Verification of Directed Acyclic Ad-Hoc Networks

Othmane Rezine
Verification of Directed Acyclic Ad-Hoc Networks

Parosh Abdulla  Mohammed Faouzi Atig  Othmane Rezine
Ad-Hoc Networks

Directed Acyclic
Directed Acyclic

Ad-Hoc Networks
Ad-Hoc Networks

- Model
- Transition System
- Reachability

Directed Acyclic
Directed Acyclic Ad-Hoc Networks

- Model
- Transition System
- Reachability

Ad-Hoc Networks

Directed Acyclic
Ad-Hoc Networks

- Model
- Transition System
- Reachability
Ad-Hoc Networks

- Model
- Transition System
- Reachability

- Wireless nodes: laptop
Ad-Hoc Networks

- Model
- Transition System
- Reachability

- Wireless nodes: laptop
- Radio Range
Ad-Hoc Networks

- Model
- Transition System
- Reachability

- Wireless nodes: laptop
- Radio Range
- Links / Topology
Ad-Hoc Networks

- Model
- Transition System
- Reachability

- Wireless nodes: laptop
- Radio Range
- Links / Topology
- Distributed management of the network
Ad-Hoc Networks

- Wireless nodes: laptop
- Radio Range
- Links / Topology
- Distributed management of the network

- Model
- Transition System
- Reachability
Ad-Hoc Networks

- Wireless nodes: laptop
- Radio Range
- Links / Topology
- Distributed management of the network

Applications
- Home area networks
- No telecom infrastructure
Ad-Hoc Networks

- Wireless nodes: laptop
- Radio Range
- Links / Topology
- Distributed management of the network
Ad-Hoc Networks

- Wireless nodes: laptop
- Radio Range
- Links / Topology
- Distributed management of the network
Ad-Hoc Networks

- Wireless nodes: laptops, sensors
- Radio Range
- Links / Topology
- Distributed management of the network

Model
Transition System
Reachability
Ad-Hoc Networks

Applications

- Monitoring of
  - Seismic activity
  - Heat
  - Pollution

- Wireless nodes: laptops
  - Sensors

- Radio Range
- Links / Topology
- Distributed management of the network

Model
- Transition System
- Reachability
Ad-Hoc Networks

- Model
- Transition System
- Reachability
Ad-Hoc Networks

- Model
- Transition System
- Reachability

Directed Acyclic
Parameterized verification of ad-hoc networks

G. Delzanno, A. Sangnier, G. Zavattaro

CONCUR’10
Ad-Hoc Networks

- Model
- Transition System
- Reachability

G. Delzanno, A. Sangnier, G. Zavattaro
Parameterized verification of ad-hoc networks

CONCUR’10

Model
Ad-Hoc Networks

- Model
- Transition System
- Reachability

G. Delzanno, A. Sangnier, G. Zavattaro
Parameterized verification of ad-hoc networks
CONCUR’10

Model

Node: Process

Directed Acyclic Model
Transition System
Reachability
Ad-Hoc Networks

G. Delzanno, A. Sangnier, G. Zavattaro
Parameterized verification of ad-hoc networks
CONCUR’10

Model

Node:
Process

- Model
- Transition System
- Reachability

Directed Acyclic Graph:

1: Path to 2, 3, 4, 5
Ad-Hoc Networks

- Model
- Transition System
- Reachability

G. Delzanno, A. Sangnier, G. Zavattaro
Parameterized verification of ad-hoc networks

CONCUR’10

Model

Node:
- Process

Transitions:

Directed Acyclic Graph

Node: i

Transitions:

Model

Transition System

Reachability
Ad-Hoc Networks

G. Delzanno, A. Sangnier, G. Zavattaro
Parameterized verification of ad-hoc networks
CONCUR’10

Model

Node: Process
Transitions:
- Local

Directed Acyclic Graph:
- Init
- Transitions: i → 1, 1 → 2, 1 → 3, 2 → 4, 3 → 5, 4 → 2, 5 → 3

Model
- Model
- Transition System
- Reachability
Ad-Hoc Networks

- Model
- Transition System
- Reachability

G. Delzanno, A. Sangnier, G. Zavattaro
Parameterized verification of ad-hoc networks
CONCUR’10

Model

Node: Process

Transitions:
- Local
- Broadcast

Directed Acyclic

\( \text{b(}\text{hello}\text{)} \)

Model

Transition System

Reachability
Ad-Hoc Networks

G. Delzanno, A. Sangnier, G. Zavattaro
Parameterized verification of ad-hoc networks

Model

- Model
- Transition System
- Reachability

Node: Process

Transitions:
- Local
- Broadcast
- Receive

Model

Example Diagram

Node: Process

Transitions: Local, Broadcast, Receive

Model:

- Model
- Transition System
- Reachability

G. Delzanno, A. Sangnier, G. Zavattaro
Parameterized verification of ad-hoc networks

CONCUR’10
Ad-Hoc Networks

G. Delzanno, A. Sangnier, G. Zavattaro
Parameterized verification of ad-hoc networks
CONCUR’10

Model

Node: Process

Transitions:
- Local
- Broadcast
- Receive

Directed Acyclic
\[ r(\text{hello}) \quad b(\text{ack}) \]
\[ b(\text{hello}) \quad r(\text{ack}) \]

Model

Transition System

Reachability
Ad-Hoc Networks

- Model
- Transition System
- Reachability
Ad-Hoc Networks

- Model
- Transition System
- Reachability
Ad-Hoc Networks

- Model
- Transition System
- Reachability
Ad-Hoc Networks

- Model
- Transition System
- Reachability

[Diagram of nodes 1, 3, and 4 connected with arrows to represent relationships and transitions.]
Ad-Hoc Networks

- Model
- Transition System
- Reachability

Model

Node: Process

Node 1

Node 1

Node i

Node 3

Directed Acyclic Model
Ad-Hoc Networks

Model

Node:
- Process

Topology:
- Symmetric graph

- Model
- Transition System
- Reachability
Ad-Hoc Networks

Model

- Node: Process
- Topology: Symmetric graph
- Configuration: Graph, state mapping

Model
- Transition System
- Reachability
Ad-Hoc Networks

- Model
- Transition System
- Reachability

\[ \langle \Gamma, \rightarrow \rangle \]
Ad-Hoc Networks

- Model
- Transition System
- Reachability

\[ \langle \Gamma, \rightarrow \rangle \]
Set of configurations
Transition Relation
Trace
Model
Transition System
Reachability

\[ \langle \Gamma, \rightarrow \rangle \]
Ad-Hoc Networks

- Model
- Transition System
- Reachability

- Set of configurations
- Transition Relation
- Trace

\[ \langle \Gamma, \rightarrow \rangle \]
Ad-Hoc Networks

- Model
- Transition System
- Reachability

Set of configurations
- Transition Relation
- Trace

Any Symmetric Graph
Ad-Hoc Networks

- Model
- Transition System
- Reachability

Set of configurations
- Transition Relation
- Trace

- Any Symmetric Graph
- Unbounded # of nodes
Ad-Hoc Networks

Set of configurations
Transition Relation
Trace

Process

\[ \langle \Gamma, \rightarrow \rangle \]

Model
Transition System
Reachability

Process

\begin{array}{c}
\text{init} \quad 1 \\
\rightarrow \quad r(\text{hello}) \quad 2 \\
\rightarrow \quad b(\text{ack}) \quad 4 \\
\rightarrow \quad b(\text{hello}) \quad 3 \\
\rightarrow \quad r(\text{ack}) \quad 5 \\
\end{array}
Ad-Hoc Networks

Set of configurations

Transition Relation

Trace

Process

Configuration

Model

Transition System

Reachability
Ad-Hoc Networks

- Model
- Transition System
- Reachability

- Set of configurations
- Transition Relation
- Trace

Configuration

Process

- local
- Set of configurations
- Transition Relation
- Trace

Diagram:

- init
- r(hello)
- b(ack)
- b(hello)
- r(ack)
- local
Directed Acyclic Graph

Set of configurations
Transition Relation
Trace

Model
Transition System
Reachability

Configuration
Transition System
Reachability

Process

Configuration

local

local

r(hello)
b(ack)

b(hello)
r(ack)

init
Ad-Hoc Networks

Set of configurations
Transition Relation
Trace

Process
- \( r(\text{hello}) \)
- \( b(\text{ack}) \)
- init
- \( b(\text{hello}) \)
- \( r(\text{ack}) \)

Configuration
- local
- Model
- Transition System
- Reachability
Directed Acyclic Graph: \( < \Gamma, \rightarrow > \)

- **Ad-Hoc Networks**
- **Set of configurations**
- **Transition Relation**
- **Trace**

**Process**
- \( r(\text{hello}) \) from 1 to 2
- \( b(\text{ack}) \) from 2 to 4
- \( \text{init} \) from 1 to 2
- \( b(\text{hello}) \) from 3 to 4
- \( r(\text{ack}) \) from 4 to 5

**Configuration**
- \( \text{local} \) configuration
- \( \text{local} \) configuration
Ad-Hoc Networks

\[ \langle \Gamma, \rightarrow \rangle \]

- Model
- Transition System
- Reachability
- Set of configurations
- Transition Relation
- Trace

### Process

1. **Init**
   - r(\textit{hello})
   - b(\textit{ack})

2. **2**
   - \textit{hello}

3. **3**
   - b(\textit{hello})
   - r(\textit{ack})

4. **4**

5. **5**

### Configuration

- Local

- \textit{local}

- \textit{local}
Ad-Hoc Networks

\[
\langle \Gamma, \rightarrow \rangle
\]

- Model
- Transition System
- Reachability
- Set of configurations
- Transition Relation
- Trace

**Process**

1. init
2. \( r(\text{hello}) \) → 3
3. \( b(\text{hello}) \) → 5
4. \( b(\text{ack}) \) → 4

**Configuration**

- local
- broadcast
- receive
- selective broadcast
Ad-Hoc Networks

- Directed Acyclic Graph
- Set of configurations
- Transition Relation
- Trace

Process

- local
- broadcast
- receive

Configuration

- local
- selective broadcast

Model
Transition System
Reachability

<i>Γ, →</i>
Ad-Hoc Networks

- Model
- Transition System
- Reachability

- Set of configurations
- Transition Relation
- Trace

Process:
- local
- broadcast
- receive

Configuration:
- local
- selective broadcast

Model:
- broadcaster

Transition System:
- Transition Relation
- Trace

Set of configurations:
- Configuration
- broadcaster

Trace:
- local
- selective broadcast

Direct

Directed Acyclic Graph
Directed Acyclic Graph

Set of configurations

Transition Relation

Trace

Ad-Hoc Networks

\(< \Gamma, \rightarrow \rangle \)

Process

- local
- broadcast
- receive

Configuration

- local
- selective broadcast
- broadcaster

\[ b(\text{hello}) \rightarrow \]

Transition System

Reachability

Model
Directed Acyclic Graph

Ad-Hoc Networks

$\langle \Gamma, \rightarrow \rangle$

- Set of configurations
- Transition Relation
- Trace
- Model
- Transition System
- Reachability

Process

- local
- broadcast
- receive

Configuration

- local
- broadcast
- receive
- selective broadcast
Ad-Hoc Networks

\[ < \Gamma, \rightarrow > \]

- Model
- Transition System
- Reachability

- Set of configurations
- Transition Relation
- Trace

Process

- local
- broadcast
- receive

Configuration

- local
- selective broadcast
- broadcaster
Directed Acyclic Graph (DAG) Model for Ad-Hoc Networks

- Model
- Transition System
- Reachability
- Set of configurations
- Transition Relation
- Trace

**Process**

- local
- broadcast
- receive

**Configuration**

- local
- selective broadcast
- broadcaster

**Diagram**

- Process: `i` (init) → `r(Hello)` → `b(ack)` → `2` → `4` → `3` → `b(Hello)` → `r(ack)` → `5`
- Configuration: `1` (receiver) → `1` (receiver) → `i` (broadcaster) → `3` (selective broadcast)
Ad-Hoc Networks

- Directed Acyclic
- Set of configurations
- Transition Relation
- Trace

Process

- local
- broadcast
- receive

Configuration

- Model
- Transition System
- Reachability

- selective broadcast
- broadcaster

- Model
- Transition System
- Reachability

< Γ, → >

Process

1. **init**
   - **i** → **1**
   - **b**(hello) → **3**
   - **r**(ack) → **5**

Configuration

1. **receiver**
   - **2** → **4**
   - **3 → 5**

2. **broadcaster**
   - **2** → **1**
   - **3** → **1**

- local
- receive
Ad-Hoc Networks

\[ < \Gamma, \rightarrow > \]

- Directed Acyclic
- Set of configurations
- Transition Relation
- Trace

**Process**
- local
- broadcast
- receive

**Configuration**
- Model
- Transition System
- Reachability

**Transition System**
- Model
- Transition System
- Reachability

**Set of configurations**
- local
- broadcast
- receive

**Reachability**
- local
- selective broadcast
- broadcast

**Trace**
Directed Acyclic Graph

Ad-Hoc Networks

Set of configurations
Transition Relation
Trace

Model
Transition System
Reachability

< \Gamma, \rightarrow >

Process

Configuration

local
broadcast
receive

local
selective broadcast

Model
Transition System
Reachability
Ad-Hoc Networks

- Model
- Transition System
- Reachability

Set of configurations
Transition Relation
Trace

Conf

Diagram showing a set of configurations linked by transition relations.
Directed Acyclic

Ad-Hoc Networks

Set of configurations
Transition Relation
Trace

Model
Transition System
Reachability

< Γ, → >

Conf

Conf’

i

1

i

i

i

i

1

*
Ad-Hoc Networks

- Model
- Transition System
- Reachability

Set of configurations
Transition Relation
Trace

Set of configurations
Transition Relation
Trace

Conf
Conf’
Conf”
Control State Reachability (COVER)
Ad-Hoc Networks

- Model
- Transition System
- Reachability

Control State Reachability (COVER)
Ad-Hoc Networks

Control State Reachability (COVER)

Given
Control State Reachability (COVER)

Given Process $P$

- Model
- Transition System
- Reachability

Diagram:

- Node 1: init
- Node 2: $r(\text{hello})$
- Node 3: $b(\text{hello})$
- Node 4: $b(\text{ack})$
- Node 5: $r(\text{ack})$
Control State Reachability (COVER)

Given

- Process P
- Control State q

```
Given r(hello) b(ack)
```

```
b(hello)  r(ack)
```

```
init i 1 2 4
```

```
b(hello)  r(ack)
```

```
3  q
```

```
Process
```

```
Model
Transition System
Reachability
```

Ad-Hoc Networks
Ad-Hoc Networks

Control State Reachability (COVER)

Given

- Process P
- Control State q

Process

- Model
- Transition System
- Reachability

Given

- Process P
- Control State q
Ad-Hoc Networks

Control State Reachability (COVER)

Given
- Process \( P \)
- Control State \( q \)

- Model
- Transition System
- Reachability
Control State Reachability (COVER)

Undecidable
Ad-Hoc Networks

Directed Acyclic
Ad-Hoc Networks

Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability
Wireless Sensor Networks
Directed Acyclic

Wireless Sensor Networks

Phases:

- Motivation
- Reachability
- Bounded Depth Reachability

Graph:
- $sens_1$
- $sens_2$
- sink
- $sens_3$
Directed Acyclic

Wireless Sensor Networks

Phases:

• Sink $\rightarrow$ Sensors:
  ‣ Data request
  ‣ Software updates

• Sensors $\rightarrow$ Sink:
  Data collection.

Motivation
• Reachability
• Bounded Depth Reachability
Directed Acyclic

Wireless Sensor Networks

Phases:

- Sink → Sensors:
  - Data request
  - Software updates

- Sensors → Sink:
  - Data collection.

- Motivation
- Reachability
- Bounded Depth Reachability
Directed Acyclic

Wireless Sensor Networks

Phases:

- Sink $\rightarrow$ Sensors:
  - Data request
  - Software updates

- Sensors $\rightarrow$ Sink:
  Data collection.

- Motivation
- Reachability
- Bounded Depth Reachability

\[ \text{sens}_1 \quad \text{sink} \quad \text{sens}_2 \quad \text{sens}_3 \]
Wireless Sensor Networks

**Phases:**

- Sink $\rightarrow$ Sensors:
  - Data request
  - Software updates

- Sensors $\rightarrow$ Sink:
  - Data collection.

- Directed Acyclic
  - Motivation
  - Reachability
  - Bounded Depth Reachability
Wireless Sensor Networks

**Phases:**

- **Sink → Sensors:**
  - Data request
  - Software updates

- **Sensors → Sink:**
  - Data collection

Root Discovery Protocol
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability
Directed Acyclic

Control State Reachability (COVER)

Given

- Process P
- Control State q

- Motivation
- Reachability
- Bounded Depth Reachability
Directed Acyclic

Control State Reachability (COVER)

- Process P
- Control State q

- Motivation
- Reachability
- Bounded Depth Reachability

Process

Given

i ➔ r(hello) ➔ 2 ➔ b(ack) ➔ 4

init ➔ i ➔ 1 ➔ b(hello) ➔ 3 ➔ r(ack) ➔ q

i ➔ i ➔ i ➔ i

i ➔ i ➔ i ➔ i
Directed Acyclic

Control State Reachability (COVER)

- Process P
- Control State q

- Motivation
- Reachability
- Bounded Depth Reachability
Control State Reachability (COVER)

Is still Undecidable
Control State Reachability (COVER)

Is still Undecidable

Given A, B and T, is there \( k \)

\[ T^k(L_A) \cap L_B \neq \emptyset \]
Control State Reachability (COVER)

Is still Undecidable

Given A, B and T, is there $k$

$$T^k(L_A) \cap L_B \neq \emptyset$$
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability
Control State Reachability (COVER)

Is still Undecidable
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability

Bounded Depth Control State Reachability
Directed Acyclic

Bounded Depth Control State Reachability

- Motivation
- Reachability
- Bounded Depth Reachability
Bounded Depth Control State Reachability
- Motivation
- Reachability
- Bounded Depth Reachability
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability

Bounded Depth Control State Reachability

Diagram showing a directed acyclic graph with nodes labeled $n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7$. The graph illustrates the concept of bounded depth reachability in Ad-Hoc Networks.
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability

Bounded Depth Control State Reachability
Motivation
Reachability
Bounded Depth Reachability

Directed Acyclic

Bounded Depth Control State Reachability

\[ n_1 \rightarrow n_2 \rightarrow n_3 \rightarrow \ldots \rightarrow n_k \]

\[ k \rightarrow n_0 \rightarrow n_1 \rightarrow n_2 \rightarrow \ldots \rightarrow n_k \]
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability

Given

k Bounded Depth Control State Reachability
Given Process P

- Process P

**Directed Acyclic**

- Motivation
- Reachability
- Bounded Depth Reachability

**k Bounded Depth**

**Control State Reachability**

![Diagram showing a process with states and transitions, including
r(hello), b(ack), init, b(hello), r(ack)]
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability

**Given**
- Process $P$
- Control State $q$

**Bounded Depth Control State Reachability**

**Process**

1. $r(\text{hello})$
2. $b(\text{ack})$
3. $b(\text{hello})$
4. $r(\text{ack})$

Diagram:
- Node 1: $i$, $\text{init}$
- Node 2: $\text{r(\text{hello})}$
- Node 3: $\text{b(\text{hello})}$
- Node 4: $\text{b(\text{ack})}$
- Node q: $\text{r(\text{ack})}$
Directed Acyclic

\[ k \text{ Bounded Depth} \]

Control State Reachability

(BOUNDED-COVER)

Given

- Process \( P \)
- Control State \( q \)

- Motivation
- Reachability
- Bounded Depth Reachability

Process

- \( r(\text{hello}) \) from 1 to 2
- \( b(\text{ack}) \) from 3 to 4
- \( b(\text{hello}) \) from 3 to 1
- \( r(\text{ack}) \) from 2 to 1
- \( \text{init} \) from 1 to 2
Directed Acyclic

Bounded Depth

Control State Reachability

(BOUNDED-COVER)

Given

- Process $P$
- Control State $q$

Motivation
Reachability
Bounded Depth Reachability

Process

![Diagram of a process with states and transitions](image)

- $r(\text{hello})$
- $b(\text{ack})$
- $b(\text{hello})$
- $r(\text{ack})$

$2 \leq k$
Directed Acyclic

**k** Bounded Depth

Control State Reachability

((BOUNDDED-COVER))

Given

- Process \( P \)
- Control State \( q \)

- Motivation
- Reachability
- Bounded Depth Reachability

Process

- \( r(\text{hello}) \)
- \( b(\text{ack}) \)

\[ r(\text{hello}) \rightarrow b(\text{ack}) \rightarrow 4 \]

\[ i \rightarrow \text{init} \rightarrow 1 \]

\[ b(\text{hello}) \rightarrow r(\text{ack}) \rightarrow 3 \rightarrow q \]

\[ * \]
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability

\[(\text{BOUNDED-COVER}) \quad \text{DECIDABLE}\]
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability

(BOUNDED-COVER) DECIDABLE

Theory of Well Structured Transition Systems

(1) The WSTS framework
(2) Reduce (BOUNDED-COVER) (TREE-BOUNDED-COVER)
(3) Define an ordering on configurations
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(1) The WSTS framework
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(1) The WSTS framework

Pre-requisites

- $\sqsubseteq$ is a Well-Quasi Order
- $\longrightarrow$: Monotonic wrt. $\sqsubseteq$
Directed Acyclic

(BOUNCED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(1) The WSTS framework

Pre-requisites

- $\sqsubseteq$ is a Well-Quasi Order
- $\rightarrow$: Monotonic wrt. $\sqsubseteq$

Motivation
Reachability
Bounded Depth Reachability
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(1) The WSTS framework

Pre-requisites

- $\sqsupseteq$ is a Well-Quasi Order
- $\rightarrow$: Monotonic wrt. $\sqsubseteq$

\[
\forall (C_i)_{i \geq 0} \\
C_0 \rightarrow C_1 \rightarrow \ldots
\]
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(1) The WSTS framework

Pre-requisites

- $\sqsubseteq$ is a Well-Quasi Order
- $\rightarrow$: Monotonic wrt. $\sqsubseteq$

\[ \forall \ (C_i)_{i \geq 0} \]
\[ C_0 \rightarrow C_1 \rightarrow \ldots \]
\[ \exists \ i < j ; C_i \sqsubseteq C_j \]
Directed Acyclic

Pre-requisites

- Monotonic wrt. $\sqsubseteq$
- $\sqsubseteq$ is a Well-Quasi Order

(1) The WSTS framework

Theory of Well Structured Transition Systems

(BOUNDED-COVER) DECIDABLE

Motivation

Bounded Reachability

Reachability

Bounded Depth Reachability
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(1) The WSTS framework

Pre-requisites

- $\sqsubseteq$ is a Well-Quasi Order
- $\rightharpoonup$: Monotonic wrt. $\sqsubseteq$

```
C_3
\sqsubseteq
C_1 \rightarrow C_2
```
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(1) The WSTS framework

Pre-requisites

- $\sqsubseteq$ is a Well-Quasi Order
- $\rightarrow$: Monotonic wrt. $\sqsubseteq$

- Motivation
- Reachability
- Bounded Depth Reachability

$C_1 \rightarrow C_2$

$C_3 \sqsupseteq C_4$

$\sqsupseteq$

$C_1 \rightarrow C_2$
Directed Acyclic

(BOUNDDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(1) The WSTS framework

Pre-requisites

- \( \sqsubseteq \) is a Well-Quasi Order
- \( \rightarrow \): Monotonic wrt. \( \sqsubseteq \)

\[
\begin{align*}
\quad & C_3 & C_4 \\
\sqsubseteq & \qquad & \sqsubseteq \\
C_1 & \rightarrow & C_2
\end{align*}
\]
Directed Acyclic

(BOUNDED-COVER) \textbf{DECIDABLE}
Theory of Well Structured Transition Systems

(1) The WSTS framework

Pre-requisites

\[\begin{array}{c}
\subseteq \text{ is a Well-Quasi Order} \\
\rightarrow: \text{Monotonic wrt. } \subseteq
\end{array}\]

\[\begin{array}{c}
C_3 \rightarrow C_4 \\
\square \rightarrow \square
\end{array}\]

\[\begin{array}{c}
C_1 \rightarrow C_2
\end{array}\]
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

1) The WSTS framework

Pre-requisites
- $\sqsubseteq$ is a Well-Quasi Order
- $\rightarrow$: Monotonic wrt. $\sqsubseteq$

WSTS framework Algorithm:
(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(1) The WSTS framework

Pre-requisites

- $\sqsubseteq$ is a Well-Quasi Order
- $\rightarrow$: Monotonic wrt. $\sqsubseteq$

WSTS framework Algorithm:

- Symbolic Representation of Infinite Sets
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(1) The WSTS framework

Pre-requisites
- $\sqsubseteq$ is a Well-Quasi Order
- $\rightarrow$: Monotonic wrt. $\sqsubseteq$

WSTS framework Algorithm:
- Symbolic Representation of Infinite Sets
- Backward Analysis
Directed Acyclic

Theory of Well Structured Transition Systems

- Motivation
- Reachability
- Bounded Depth Reachability

(\text{BOUNDED-COVER}) \text{ DECIDABLE}

1. The WSTS framework
2. Reduce (\text{BOUNDED-COVER}) (\text{TREE-BOUNDED-COVER})
3. Define an ordering on configurations
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

Reduce (BOUNDED-COVER) (TREE-BOUNDED-COVER)
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

Reduce (BOUNDED-COVER) (TREE-BOUNDED-COVER)
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(2) Reduce (BOUNDED-COVER) (TREE-BOUNDED-COVER)

Motivation
Reachability
Bounded Depth Reachability
Directed Acyclic

(BOUNDDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(2) Reduce (BOUNDDED-COVER) (TREE-BOUNDDED-COVER)
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(2) Reduce (BOUNDED-COVER) (TREE-BOUNDED-COVER)
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(2) Reduce (BOUNDED-COVER) (TREE-BOUNDED-COVER)

Motivation
Reachability
Bounded Depth Reachability
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

Reduce (BOUNDED-COVER) (TREE-BOUNDED-COVER)
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(2) Reduce (BOUNDED-COVER) (TREE-BOUNDED-COVER)
Directed Acyclic

(BOUNDDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(2) Reduce (BOUNDDED-COVER) (TREE-BOUNDDED-COVER)
Directed Acyclic

**(BOUNDED-COVER)** **DECIDABLE**
Theory of Well Structured Transition Systems

(2) Reduce **(BOUNDED-COVER)** **(TREE-BOUNDED-COVER)**
Define an ordering on configurations

Reduce \textit{(BOUNDED-COVER)} \textit{(TREE-BOUNDED-COVER)}

The WSTS framework

Theory of Well Structured Transition Systems

1. The WSTS framework
2. Reduce \textit{(BOUNDED-COVER)} \textit{(TREE-BOUNDED-COVER)}
3. Define an ordering on configurations
Define an ordering on configurations
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(3) Define an ordering on configurations
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(3) Define an ordering on configurations
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(3) Define an ordering on configurations
(BOUNDDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(3) Define an ordering on configurations
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(3) Define an ordering on configurations
(3) Define an ordering on configurations

- Motivation
- Reachability
- Bounded Depth Reachability

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

Define an ordering on configurations
Motivation
Reachability
Bounded Depth Reachability

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(3) Define an ordering on configurations
Directed Acyclic

Monotonicity

Theory of Well Structured Transition Systems

(3) Define an ordering on configurations
Directed Acyclic

Monotonicity

**(BOUNDDED-COVER)** DECIDABLE

Theory of Well Structured Transition Systems

(3) Define an ordering on configurations
Directed Acyclic

Monotonicity

\textbf{(BOUNDED-COVER) DECIDABLE}

Theory of Well Structured Transition Systems

(3) Define an ordering on configurations

- Broadcast: c -> d
Directed Acyclic

Monotonicity

**(BOUNDDED-COVER) DECIDABLE**

Theory of Well Structured Transition Systems

(3) Define an ordering on configurations

- Broadcast: c -> d
- Receive: b -> f
(3) Define an ordering on configurations

- Broadcast: c \rightarrow d
- Receive: b \rightarrow f
Directed Acyclic

- Motivation
- Reachability
- Bounded Depth Reachability

Monotonicity (BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(3) Define an ordering on configurations

- Broadcast: c -> d
- Receive: b -> f
- Local: a -> g
Directed Acyclic

Monotonicity
\textbf{(BOUNDED-COVER)} \textbf{DECIDABLE}
Theory of Well Structured Transition Systems

(3) Define an ordering on configurations

- Broadcast: $c \rightarrow d$
- Receive: $b \rightarrow f$
- Local: $a \rightarrow g$
Ad-Hoc Networks

- Motivation
- Reachability
- Bounded Depth Reachability

Directed Acyclic

- Model
- Transition System
- Reachability
FUTURE WORK

- Consider:
  - REPEATED-COVER
  - Bounded number of Phases

- Dynamic Communicating Automata
Directed Acyclic

- TRANSD proof
Directed Acyclic

Control State Reachability (COVER)

Undecidable
Directed Acyclic

- TRANSD proof

Cover is **Undecidable**

Sketch of the proof:

- TRANSD Problem
- TRANSD Undecidable
- Encode TRANSD into COVER
Undecidable

Sketch of the proof:
- TRANSD Problem
- TRANSD Undecidable
- Encode TRANSD into COVER
Undecidable

Sketch of the proof:
- TRANSD Problem
- TRANSD Undecidable
- Encode TRANSD into COVER

TRANSD proof
Undecidable

Sketch of the proof:

- TRANSD Problem
- TRANSD Undecidable
- Encode TRANSD into COVER

Directed Acyclic

TRANSD proof
Directed Acyclic

Undecidable

Sketch of the proof:
- TRANSD Problem
- TRANSD Undecidable
- Encode TRANSD into COVER

TRANSD proof
Undecidable

Sketch of the proof:
• TRANSND Problem
• TRANSND Undecidable
• Encode TRANSND into COVER

Input / Output Language
Undecidable

Sketch of the proof:

• TRANSD Problem
  • TRANSD Undecidable
  • Encode TRANSD into COVER

TRANSD proof
Directed Acyclic

Undecidable

Sketch of the proof:
• TRANSD Problem
• TRANSD Undecidable
• Encode TRANSD into COVER

TRANSD proof
Directed Acyclic

Undecidable

Sketch of the proof:
• TRANSND Problem
• TRANSND Undecidable
• Encode TRANSND into COVER

TRANSD proof
Directed Acyclic

Undecidable

Sketch of the proof:
• TRANSD Problem
  • TRANSD Undecidable
  • Encode TRANSD into COVER

TRANSD proof
Directed Acyclic

Undecidable

Sketch of the proof:
• TRANSD Problem
  • TRANSD Undecidable
  • Encode TRANSD into COVER

TRANSD proof
Directed Acyclic

Undecidable

Sketch of the proof:
• TRANSD Problem
• TRANSD Undecidable
• Encode TRANSD into COVER

TRANSD proof
Directed Acyclic

Undecidable

Sketch of the proof:
• TRANSD Problem
  • TRANSD Undecidable
  • Encode TRANSD into COVER
  • TRANSD proof

\[ T^2(L_A) \]
Undecidable

Sketch of the proof:

• TRANSD Problem
  • TRANSD Undecidable
  • Encode TRANSD into COVER

\[ T^k(L_A) \]
Directed Acyclic

Undecidable

Sketch of the proof:
• TRANSD Problem
  • TRANSD Undecidable
  • Encode TRANSD into COVER

TRANSD proof

\[ T^k(L_A) \]
Directed Acyclic

Undecidable

Sketch of the proof:
• TRANSD Problem
• TRANSD Undecidable
• Encode TRANSD into COVER

Given A, B and T, is there \( k \)

\( T^k(L_A) \)
Undecidable

Sketch of the proof:
• TRANSD Problem
• TRANSD Undecidable
• Encode TRANSD into COVER

Given A, B and T, is there $k$

$$T^k(L_A) \cap L_B \neq \emptyset$$
Undecidable

Sketch of the proof:

• TRANSD Problem
• TRANSD Undecidable
• Encode TRANSD into COVER
Undecidable

Sketch of the proof:

- TRANSD Problem
- TRANSD Undecidable
- Encode TRANSD into COVER

TRANSD proof
Undecidable

Sketch of the proof:

- TRANSD Problem
- TRANSD Undecidable
- Encode TRANSD into COVER

TRANSD proof
Directed Acyclic

Undecidable

Sketch of the proof:
- TRANSD Problem
- TRANSD Undecidable
- Encode TRANSD into COVER

TRANSD proof
Directed Acyclic

Monotonicity

Process transitions:
Directed Acyclic

Monotonicity

Process transitions:

- Broadcast: c -> d
Directed Acyclic

Monotonicity

Process transitions:

- Broadcast: c -> d
- Receive: b -> f
Directed Acyclic

Monotonicity

Process transitions:

- Broadcast: c -> d
- Receive: b -> f
Directed Acyclic

Motivation
Reachability
Bounded Depth Reachability

Monotonicity

Process transitions:
- Broadcast: c -> d
- Receive: b -> f
- Local: a -> g
(BOUNDDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

Use the WSTS framework

Γ: Inverted Tree Configurations \( \gamma \mid \text{height}(\gamma) = q_0 \)
Use the WSTS framework

- $\Gamma$: Inverted Tree Configurations $\gamma | \text{height}(\gamma) = q_0$
- $\Gamma_{\text{init}}$: $\gamma \in \Gamma | \text{States}(\gamma) = q_0$
Use the WSTS framework

- \( \Gamma : \) Inverted Tree Configurations \( \gamma \mid \text{height}(\gamma) = q_0 \)
- \( \Gamma_{\text{init}} : \gamma \in \Gamma \mid \text{States}(\gamma) = q_0 \)
- \( \sqsubseteq : \) Higher Order Multiset Ordering. Computable and is a Well-Quasi Order
Use the WSTS framework

- $\Gamma$: Inverted Tree Configurations $\gamma|\text{height}(\gamma) = q_0$
- $\Gamma_{\text{init}}$: $\gamma \in \Gamma|\text{States}(\gamma) = q_0$
- $\sqsubseteq$: Higher Order Multiset Ordering. Computable and is a Well-Quasi Order
- $\rightarrow$: Monotonic wrt. $\sqsubseteq$
Use the WSTS framework

- $\Gamma$: Inverted Tree Configurations $\gamma|height(\gamma) = q_0$
- $\Gamma_{init}$: $\gamma \in \Gamma|States(\gamma) = q_0$
- $\subseteq$: Higher Order Multiset Ordering. Computable and is a Well-Quasi Order
- $\rightarrow$: Monotonic wrt. $\subseteq$
- $U$: Upward closed set; minimal element $\{q\}$
Directed Acyclic

(BOUNDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(3) Use the WSTS framework

Reachability is decidable if, for any $\gamma \in \Gamma$
(Bounded-Cover) **Decidable**
Theory of Well Structured Transition Systems

(3) Use the WSTS framework

Reachability is decidable if, for any $\gamma \in \Gamma$

- We can check if $\gamma \in \Gamma_{init}$
(BOUNDDED-COVER) DECIDABLE
Theory of Well Structured Transition Systems

(3) Use the WSTS framework

Reachability is decidable if, for any \( \gamma \in \Gamma \)

- We can check if \( \gamma \in \Gamma_{init} \)
- We can compute the minimal set of \( Pre(\gamma) \), and it’s finite