \geq y$ (this is the definition !) iff
- $x \in \mathrm{Pre}^*(\uparrow y)$ (this could be the definition !) iff
- $y \in \downarrow \mathrm{Post}^*(x)$ (this could be the definition !).

Remark

- $\mathrm{Pre}^*(\uparrow y) = \uparrow \mathrm{Pre}^*(\uparrow y)$
- $\downarrow \mathrm{Post}^*(x) = \downarrow \mathrm{Post}^*(\downarrow x)$.

Consequence

- Compute $\uparrow \mathrm{Pre}^*(\uparrow y)$ or $\downarrow \mathrm{Post}^*(x)$.
- Compute $\uparrow$ or $\downarrow$ over-approximations invariants.

The backward coverability algorithm (based on ↑-sets)

- 1978: coverability for reset VAS is decidable (Arnold and Latteux published in French in CALCOLO'78). Their algorithm is an instance of the backward algorithm (LICS'96).
- 1993: decidability of coverability for LCS (Abdulla, Cerans, Jonsson, Tsay, LICS'93)
- 1996: decidability of coverability for strong WSTS assuming Pre($\uparrow x$) is computable (Abdulla, Cerans, Jonsson, Tsay, LICS'96)
- 1998: decidability of coverability for WSTS assuming $\uparrow$Pre($\uparrow x$) is computable (F., Schnoebelen LATIN'98)

Remarks on the backward coverability algorithm

- It computes $\text{Pre}^*(\uparrow y)$ that is more than solving coverability.
- It is often but not always computable, ex: depth-bounded processes (Wies, Zufferey, Henzinger, FOSSACS'10)
- Backward algorithms are often less efficient than forward algorithms.

Coverability: a conceptual algorithm (still based on ↑-sets)

Execute two procedures in parallel, one looking for a coverability certificate and one looking for a non coverability certificate.

Coverability: a conceptual algorithm (still based on ↑-sets)

Execute two procedures in parallel, one looking for a coverability certificate and one looking for a non coverability certificate.

- Coverability is semi-decidable:
  - if $\exists x' \geq y$, $x \xrightarrow{*} x'$, one finally will find x'.

Coverability: a conceptual algorithm (still based on ↑-sets)

Execute two procedures in parallel, one looking for a coverability certificate and one looking for a non coverability certificate.

- Coverability is semi-decidable:
  - if $\exists x' \geq y,\ x \xrightarrow{*} x'$, one finally will find x'.
- Non-coverability is also semi-decidable:
  - $\neg(\exists x' \geq y,\ x \xrightarrow{*} x')$ iff $x \notin Pre^*(\uparrow y) = \uparrow J_m$ for some m.

Coverability: a conceptual algorithm (still based on ↑-sets)

Execute two procedures in parallel, one looking for a coverability certificate and one looking for a non coverability certificate.

- Coverability is semi-decidable:
  - if $\exists x' \geq y$, $x \xrightarrow{*} x'$, one finally will find x'.
- Non-coverability is also semi-decidable:
  - $\neg(\exists x' \geq y, x \xrightarrow{*} x')$ iff $x \notin Pre^*(\uparrow y) = \uparrow J_m$ for some m.
  - One enumerates all the finite sets $J \subseteq X$ such that $y \in \uparrow J$ and $Pre(\uparrow J) \subseteq \uparrow J$ and $x \notin \uparrow J$, hence $\uparrow J_m \subseteq Pre^*(\uparrow J) = \uparrow J$. Enumeration of upward closed sets by a finite set of minimal elements is a consequence of $(X, \leq)$ is WQO.

Coverability: a conceptual algorithm (still based on ↑-sets)

Execute two procedures in parallel, one looking for a coverability certificate and one looking for a non coverability certificate.

- Coverability is semi-decidable:
  - if $\exists x' \geq y,\ x \xrightarrow{*} x'$, one finally will find x'.
- Non-coverability is also semi-decidable:
  - $\neg(\exists x' \geq y,\ x \xrightarrow{*} x')$ iff $x \notin Pre^*(\uparrow y) = \uparrow J_m$ for some m.
  - One enumerates all the finite sets $J \subseteq X$ such that $y \in \uparrow J$ and $Pre(\uparrow J) \subseteq \uparrow J$ and $x \notin \uparrow J$, hence $\uparrow J_m \subseteq Pre^*(\uparrow J) = \uparrow J$.
    Enumeration of upward closed sets by a finite set of minimal elements is a consequence of $(X, \leq)$ is WQO.
  - One finally will find such an invariant J. May be we find a large J_p s.t. $\uparrow J_m = Pre^*(\uparrow y) \subsetneq \uparrow J_p$ but $x \notin \uparrow J_p \implies x \notin \uparrow J_m = Pre^*(\uparrow y)$.

A very interesting characterization of WQO (Finite Basis Property):

Theorem (Higman'52, Nash-Williams' 63-64, Fraïsse'86,...)

$$(X, \leq) \text{ WQO} \iff \text{for all } U = \uparrow U \subseteq X \text{ we have: } U = \bigcup_{\text{finite}} \uparrow x$$

The downward approach for coverability

- Initially presented by Geeraerts, Raskin, and Van Begin (FSTTCS'04) for strongly monotone WSTS with Adequate Domain of Limits (ADL).

The downward approach for coverability

- Initially presented by Geeraerts, Raskin, and Van Begin (FSTTCS'04) for strongly monotone WSTS with Adequate Domain of Limits (ADL).
- Simplified and extended with Goubault-Larrecq (STACS'09): ADL is not an hypothesis, it always exists.

The downward approach for coverability

- Initially presented by Geeraerts, Raskin, and Van Begin (FSTTCS'04) for strongly monotone WSTS with Adequate Domain of Limits (ADL).
- Simplified and extended with Goubault-Larrecq (STACS'09): ADL is not an hypothesis, it always exists.
- Still simplified and extended with Blondin, McKenzie (ICALP'14): ideal completion for infinitely branching.

The downward approach for coverability

- Initially presented by Geeraerts, Raskin, and Van Begin (FSTTCS'04) for strongly monotone WSTS with Adequate Domain of Limits (ADL).
- Simplified and extended with Goubault-Larrecq (STACS'09): ADL is not an hypothesis, it always exists.
- Still simplified and extended with Blondin, McKenzie (ICALP'14): ideal completion for infinitely branching.
- Still simplified and extended with Blondin, McKenzie: WQO is not necessary. (arxiv, august 2016, to appear in LMCS'2017).

In order to decide whether y is coverable from x, Procedure 1 iteratively computes

$$\mathrm{Post}(x), \mathrm{Post}(\mathrm{Post}(x)), \mathrm{Post}^3(x), \ \ldots$$

until it finds y in $\downarrow \mathrm{Post}^n(x)$.

procedure 1: searches for a coverability certificate of y from x

$D \leftarrow x$;
while $y \notin \downarrow D$ **do**
    $D \leftarrow D \cup \mathrm{Post}(D)$
return *true*

y is not coverable from x iff $y \notin \downarrow \text{Post}^*(x)$.

Let $(D_i)_i$ be an enumeration of dc sets, hence $\downarrow \text{Post}^*(x) = D_m$, for some m.

procedure 2: enumerates dcs to find non coverability certificate of y from x

$i \leftarrow 0$;
while $\neg(\downarrow \text{Post}(D_i) \subseteq D_i$ and $x \in D_i$ and $y \notin D_i)$ **do**
  $i \leftarrow i + 1$
return *false*

procedure 1: searches for a coverability certificate of y from x

$D \leftarrow x$;
while $y \notin \downarrow D$ **do**
    $D \leftarrow D \cup \mathrm{Post}(D)$
return *true*

procedure 2: enumerates dcs to find non coverability certificate

$i \leftarrow 0$;
while $\neg(\downarrow \mathrm{Post}(D_i) \subseteq D_i$ and $x \in D_i$ and $y \notin D_i)$ **do**
    $i \leftarrow i + 1$
return *false*

Effective hypotheses

- dcs are recursive.
- Union of dcs is computable
- $\downarrow \mathrm{Post}(D)$ is computable.
- Inclusion between dcs is decidable.

Theorem

Let $S = (X, \rightarrow, \leq)$ be a monotone transition system + there exists an enumeration of downward closed sets of X, and let $x, y \in X$.

1. y is coverable from x iff Procedure 1 terminates.
2. y is not coverable from x iff Procedure 2 terminates.

This theorem does not provide an algorithm.

Remark

WSTS, hence WQO implies possible enumeration of downward closed sets (by minimal elements of upward closed sets) but the converse is false: $(\mathbb{Z}, \leq)$ is not WQO but one may enumerate the D_i as follows: $D_i = \downarrow x_i$ for $x_i \in \mathbb{Z}$ or $D_i = \mathbb{Z}$.

Question

How to enumerate downward closed sets ?

Answer

By enumerating ideals ! (Erdös & Tarski)

Definition

- Now $(X, \leq)$ is a qo (shortly written X or $\leq$)

Definition

- Now $(X, \leq)$ is a qo (shortly written X or $\leq$)
- $A \subseteq X$ is an antichain if all couples of A are uncomparable.

Definition

- Now $(X, \leq)$ is a qo (shortly written X or $\leq$)
- $A \subseteq X$ is an antichain if all couples of A are uncomparable.
- X is Finite AntiChain (FAC) if all antichains in X are finite.

Definition

- Now $(X, \leq)$ is a qo (shortly written X or $\leq$)
- $A \subseteq X$ is an antichain if all couples of A are uncomparable.
- X is Finite AntiChain (FAC) if all antichains in X are finite.
- X is Well Founded (WF) if all strictly decreasing sequences in X are finite.

Definition

- Now $(X, \leq)$ is a qo (shortly written X or $\leq$)
- $A \subseteq X$ is an antichain if all couples of A are uncomparable.
- X is Finite AntiChain (FAC) if all antichains in X are finite.
- X is Well Founded (WF) if all strictly decreasing sequences in X are finite.
- X is WQO if X is FAC & X is WF (can be a Theorem).

Definition

- Now $(X, \leq)$ is a qo (shortly written X or $\leq$)
- $A \subseteq X$ is an antichain if all couples of A are uncomparable.
- X is Finite AntiChain (FAC) if all antichains in X are finite.
- X is Well Founded (WF) if all strictly decreasing sequences in X are finite.
- X is WQO if X is FAC & X is WF (can be a Theorem).

- Every WQO is FAC

Definition

- Now $(X, \leq)$ is a qo (shortly written X or $\leq$)
- $A \subseteq X$ is an antichain if all couples of A are uncomparable.
- X is Finite AntiChain (FAC) if all antichains in X are finite.
- X is Well Founded (WF) if all strictly decreasing sequences in X are finite.
- X is WQO if X is FAC & X is WF (can be a Theorem).

- Every WQO is FAC
- The converse is false since $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ are FAC but not WF: $13/7, \pi, 3, 2, 1, 0, -1/2, -1, -2, -3, -41, -78695/12, \ldots$

Definition

- Now $(X, \leq)$ is a qo (shortly written X or $\leq$)
- $A \subseteq X$ is an antichain if all couples of A are uncomparable.
- X is Finite AntiChain (FAC) if all antichains in X are finite.
- X is Well Founded (WF) if all strictly decreasing sequences in X are finite.
- X is WQO if X is FAC & X is WF (can be a Theorem).

- Every WQO is FAC
- The converse is false since $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ are FAC but not WF: $13/7, \pi, 3, 2, 1, 0, -1/2, -1, -2, -3, -41, -78695/12, \ldots$

Definition

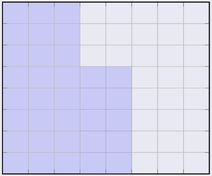
- Now $(X, \leq)$ is a qo (shortly written X or $\leq$)
- $A \subseteq X$ is an antichain if all couples of A are uncomparable.
- X is Finite AntiChain (FAC) if all antichains in X are finite.
- X is Well Founded (WF) if all strictly decreasing sequences in X are finite.
- X is WQO if X is FAC & X is WF (can be a Theorem).

- Every WQO is FAC
- The converse is false since $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ are FAC but not WF: $13/7, \pi, 3, 2, 1, 0, -1/2, -1, -2, -3, -41, -78695/12, \ldots$
- $\mathbb{Z}^2$ contains infinite antichains, $A = \{(n, -n) \mid n \in \mathbb{N}\}$, hence the cartesian product of two FAC's is not necessary a FAC.

Ideals

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

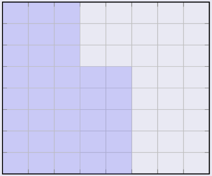
- downward closed: $I = \downarrow I$,



Ideals

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

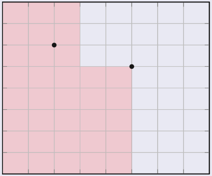
- downward closed: $I = \downarrow I$,
- directed: $a, b \in I \implies \exists c \in I$ s.t. $a \leq c$ and $b \leq c$.



Ideals

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

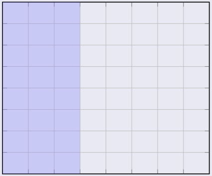
- downward closed: $I = \downarrow I$,
- directed: $a, b \in I \implies \exists c \in I$ s.t. $a \leq c$ and $b \leq c$.



Ideals

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

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



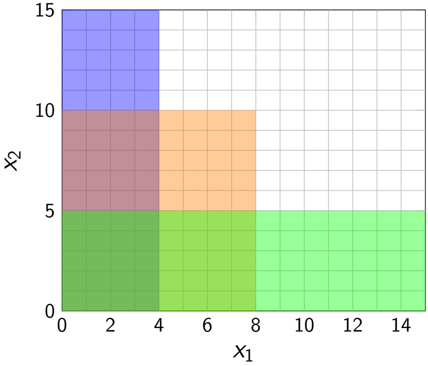
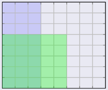


Figure: Decomposition of $X = \{(x_1, x_2) \in \mathbb{N}^2 : (x_1 \leq 4) \vee (x_1 \leq 8 \wedge x_2 \leq 10) \vee (x_2 \leq 5)\}$ into finitely many ideals. The three ideals $\downarrow 4 \times \mathbb{N}, \downarrow 8 \times \downarrow 10$ and $\mathbb{N} \times \downarrow 5$ appear respectively in blue, orange and green.

A very interesting but unknown theorem in the verification community like was the Higman theorem before WSTS'87 - *front* lossy channel systems'89 - lossy channel systems'93.

Theorem (Erdös & Tarski'43, Bonnet'75, Fraïsse'86,...)

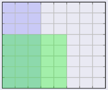
$$(X,\leq) \text{ FAC} \iff \text{for all } D = \downarrow D \subseteq X \text{ we have: } D = \bigcup_{\text{finite}} \text{Ideals}$$



A very interesting but unknown theorem in the verification community like was the Higman theorem before WSTS'87 - *front* lossy channel systems'89 - lossy channel systems'93.

Theorem (Erdös & Tarski'43, Bonnet'75, Fraïsse'86,...)

$$(X, \leq) \text{ FAC} \iff \text{for all } D = \downarrow D \subseteq X \text{ we have: } D = \bigcup_{\text{finite}} \text{Ideals}$$



Corollary

Every downward closed set decomposes canonically as the union of its $\subseteq$-maximal ideals.

Annals of Mathematics
Vol. 44, No. 2, April, 1943

ON FAMILIES OF MUTUALLY EXCLUSIVE SETS

By P. Erdös and A. Tarski

(Received August 11, 1942)

In this paper we shall be concerned with a certain particular problem from the general theory of sets, namely with the problem of the existence of families of mutually exclusive sets with a maximal power. It will turn out—in a rather unexpected way—that the solution of these problems essentially involves the notion of the so-called "inaccessible numbers." In this connection we shall make some general remarks regarding inaccessible numbers in the last section of our paper.

Theorem

Let $D = \downarrow D \subseteq X$ and X a WQO. Then $D = I_1 \cup I_2 \cup \cdots \cup I_m$ for some $I_1, I_2, \ldots, I_m \in Ideals(X)$.

Assume that a bad D (bad = dc set that does not admit a finite (may be empty) decomposition in ideals) exists.

Theorem

Let $D = \downarrow D \subseteq X$ and X a WQO. Then $D = I_1 \cup I_2 \cup \cdots \cup I_m$ for some $I_1, I_2, \ldots, I_m \in Ideals(X)$.

Assume that a bad D (bad = dc set that does not admit a finite (may be empty) decomposition in ideals) exists.

$\exists D$ bad and minimal for inclusion among bad subsets (strictly decreasing subsequences of dc subsets are finite in a WQO).

- $D \neq \emptyset$ since $\emptyset$ is equal to an empty union ($\emptyset$ is not an ideal).
- $D \neq \{d\}$ since $\{d\}$ is an ideal.

Theorem

Let $D = \downarrow D \subseteq X$ and X a WQO. Then $D = I_1 \cup I_2 \cup \cdots \cup I_m$ for some $I_1, I_2, \ldots, I_m \in Ideals(X)$.

Assume that a bad D (bad = dc set that does not admit a finite (may be empty) decomposition in ideals) exists.

$\exists D$ bad and minimal for inclusion among bad subsets (strictly decreasing subsequences of dc subsets are finite in a WQO).

- $D \neq \emptyset$ since $\emptyset$ is equal to an empty union ($\emptyset$ is not an ideal).
- $D \neq \{d\}$ since $\{d\}$ is an ideal.
- Let $x_1 \neq x_2 \in D$.

Since $D \setminus \uparrow x_1$ and $D \setminus \uparrow x_2$ are dc and strictly included in D, they are not bad (by minimality of D).

Theorem

Let $D = \downarrow D \subseteq X$ and X a WQO. Then $D = I_1 \cup I_2 \cup \cdots \cup I_m$ for some $I_1, I_2, \ldots, I_m \in Ideals(X)$.

Assume that a bad D (bad = dc set that does not admit a finite (may be empty) decomposition in ideals) exists.

$\exists D$ bad and minimal for inclusion among bad subsets (strictly decreasing subsequences of dc subsets are finite in a WQO).

- $D \neq \emptyset$ since $\emptyset$ is equal to an empty union ($\emptyset$ is not an ideal).
- $D \neq \{d\}$ since $\{d\}$ is an ideal.
- Let $x_1 \neq x_2 \in D$.

Since $D \setminus \uparrow x_1$ and $D \setminus \uparrow x_2$ are dc and strictly included in D, they are not bad (by minimality of D).
Thus, $D \setminus \uparrow x_1 = \bigcup_{j=1}^{n} I_j$ and $D \setminus \uparrow x_2 = \bigcup_{j=n+1}^{m} I_j$ for some ideals $I_1, I_2, \ldots, I_m \subseteq X$.

Hence

$$D' = (D \setminus \uparrow x_1) \cup (D \setminus \uparrow x_2) = \bigcup_{j=1}^{m} I_j$$

Hence

$$D' = (D \setminus \uparrow x_1) \cup (D \setminus \uparrow x_2) = \bigcup_{j=1}^{m} I_j$$

We have:

$$D' = D \setminus (\uparrow x_1 \cap \uparrow x_2)$$

Hence

$$D' = (D \setminus \uparrow x_1) \cup (D \setminus \uparrow x_2) = \bigcup_{j=1}^{m} I_j$$

We have:

$$D' = D \setminus (\uparrow x_1 \cap \uparrow x_2)$$

As $D' \neq D$,

Hence

$$D' = (D \setminus \uparrow x_1) \cup (D \setminus \uparrow x_2) = \bigcup_{j=1}^{m} I_j$$

We have:

$$D' = D \setminus (\uparrow x_1 \cap \uparrow x_2)$$

As $D' \neq D$, therefore, $D \cap (\uparrow x_1 \cap \uparrow x_2) \neq \emptyset$

Thus:

Hence

$$D' = (D \setminus \uparrow x_1) \cup (D \setminus \uparrow x_2) = \bigcup_{j=1}^{m} I_j$$

We have:

$$D' = D \setminus (\uparrow x_1 \cap \uparrow x_2)$$

As $D' \neq D$, therefore, $D \cap (\uparrow x_1 \cap \uparrow x_2) \neq \emptyset$

Thus:

$\exists d \in D \cap (\uparrow x_1 \cap \uparrow x_2)$ s.t. $x_1 \leq d$ and $x_2 \leq d$.

Hence

$$D' = (D \setminus \uparrow x_1) \cup (D \setminus \uparrow x_2) = \bigcup_{j=1}^{m} I_j$$

We have:

$$D' = D \setminus (\uparrow x_1 \cap \uparrow x_2)$$

As $D' \neq D$, therefore, $D \cap (\uparrow x_1 \cap \uparrow x_2) \neq \emptyset$

Thus:

$\exists d \in D \cap (\uparrow x_1 \cap \uparrow x_2)$ s.t. $x_1 \leq d$ and $x_2 \leq d$.

Hence D is directed and therefore D is an ideal, contradicting our assumption. Thus, D is equal to a finite union of ideals. □

Definition (new)

A *Well Behaved Transition System (WBTS)* is a monotone transition system $S = (X, \rightarrow, \leq)$ such that $(X, \leq)$ is FAC.

WSTS and WBTS

- WSTS are WBTS but the converse is false: $\mathbb{Z}$-VASS are WBTS but are not WSTS.
- Weighted VASS are WBTS for $\leq$ (but are not WSTS).
- Multi-weighted VASS are WBTS for $\leq_{lex}$ (but are not WSTS).

Corollary (new)

Coverability is decidable for any ideally effective class of WBTS.

Remark

The backward coverability procedure does not terminate on $\mathbb{Z}$-VASS, weighted VASS and multi-weighted VASS.

The survey/story of coverability for WSTS

Year	**Authors**	**Mathematical hyp.**	**Effectivity hyp.**	**back/forward**
1978	Arnold & Latteux	reset VAS	YES	backward
1987	F.	very WSTS (strong+strict, ω^2-wqo,...)	effective very WSTS	forward
1996	Abdulla & CJT	strong monotony	$\mathrm{Pre}_S(\uparrow x)$ comp.	backward
1998	F. Schnoebelen	monotony	$\uparrow \mathrm{Pre}_S(\uparrow x)$ comp.	backward
2004	Geeraerts & RV	strong monotony, ADL	effective ADL	forward
2006	Geeraerts & RV	monotony, ADL	effective ADL	forward
2009	F. & Goubault-Larrecq	strong monotony, weak ADL, flattable	effective WADL	forward
2009	F. & Goubault-Larrecq	strong monotony, flattable	ideally effective	forward
2014	Blondin & FM	monotony,	ideally effective	forward
2016	Blondin & FM	monotony, no wqo but FAC	ideally effective	forward
2017	Trivial	no monotony, wqo (Minsky machines)	ideally effective	Undec.
2017	New question	monotony, no wqo but WF	ideally effective	Undec ?

Further work

Further work

- Explore more in details WBTS.

Further work

- Explore more in details WBTS.
- Find applications of WBTS.

Further work

- Explore more in details WBTS.
- Find applications of WBTS.
- Computing efficiently with ideals (no brut force enumeration).

Further work

- Explore more in details WBTS.
- Find applications of WBTS.
- Computing efficiently with ideals (no brut force enumeration).
- Design Karp and Miller tree algorithm for ω^2-WSTS.

Further work

- Explore more in details WBTS.
- Find applications of WBTS.
- Computing efficiently with ideals (no brut force enumeration).
- Design Karp and Miller tree algorithm for ω^2-WSTS.

Interships available at the ENS Paris-Saclay

Further work

- Explore more in details WBTS.
- Find applications of WBTS.
- Computing efficiently with ideals (no brut force enumeration).
- Design Karp and Miller tree algorithm for ω^2-WSTS.

Interships available at the ENS Paris-Saclay

- Different topics: theoretical and/or applied subjects.

Further work

- Explore more in details WBTS.
- Find applications of WBTS.
- Computing efficiently with ideals (no brut force enumeration).
- Design Karp and Miller tree algorithm for ω^2-WSTS.

Interships available at the ENS Paris-Saclay

- Different topics: theoretical and/or applied subjects.
- Developping the WSTS theory or a prototype for finding bugs in web services and choreographies.

Further work

- Explore more in details WBTS.
- Find applications of WBTS.
- Computing efficiently with ideals (no brut force enumeration).
- Design Karp and Miller tree algorithm for ω^2-WSTS.

Interships available at the ENS Paris-Saclay

- Different topics: theoretical and/or applied subjects.
- Developping the WSTS theory or a prototype for finding bugs in web services and choreographies.
- Different levels: Bachelor, Master, PhD, post-PhD.

Thank you!