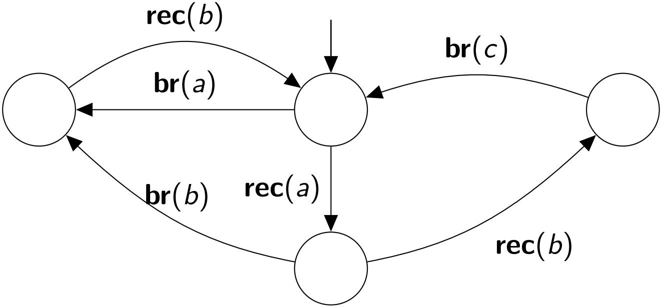
Parameterized verification of Broadcast networks of Register automata

Corto Mascle
joint work with Lucie Guillou and Nicolas Waldburger

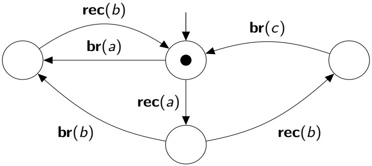
June 8th, 2023

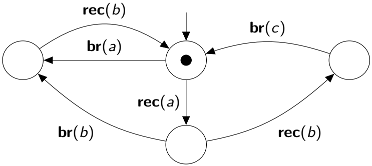
1. Broadcast networks
   - Basic model
   - With registers
2. Signature BNRA
   - Well quasi-orders
   - Decidability proof
3. General case
4. Complexity bounds

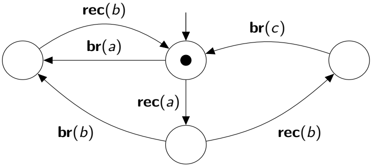
Broadcast networks



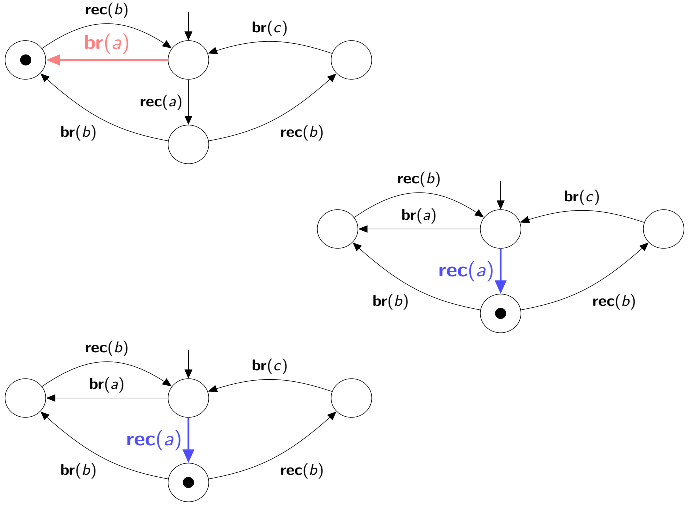
Broadcast networks



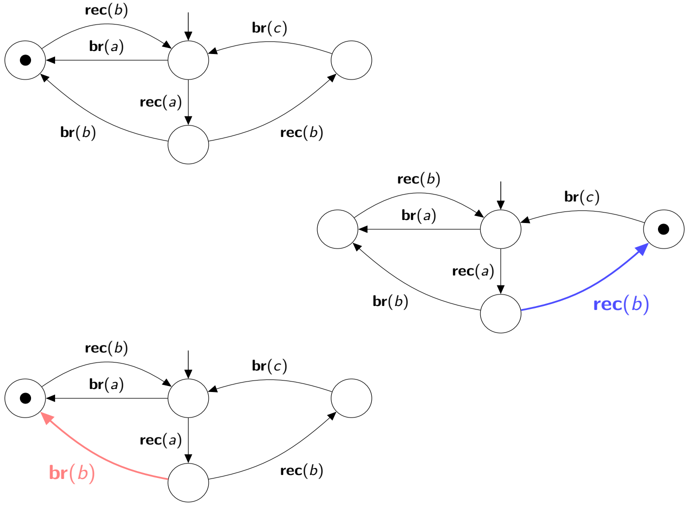




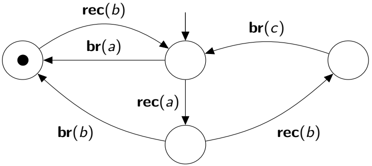
Broadcast networks

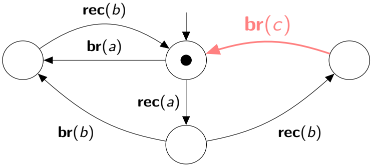


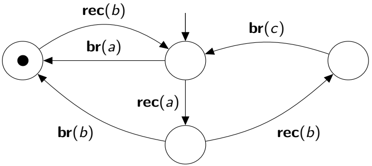
Broadcast networks



Broadcast networks







Broadcast Networks

Definition[1]

(Reconfigurable) Broadcast Network $= (Q, M, \Delta, q_0)$ with
$\Delta \subseteq Q \times \{\mathbf{br}(m), \mathbf{rec}(m) \mid m \in M\} \times Q$.

[1]Delzanno, Sangnier, Zavattaro, CONCUR'10
[2]Delzanno, Sangnier, Traverso, Zavattaro, FSTTCS'12

Broadcast Networks

Definition[1]

(Reconfigurable) Broadcast Network $= (Q, M, \Delta, q_0)$ with
$\Delta \subseteq Q \times \{\mathbf{br}(m), \mathbf{rec}(m) \mid m \in M\} \times Q$.

- Arbitrarily many agents at the start
- One step = an agent broadcasts a message m, some (arbitrary subset of) other agents receive it.

[1]Delzanno, Sangnier, Zavattaro, CONCUR'10
[2]Delzanno, Sangnier, Traverso, Zavattaro, FSTTCS'12

Broadcast Networks

Definition[1]

(Reconfigurable) Broadcast Network $= (Q, M, \Delta, q_0)$ with
$\Delta \subseteq Q \times \{\mathbf{br}(m), \mathbf{rec}(m) \mid m \in M\} \times Q$.

- Arbitrarily many agents at the start
- One step = an agent broadcasts a message m, some (arbitrary subset of) other agents receive it.

Problems

COVER: Is there a run in which **an** agent reaches q_f?
TARGET: Is there a run in which **all agents** reach q_f **simultaneously**?

Both problems are decidable in PTIME[1,2].

[1] Delzanno, Sangnier, Zavattaro, CONCUR'10
[2] Delzanno, Sangnier, Traverso, Zavattaro, FSTTCS'12

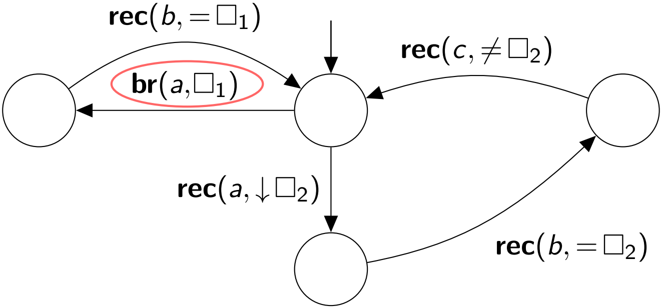
1. Broadcast networks
   - Basic model
   - With registers
2. Signature BNRA
   - Well quasi-orders
   - Decidability proof
3. General case
4. Complexity bounds

Registers

Each agent now has local *registers* $\square_1, \ldots, \square_r$, containing values in $\mathbb{N}$.

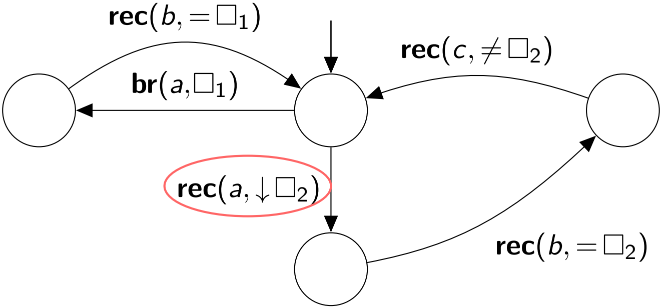
Registers

Each agent now has local *registers* $\square_1, \ldots, \square_r$, containing values in $\mathbb{N}$.



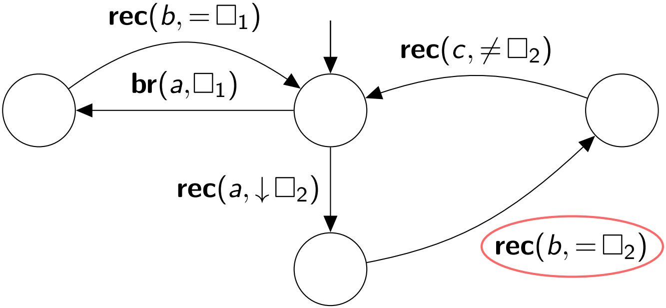
Registers

Each agent now has local *registers* $\square_1, \ldots, \square_r$, containing values in $\mathbb{N}$.



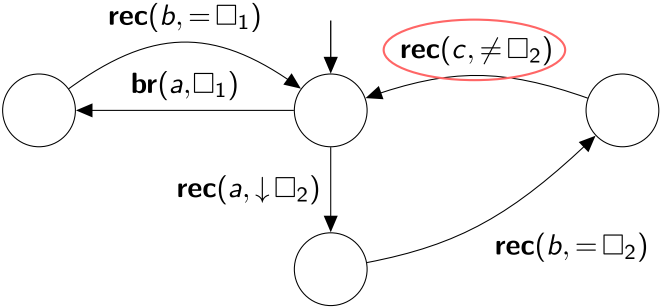
Registers

Each agent now has local *registers* $\square_1, \ldots, \square_r$, containing values in $\mathbb{N}$.



Registers

Each agent now has local *registers* $\square_1, \ldots, \square_r$, containing values in $\mathbb{N}$.



Broadcast Networks of Register Automata (BNRA)[3]

Each agent now has local *registers* $\square_1, \ldots, \square_r$, containing values in $\mathbb{N}$.

[3] Delzanno, Sangnier, Traverso, RP'13

Broadcast Networks of Register Automata (BNRA)[3]

Each agent now has local *registers* $\square_1, \ldots, \square_r$, containing values in $\mathbb{N}$.
Initially, all registers of all agents contain distinct values.

[3] Delzanno, Sangnier, Traverso, RP'13

Broadcast Networks of Register Automata (BNRA)[3]

Each agent now has local *registers* $\square_1, \ldots, \square_r$, containing values in $\mathbb{N}$.
Initially, all registers of all agents contain distinct values.

Messages also contain values: $(m, v) \in M \times \mathbb{N}$. An agent can:

- Broadcast a message with a register value $\mathbf{br}(m, r_i)$

[3] Delzanno, Sangnier, Traverso, RP'13

Broadcast Networks of Register Automata (BNRA)[3]

Each agent now has local *registers* $\square_1, \ldots, \square_r$, containing values in $\mathbb{N}$.
Initially, all registers of all agents contain distinct values.

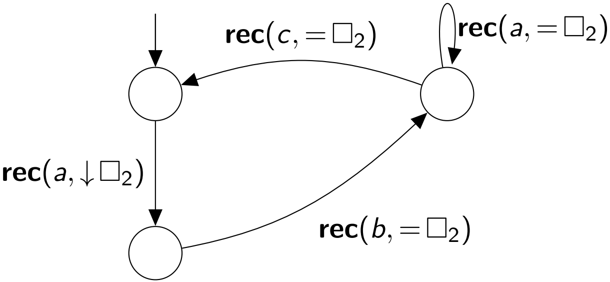
Messages also contain values: $(m, v) \in M \times \mathbb{N}$. An agent can:

- Broadcast a message with a register value $\mathbf{br}(m, r_i)$
- Receive messages $\mathbf{rec}(m, r_i, op)$, with *op* either
  - store the value $\downarrow$,
  - test it for equality $=, \neq$
  - or do nothing $*$.

[3] Delzanno, Sangnier, Traverso, RP'13

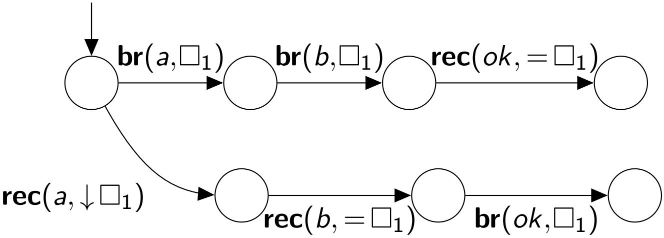
Things we can do

We can check that a sequence of messages all come from the same process.



Things we can do

We can check that a sequence of messages we sent was received.



Parameterized verification principles

- Unlimited supply of agents.
- For COVER, we can add as many agents as we need.

Parameterized verification principles

- Unlimited supply of agents.
- For Cover, we can add as many agents as we need.

Copycat principle

Given a run ρ, we can construct a run made of many copies of ρ running in parallel.

Main theorem

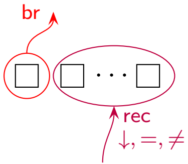
Cover is decidable for BNRA.

1. Broadcast networks
   - Basic model
   - With registers
2. Signature BNRA
   - Well quasi-orders
   - Decidability proof
3. General case
4. Complexity bounds

Signature BNRA

Signature BNRA

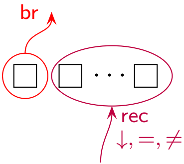
A process never modifies its first register, and only broadcasts with its value.
Other registers are used to store and compare values received.



Signature BNRA

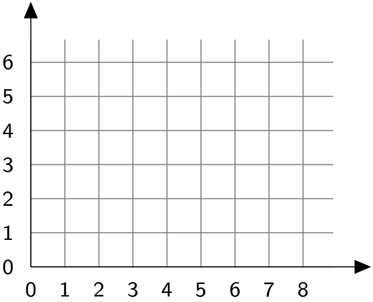
Signature BNRA

A process never modifies its first register, and only broadcasts with its value.
Other registers are used to store and compare values received.

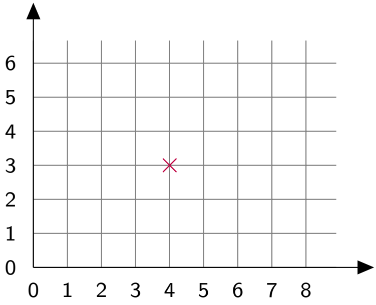


Messages received with the same value come from the same process.

Well quasi-orders

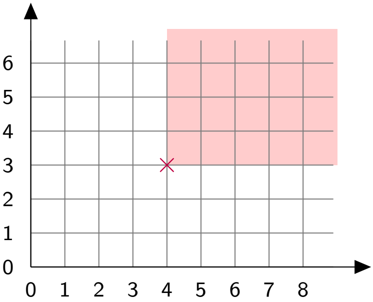


Well quasi-orders



$(4, 3)$

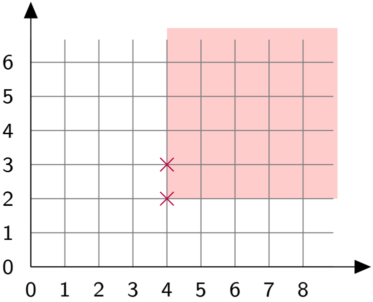
Well quasi-orders



$(4, 3)$

- You cannot pick a point higher on both coordinates than one of the previous ones.

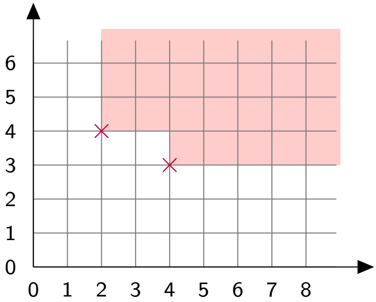
Well quasi-orders



$(4,3) \rightarrow (4,2)$

- You cannot pick a point higher on both coordinates than one of the previous ones.

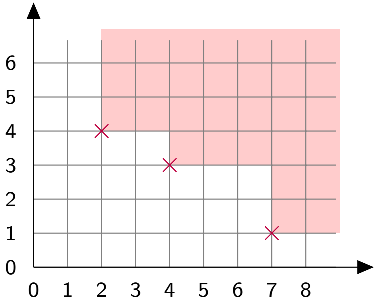
Well quasi-orders



$(4, 3) \rightarrow (2, 4)$

- You cannot pick a point higher on both coordinates than one of the previous ones.

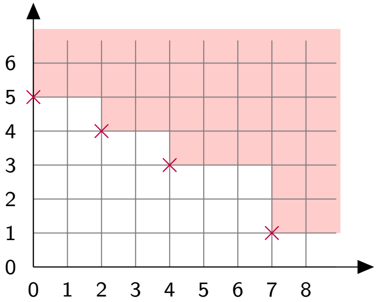
Well quasi-orders



$(4,3) \rightarrow (2,4) \rightarrow (7,1)$

- You cannot pick a point higher on both coordinates than one of the previous ones.

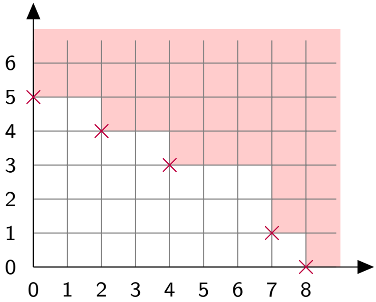
Well quasi-orders



$(4,3) \rightarrow (2,4) \rightarrow (7,1) \rightarrow (0,5)$

- You cannot pick a point higher on both coordinates than one of the previous ones.

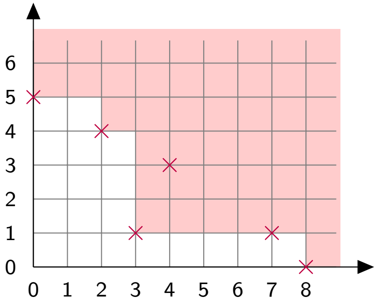
Well quasi-orders



$(4,3) \rightarrow (2,4) \rightarrow (7,1) \rightarrow (0,5) \rightarrow (8,0)$

- You cannot pick a point higher on both coordinates than one of the previous ones.

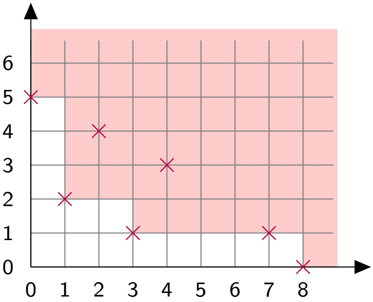
Well quasi-orders



- You cannot pick a point higher on both coordinates than one of the previous ones.

$$(4,3) \rightarrow (2,4) \rightarrow (7,1) \rightarrow (0,5) \rightarrow (8,0) \rightarrow (3,1)$$

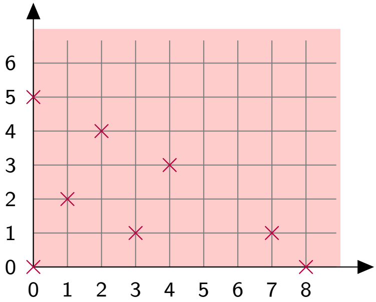
Well quasi-orders



- You cannot pick a point higher on both coordinates than one of the previous ones.

$$(4,3) \to (2,4) \to (7,1) \to (0,5) \to (8,0) \to (3,1) \to (1,2)$$

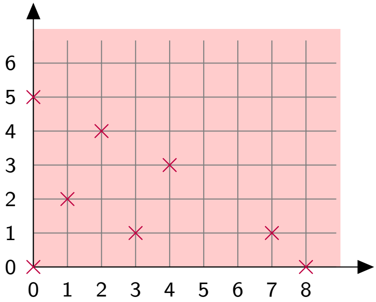
Well quasi-orders



- You cannot pick a point higher on both coordinates than one of the previous ones.

$$(4,3)\rightarrow (2,4)\rightarrow (7,1)\rightarrow (0,5)\rightarrow (8,0)\rightarrow (3,1)\rightarrow (1,2)\rightarrow (0,0)$$

Well quasi-orders

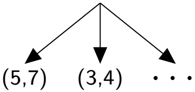


- You cannot pick a point higher on both coordinates than one of the previous ones.
- Your ith point (x_i, y_i) has to be such that $|x_i|, |y_i| \leq 10^i$.

$$(4,3) \rightarrow (2,4) \rightarrow (7,1) \rightarrow (0,5) \rightarrow (8,0) \rightarrow (3,1) \rightarrow (1,2) \rightarrow (0,0)$$

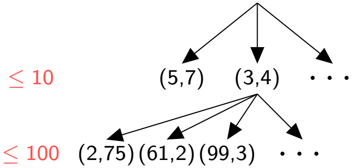
Well quasi-orders

≤ 10



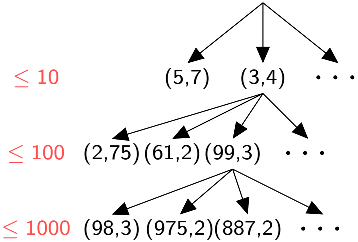
- You cannot pick a point higher on both coordinates than one of the previous ones.
- Your ith point (x_i, y_i) has to be such that $|x_i|, |y_i| \leq 10^i$.

Well quasi-orders



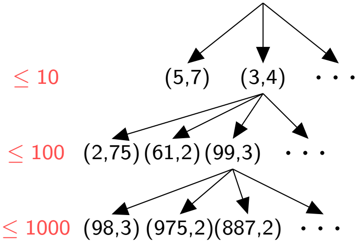
- You cannot pick a point higher on both coordinates than one of the previous ones.
- Your ith point (x_i, y_i) has to be such that $|x_i|, |y_i| \leq 10^i$.

Well quasi-orders



- You cannot pick a point higher on both coordinates than one of the previous ones.
- Your ith point (x_i, y_i) has to be such that $|x_i|, |y_i| \leq 10^i$.

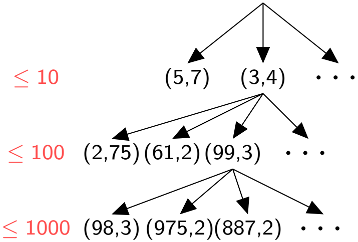
Well quasi-orders



- You cannot pick a point higher on both coordinates than one of the previous ones.
- Your ith point (x_i, y_i) has to be such that $|x_i|, |y_i| \leq 10^i$.

König's lemma $\rightarrow$ this tree is finite.

Well quasi-orders



- You cannot pick a point higher on both coordinates than one of the previous ones.
- Your ith point (x_i, y_i) has to be such that $|x_i|, |y_i| \leq 10^i$.

König's lemma $\rightarrow$ this tree is finite.
We can enumerate all possible sequences!

Well quasi-orders: Subwords

Higman's lemma

For all finite alphabet Σ, the subword order $\preceq$ is a well quasi-order over Σ^*.

Well quasi-orders: Subwords

Higman's lemma

For all finite alphabet Σ, the subword order $\preceq$ is a well quasi-order over Σ^*.

$\Leftrightarrow$ Any sequence $w_0, w_1, w_2, \ldots$ of words over Σ such that $w_i \npreceq w_j$ for all $i < j$ is **finite**.

Well quasi-orders: Subwords

Higman's lemma

For all finite alphabet Σ, the subword order $\preceq$ is a well quasi-order over Σ^*.

$\Leftrightarrow$ Any sequence $w_0, w_1, w_2, \ldots$ of words over Σ such that $w_i \not\preceq w_j$ for all $i < j$ is **finite**.

Given a finite alphabet Σ and a computable function $B : \mathbb{N} \to \mathbb{N}$, the set of sequences $(w_i)_{i \in \mathbb{N}}$ over Σ such that

- $w_i \not\preceq w_j$ for all $i < j$
- $|w_i| \leq B(i)$ for all i

is finite and computable.

1. Broadcast networks
   - Basic model
   - With registers
2. Signature BNRA
   - Well quasi-orders
   - Decidability proof
3. General case
4. Complexity bounds

Tree unfolding

$$\xrightarrow[\textbf{rec}(m_1, v_1)\textbf{rec}(m_2, v_2)\textbf{rec}(m_3, v_1)]{\textbf{br}(m_0, v_0) \qquad\qquad\qquad\qquad\qquad\qquad \textbf{br}(m, v_0)}$$

Tree unfolding

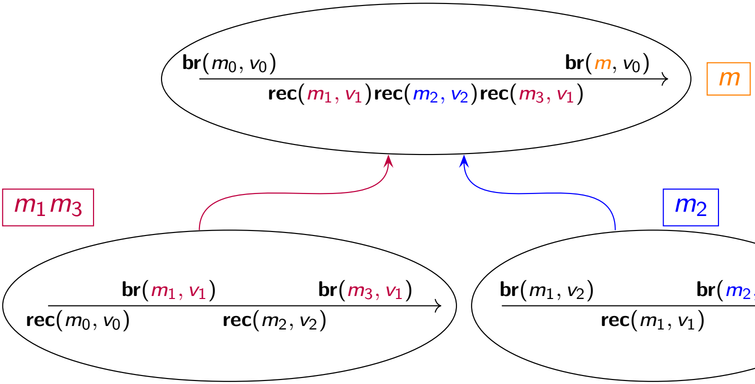
$$\xrightarrow[\mathbf{rec}(m_1, v_1)\mathbf{rec}(m_2, v_2)\mathbf{rec}(m_3, v_1)]{\mathbf{br}(m_0, v_0) \qquad\qquad\qquad\qquad\qquad\qquad \mathbf{br}(m, v_0)}$$

Tree unfolding

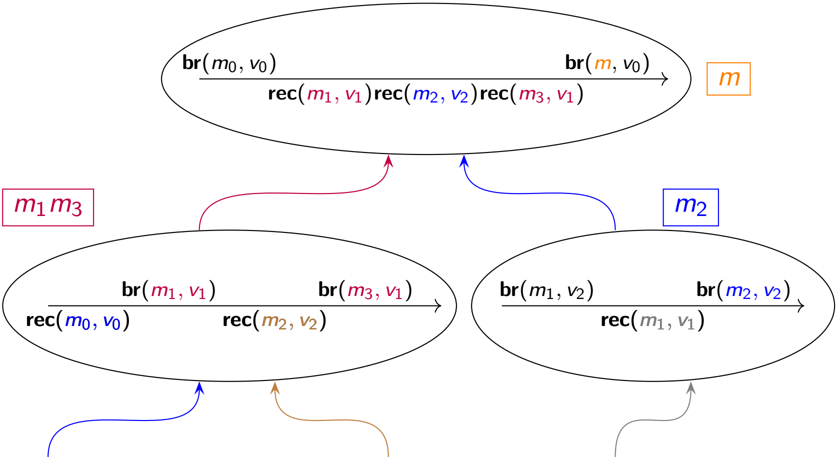
$$\xrightarrow[\mathbf{rec}(m_1, v_1)\mathbf{rec}(m_2, v_2)\mathbf{rec}(m_3, v_1)]{\mathbf{br}(m_0, v_0) \qquad\qquad\qquad \mathbf{br}(m, v_0)}$$

$$\xrightarrow[\mathbf{rec}(m_0, v_0) \qquad\qquad \mathbf{rec}(m_2, v_2)]{\mathbf{br}(m_1, v_1) \qquad\qquad \mathbf{br}(m_3, v_1)} \qquad \xrightarrow[\mathbf{rec}(m_1, v_1)]{\mathbf{br}(m_1, v_2) \qquad\qquad \mathbf{br}(m_2, v_2)}$$

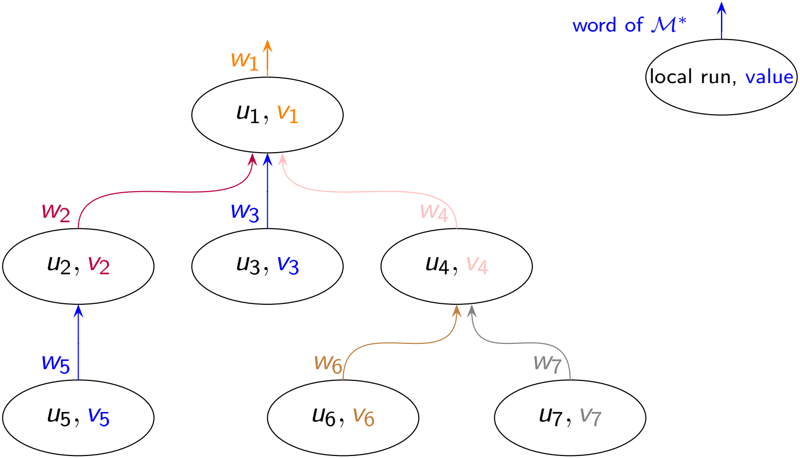
Tree unfolding



Tree unfolding



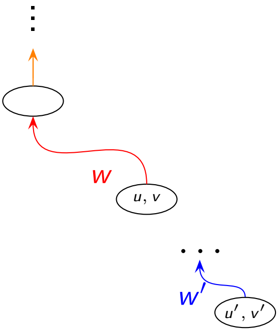
Tree unfolding



Branch reduction

Lemma

If a node labelled w has a descendant labelled w' with w a subword of w' then the tree can be reduced.

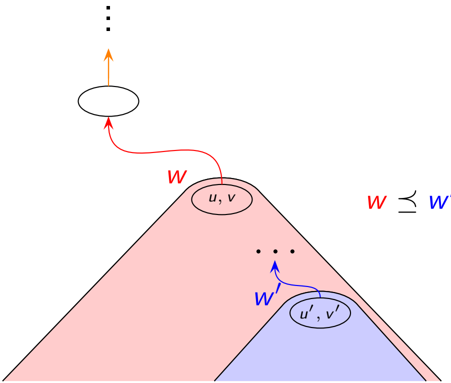


$w \preceq w'$

Branch reduction

Lemma

If a node labelled w has a descendant labelled w' with w a subword of w' then the tree can be reduced.

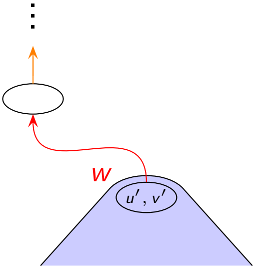


$w \preceq w'$

Branch reduction

Lemma

If a node labelled w has a descendant labelled w' with w a subword of w' then the tree can be reduced.



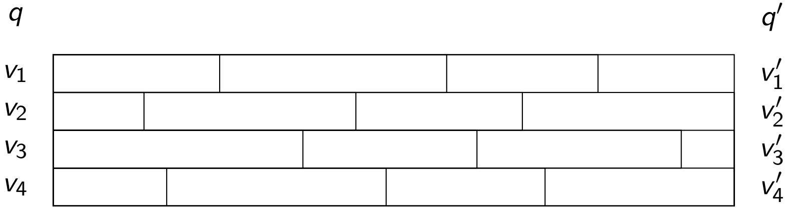
$w \preceq w'$

Branch reduction

- We can assume that a node labelled w has no descendant labelled $w' \succeq w$.
- To bound the height, we need to bound the size of the labels.
- We now aim to reduce long local runs.

Bounding local runs

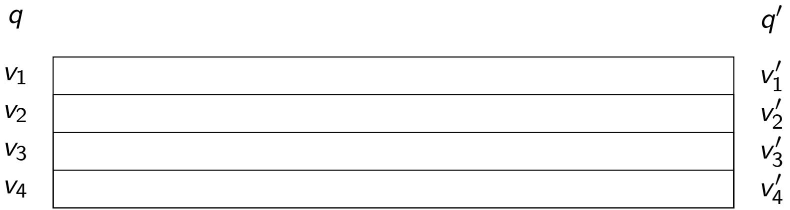
By induction on the number of *active* registers.
Active = a new value is stored at some point.



Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

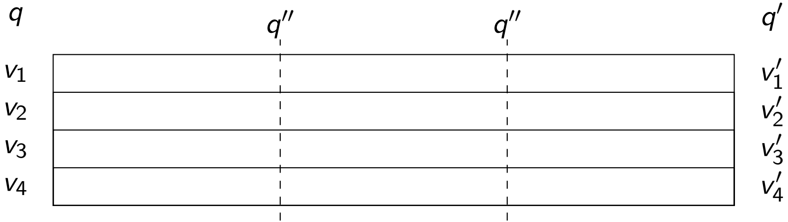
0 active registers



Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

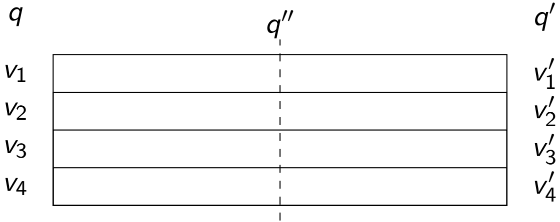
0 active registers



Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

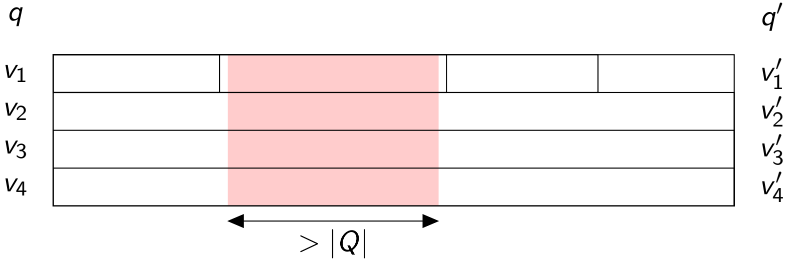
0 active registers



Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

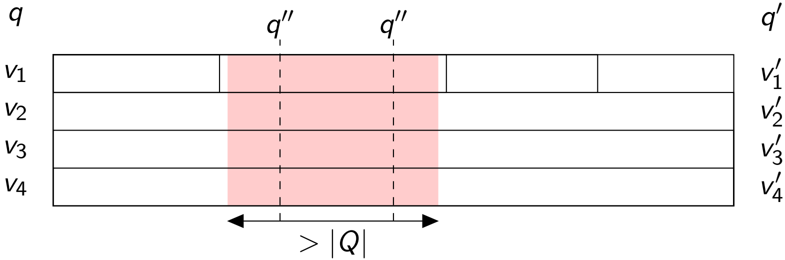
1 active registers



Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

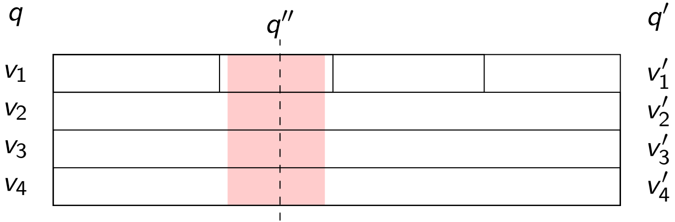
1 active registers



Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

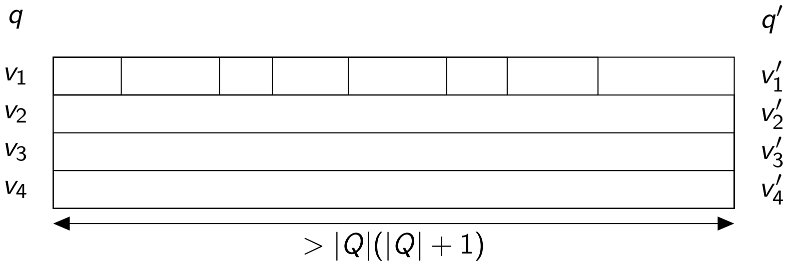
1 active registers



Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

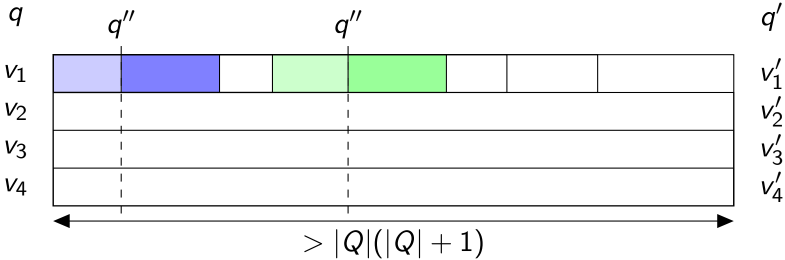
1 active registers



Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

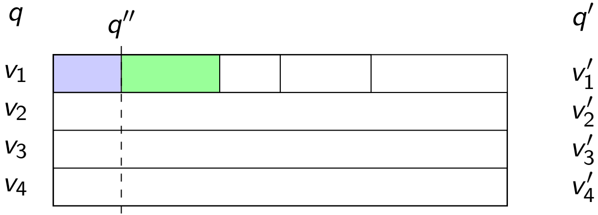
1 active registers



Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

1 active registers

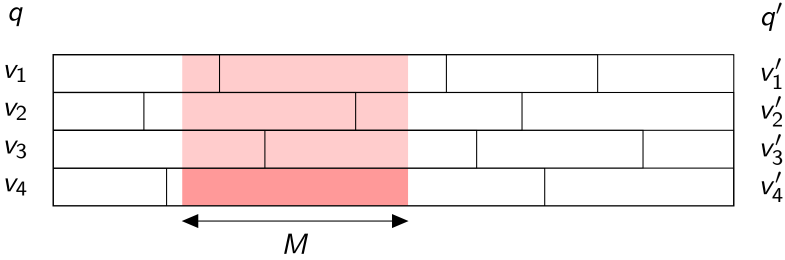


Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

n active registers

Say we can reduce any local run with $< n$ active registers of length $\geq M$.

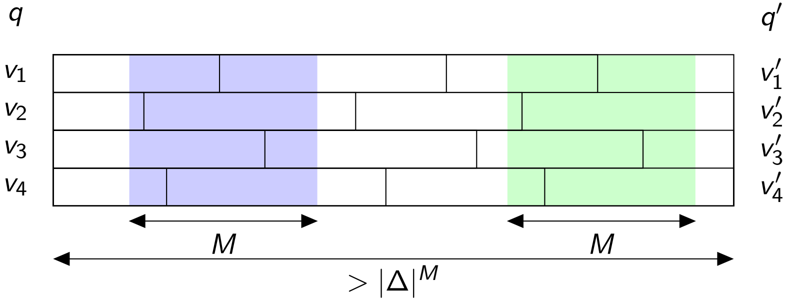


Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

n active registers

Say we can reduce any local run with $< n$ active registers of length $\geq M$.

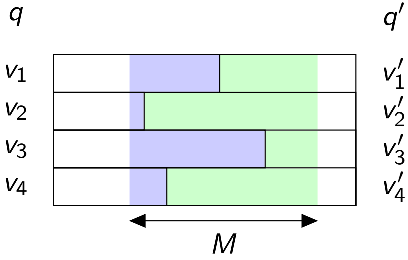


Bounding local runs

By induction on the number of *active* registers.
Active = a new value is stored at some point.

n active registers

Say we can reduce any local run with $< n$ active registers of length $\geq M$.



Bounding the tree

Lemma

There is a function φ such that if an agent has a local run between two local configurations, then it has a "cheaper" one of length $\leq \varphi(|\Delta|, r)$.

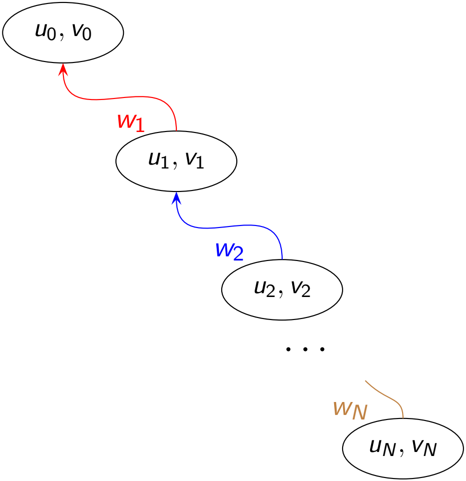
Δ: set of transitions $\quad$ r: number of registers.

Corollary

If an agent has a local run between two local configurations broadcasting $(m_1, v_1) \cdots (m_K, v_K)$, then it has a "cheaper" one of length $\leq K\varphi(|\Delta|, r)$.

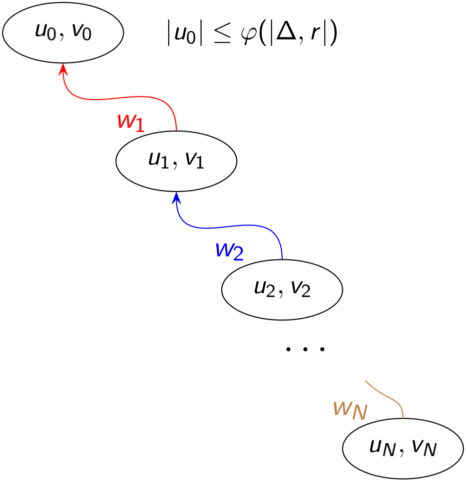
Bounding the branches

root



Bounding the branches

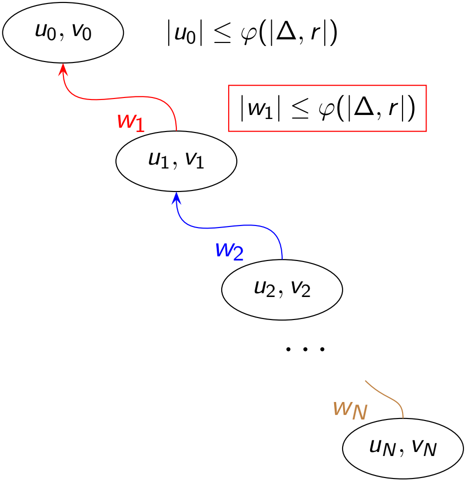
root



$|u_0| \leq \varphi(|\Delta, r|)$

Bounding the branches

root

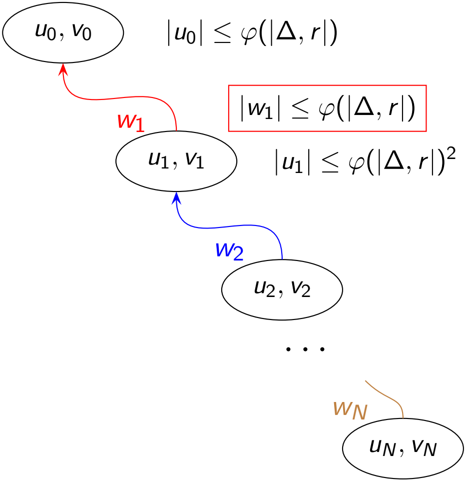


$|u_0| \leq \varphi(|\Delta, r|)$

$\boxed{|w_1| \leq \varphi(|\Delta, r|)}$

Bounding the branches

root



Bounding the branches

root

u_0, v_0

$$|u_0| \leq \varphi(|\Delta, r|)$$

w_1

$$\boxed{|w_1| \leq \varphi(|\Delta, r|)}$$

u_1, v_1

$$|u_1| \leq \varphi(|\Delta, r|)^2$$

w_2

$$\boxed{|w_2| \leq \varphi(|\Delta, r|)^2}$$

u_2, v_2

$$|u_2| \leq \varphi(|\Delta, r|)^3$$

$\cdots$

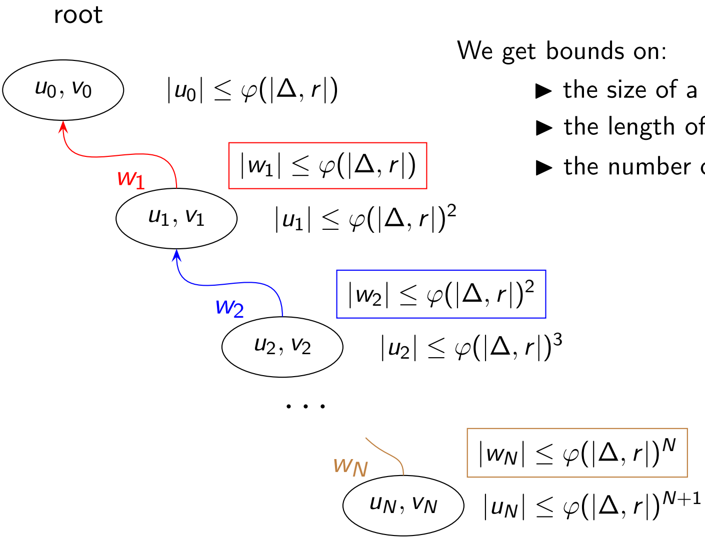
w_N

$$\boxed{|w_N| \leq \varphi(|\Delta, r|)^N}$$

u_N, v_N

$$|u_N| \leq \varphi(|\Delta, r|)^{N+1}$$

Bounding the branches



We get bounds on:

- the size of a node,
- the length of a branch (wqo),
- the number of children of a node.

Decidability

We can enumerate all irreducible trees in finite time, thus

Theorem

The COVER problem is decidable for signature BNRA.

General case

Agents can broadcast messages with values they received before.

An agent now receives:

- Messages with values that belonged to other agents initially.
- Messages with values that the agent had initially, that it has broadcast and receives back.

Observation

An agent may do this:

$$\mathbf{br}(a,\Box_1)\mathbf{br}(b,\Box_1)\mathbf{rec}(c,=\Box_1)\mathbf{rec}(d,=\Box_1)\mathbf{rec}(c,=\Box_1)$$

Observation

An agent may do this:

$$\mathbf{br}(a,\square_1)\mathbf{br}(b,\square_1)\mathbf{rec}(c,=\square_1)\mathbf{rec}(d,=\square_1)\mathbf{rec}(c,=\square_1)$$

To witness that this is feasible, we can exhibit:

- A run that receives $(a, v)(b, v)$ and broadcasts (c, v), and
- A run that receives $(a, v)(c, v)$ and broadcasts (d, v).

Observation

An agent may do this:

$$\mathbf{br}(a,\Box_1)\mathbf{br}(b,\Box_1)\mathbf{rec}(c,=\Box_1)\mathbf{rec}(d,=\Box_1)\mathbf{rec}(c,=\Box_1)$$

To witness that this is feasible, we can exhibit:

- A run that receives $(a,v)(b,v)$ and broadcasts (c,v), and
- A run that receives $(a,v)(c,v)$ and broadcasts (d,v).

We add nodes labelled $\boxed{w \to m}$ that witness that there is a run that, if it receives w over a value v, can broadcast (m,v).

Feasability of a local run

$$\xrightarrow[\textbf{rec}(m_2, v_2)\textbf{rec}(m_3, v_0)\textbf{rec}(m_3, v_0)]{\textbf{br}(m_0, v_0)\ \ \textbf{br}(m_1, v_0)}$$

Feasability of a local run

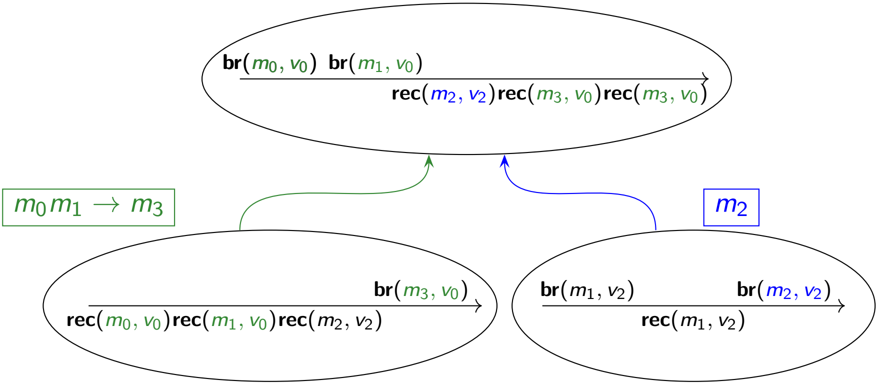
$$\xrightarrow[\mathbf{rec}(m_2, v_2)\mathbf{rec}(m_3, v_0)\mathbf{rec}(m_3, v_0)]{\mathbf{br}(m_0, v_0)\ \ \mathbf{br}(m_1, v_0)}$$

Feasability of a local run

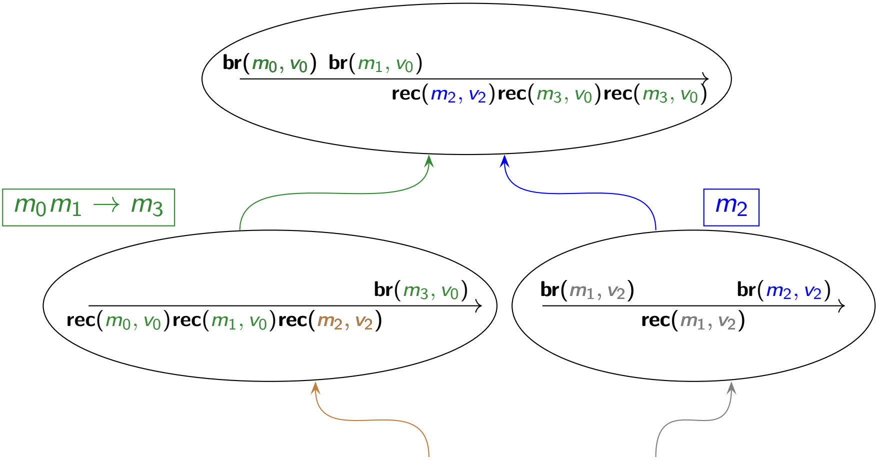
$$\xrightarrow[\textbf{rec}(m_2, v_2)\textbf{rec}(m_3, v_0)\textbf{rec}(m_3, v_0)]{\textbf{br}(m_0, v_0)\ \textbf{br}(m_1, v_0)}$$

$$\xrightarrow[\textbf{rec}(m_0, v_0)\textbf{rec}(m_1, v_0)\textbf{rec}(m_2, v_2)]{\textbf{br}(m_3, v_0)} \qquad \xrightarrow[\textbf{rec}(m_1, v_2)]{\textbf{br}(m_1, v_2)\ \textbf{br}(m_2, v_2)}$$

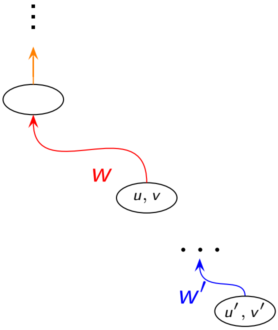
Feasability of a local run



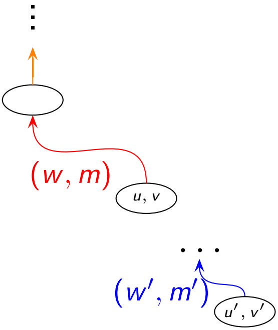
Feasability of a local run



Branch reductions



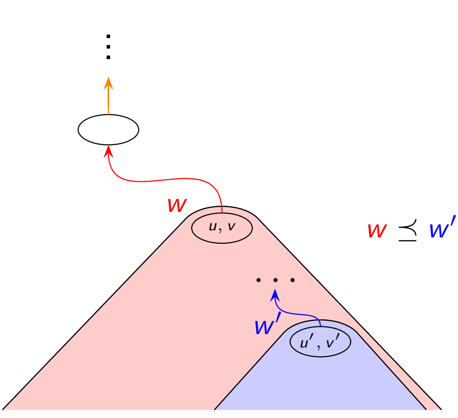
$w \preceq w'$



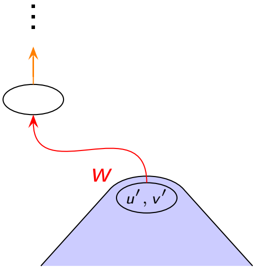
$w \succeq w'$

$m = m'$

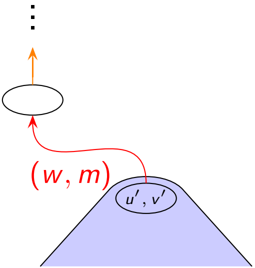
Branch reductions



Branch reductions

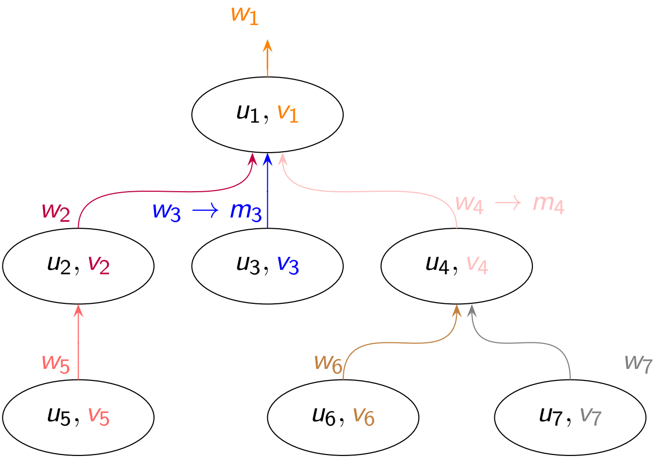


$w \preceq w'$



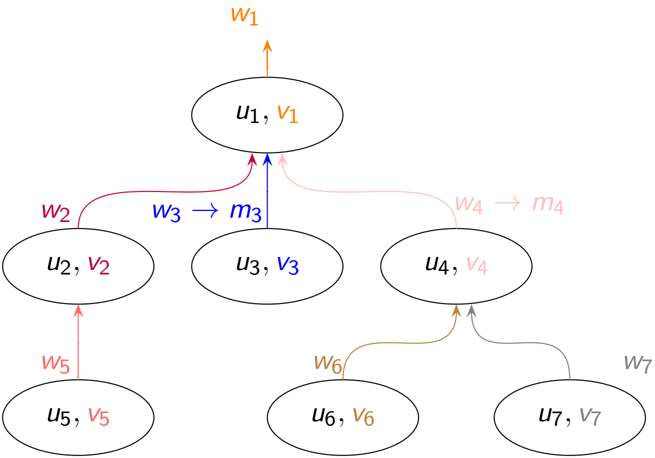
$w \succeq w'$
$m = m'$

New tree



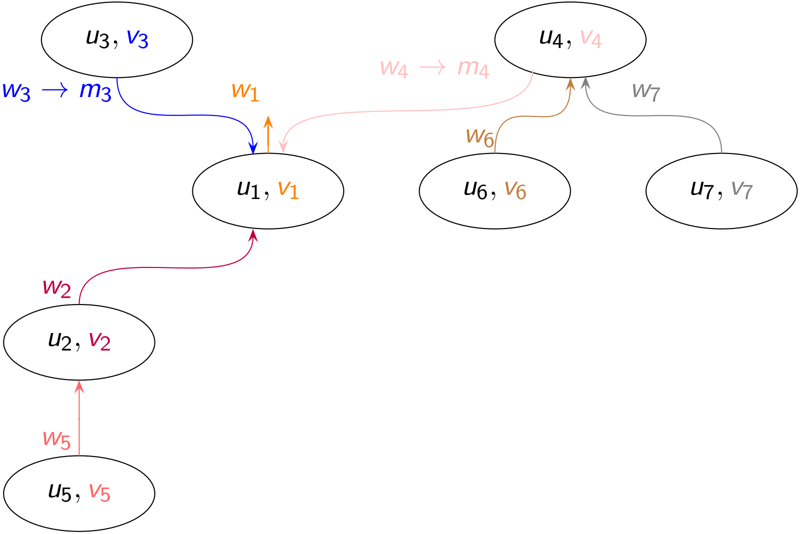
Problem: The length of a node now depends on its $\boxed{w \to m}$ children, and not just on its father.

New tree



w_1
u_1, v_1
w_2
$w_3 \to m_3$
$w_4 \to m_4$
u_2, v_2
u_3, v_3
u_4, v_4
w_5
w_6
w_7
u_5, v_5
u_6, v_6
u_7, v_7

New tree



u_3, v_3
u_4, v_4
$w_3 \to m_3$
w_1
$w_4 \to m_4$
w_7
w_6
u_1, v_1
u_6, v_6
u_7, v_7
w_2
u_2, v_2
w_5
u_5, v_5

Rearranging the tree

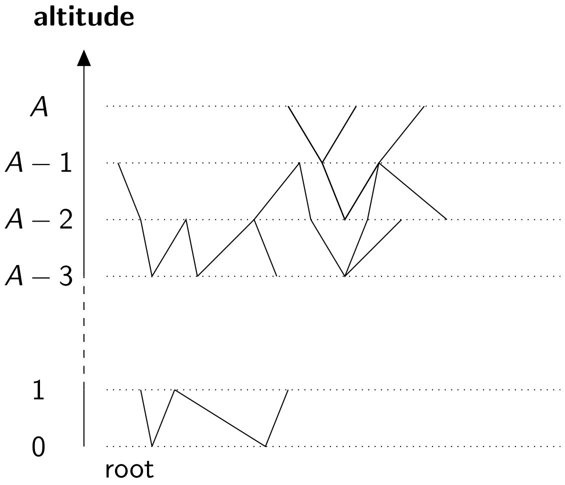
Definition

The *altitude* of a node is

- 0 if it is the root
- its father's altitude $+1$ if it is labelled $\boxed{w \to m}$
- its father's altitude -1 if it is labelled $\boxed{w}$

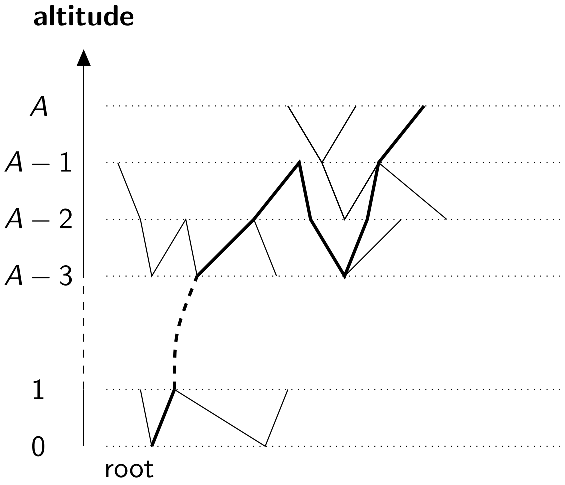
Bounding the altitude

Let A be the maximal altitude in the tree, we follow a branch reaching it.



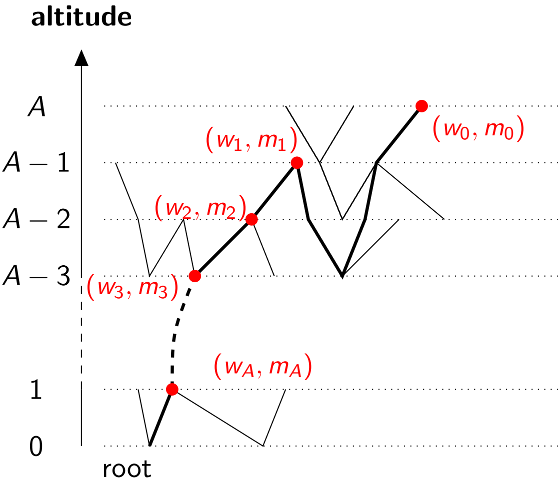
Bounding the altitude

Let A be the maximal altitude in the tree, we follow a branch reaching it.



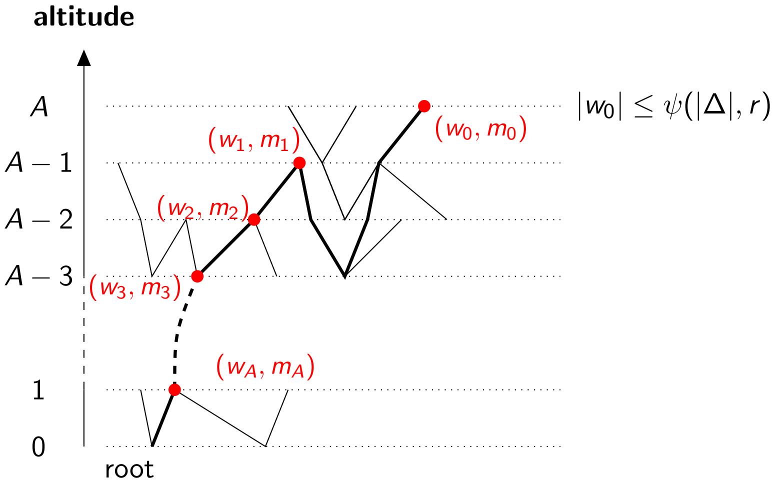
Bounding the altitude

Let A be the maximal altitude in the tree, we follow a branch reaching it.



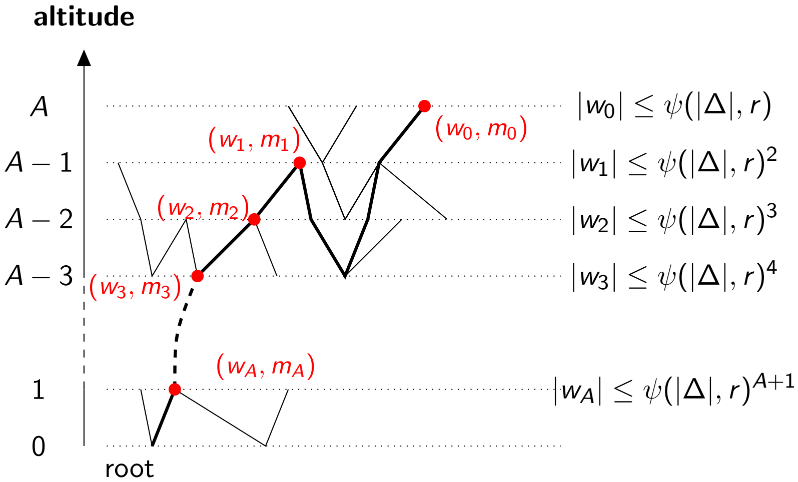
Bounding the altitude

Let A be the maximal altitude in the tree, we follow a branch reaching it.



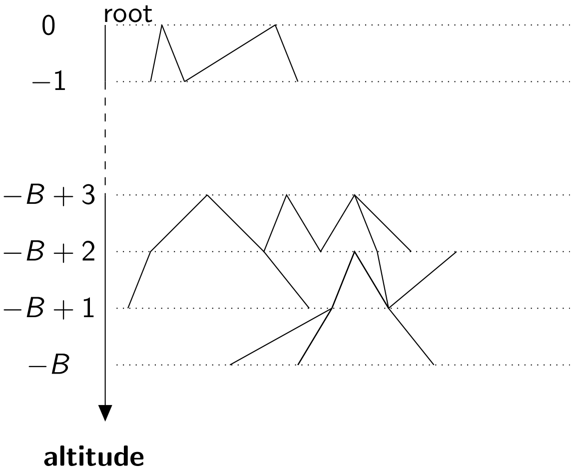
Bounding the altitude

Let A be the maximal altitude in the tree, we follow a branch reaching it.



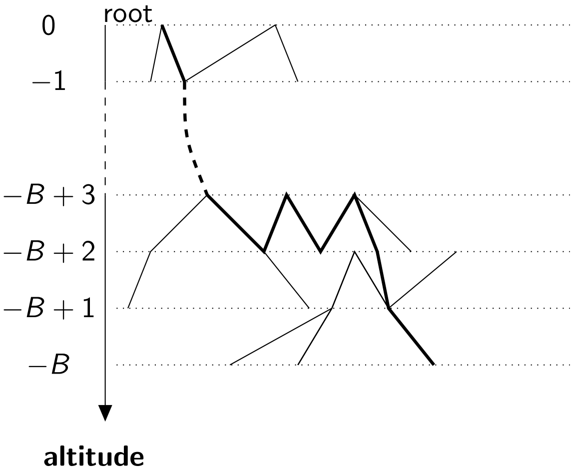
Bounding the altitude

We have bounds on the maximal altitude and the size of the root.
Let R be the size of the root, $-B$ the minimal altitude.



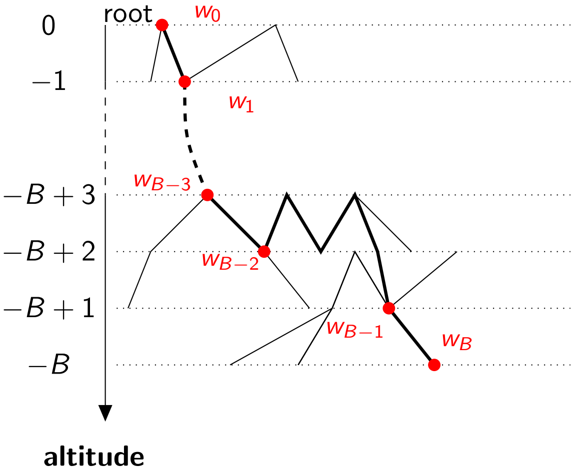
Bounding the altitude

We have bounds on the maximal altitude and the size of the root.
Let R be the size of the root, $-B$ the minimal altitude.



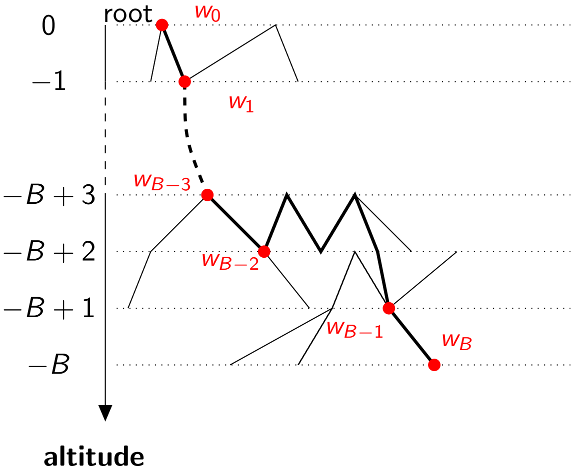
Bounding the altitude

We have bounds on the maximal altitude and the size of the root.
Let R be the size of the root, $-B$ the minimal altitude.



Bounding the altitude

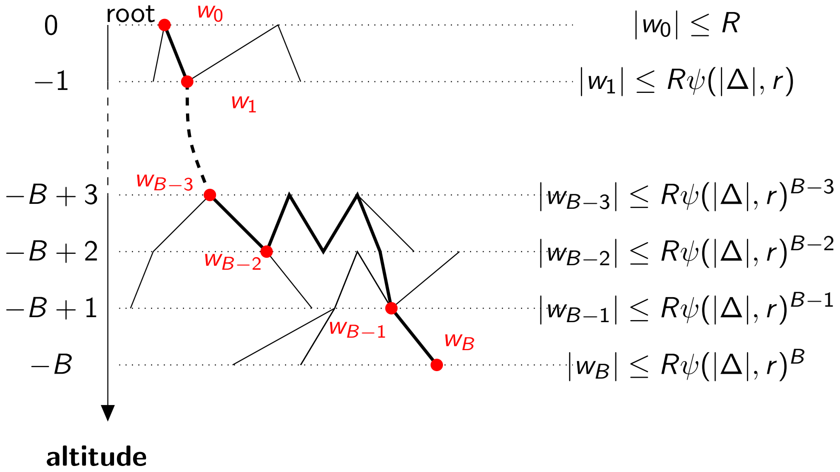
We have bounds on the maximal altitude and the size of the root.
Let R be the size of the root, $-B$ the minimal altitude.



$|w_0| \leq R$

Bounding the altitude

We have bounds on the maximal altitude and the size of the root.
Let R be the size of the root, $-B$ the minimal altitude.



Bounding the tree

We have bounds on:

- the maximal altitude
- the minimal altitude
- the size of nodes

Bounding the tree

We have bounds on:

- the maximal altitude
- the minimal altitude
- the size of nodes

We can infer bounds on

- The length of branches
- The number of children of nodes
- The tree

Decidability

We can simply enumerate irreducible trees, thus

Theorem

Cover is decidable for BNRA.

Decidability

We can simply enumerate irreducible trees, thus

Theorem

COVER is decidable for BNRA.

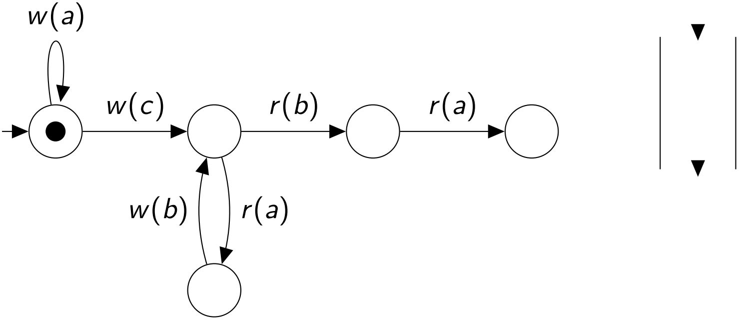
By contrast,

Theorem

TARGET is undecidable for BNRA.

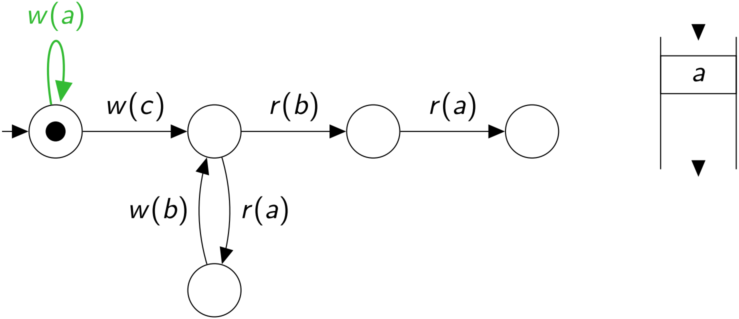
Complexity: encoding Lossy Channel Systems

Lossy Channel System = Transition system with FIFO memory + unreliable writes.



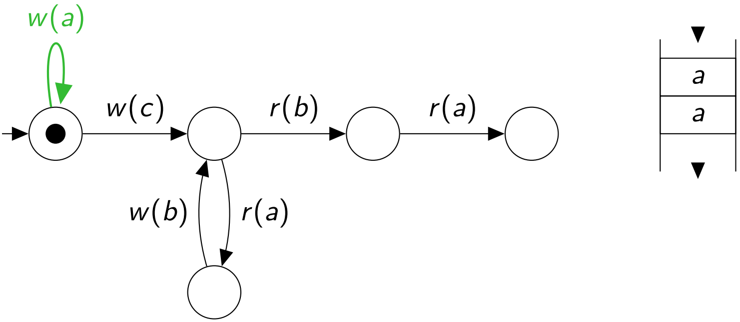
Complexity: encoding Lossy Channel Systems

Lossy Channel System = Transition system with FIFO memory + unreliable writes.



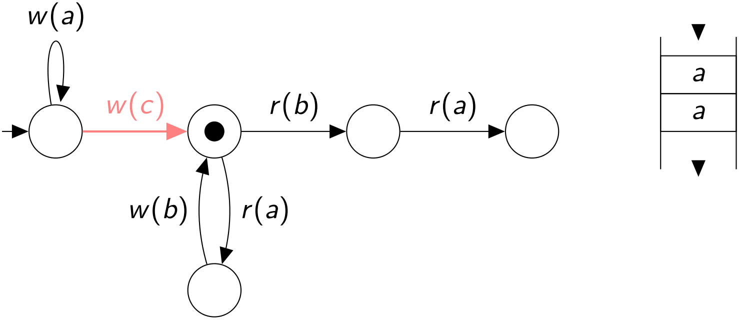
Complexity: encoding Lossy Channel Systems

Lossy Channel System = Transition system with FIFO memory + unreliable writes.



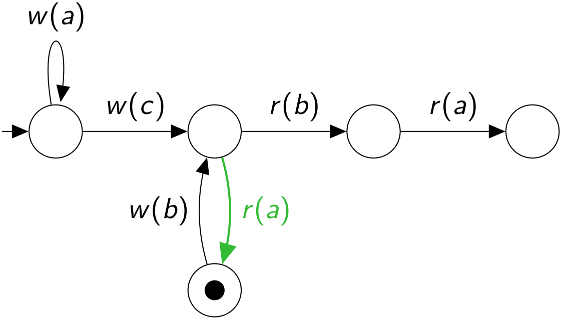
Complexity: encoding Lossy Channel Systems

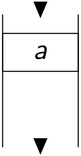
Lossy Channel System = Transition system with FIFO memory + unreliable writes.



Complexity: encoding Lossy Channel Systems

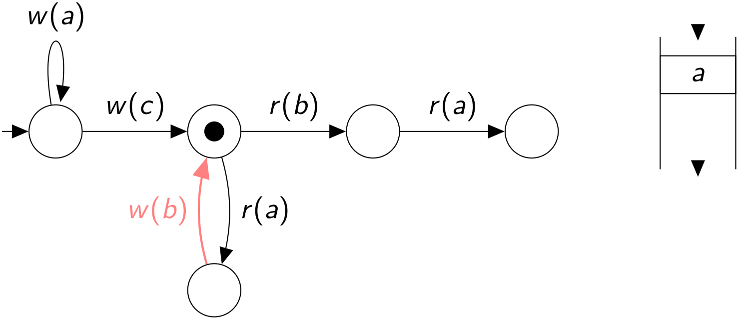
Lossy Channel System = Transition system with FIFO memory + unreliable writes.





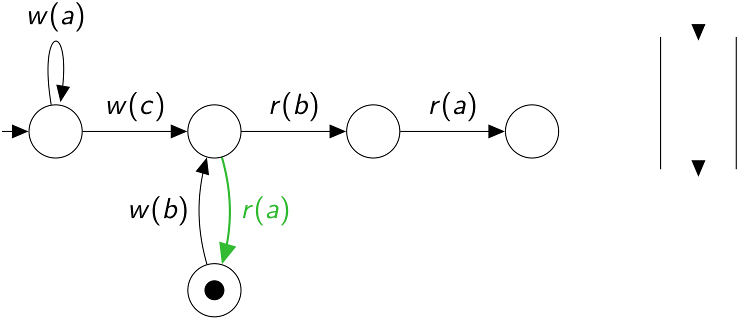
Complexity: encoding Lossy Channel Systems

Lossy Channel System = Transition system with FIFO memory + unreliable writes.



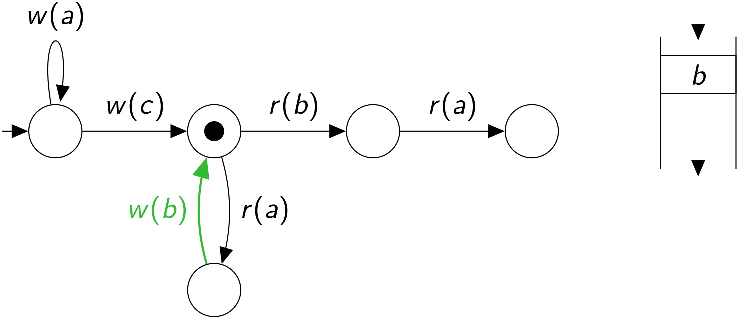
Complexity: encoding Lossy Channel Systems

Lossy Channel System = Transition system with FIFO memory + unreliable writes.



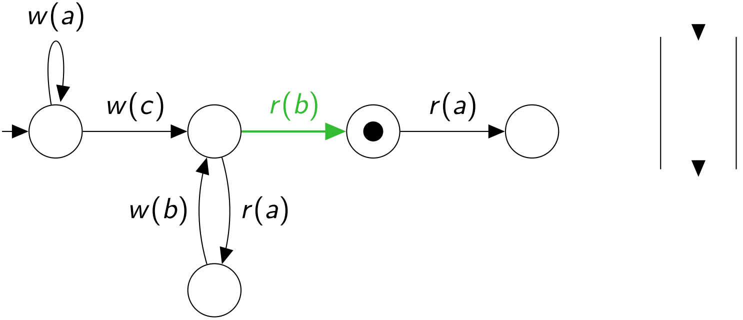
Complexity: encoding Lossy Channel Systems

Lossy Channel System = Transition system with FIFO memory + unreliable writes.



Complexity: encoding Lossy Channel Systems

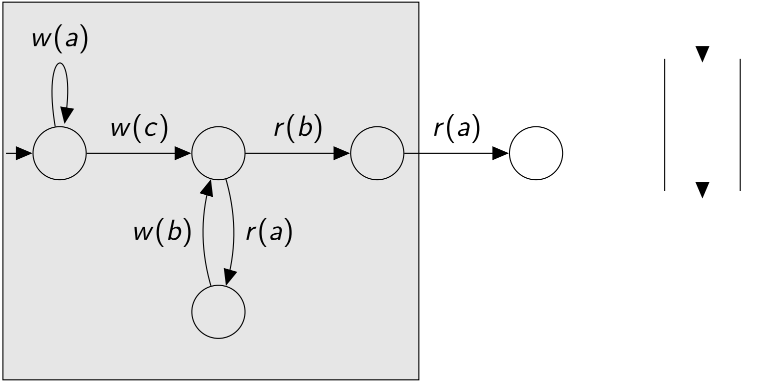
Lossy Channel System = Transition system with FIFO memory + unreliable writes.



Complexity: encoding Lossy Channel Systems

Lossy Channel System = Transition system with FIFO memory + unreliable writes.

Reachable states

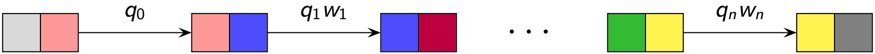


Complexity: encoding Lossy Channel Systems

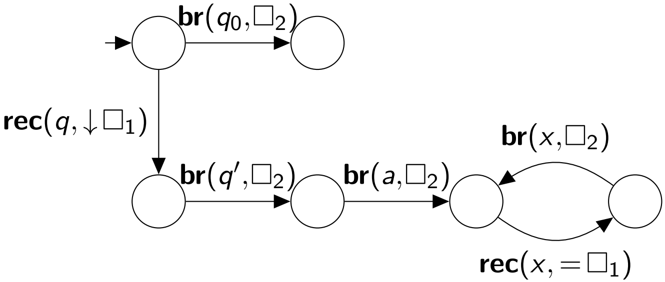
We simulate an LCS through a chain of agents that each apply a transition.

Each agent stores:

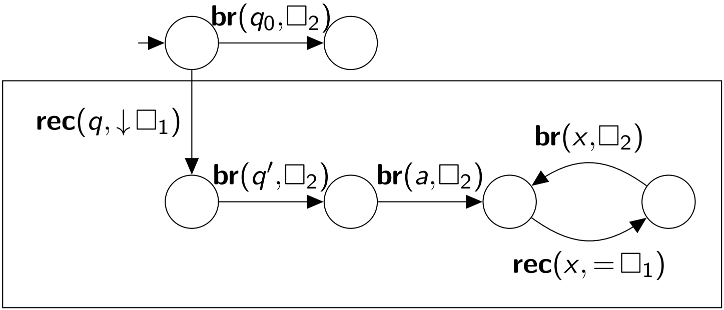
- An identifier for itself
- Its predecessor's identifier



Complexity: encoding Lossy Channel Systems



Complexity: encoding Lossy Channel Systems



For each transition $q \xrightarrow{w(a)} q'$ of the LCS

Complexity: encoding Lossy Channel Systems

$\mathbf{F}_{\omega^\omega}$ = Hyper-Ackermannian complexity class.

Theorem

LCS reachability is $\mathbf{F}_{\omega^\omega}$-hard[a].

[a] Schnoebelen, Information Processing Letters '08

Theorem

Cover in BNRA is $\mathbf{F}_{\omega^\omega}$-hard.

Complexity bounds

Length function theorem

For all finite Σ, for all $g : \mathbb{N} \to \mathbb{N}$ primitive recursive, there exists a function $F : \mathbb{N} \to \mathbb{N}$ of $\mathcal{F}_{\omega^{|\Sigma|-1}}$ such that for all $n \in \mathbb{N}$, every sequence $w_0, w_1, \ldots \in \Sigma^*$ such that $|w_i| \leq g^i(n)$ for all i has at most $F(n)$ terms.[a][b]

[a] Cichon, Tahan Bittar, Theoretical Computer Science, '98
[b] Schmitz, Schnoebelen, ICALP'11

- We can bound the tree size by a function of $\mathcal{F}_{\omega^\omega}$.

Complexity bounds

Length function theorem

For all finite Σ, for all $g : \mathbb{N} \to \mathbb{N}$ primitive recursive, there exists a function $F : \mathbb{N} \to \mathbb{N}$ of $\mathcal{F}_{\omega^{|\Sigma|-1}}$ such that for all $n \in \mathbb{N}$, every sequence $w_0, w_1, \ldots \in \Sigma^*$ such that $|w_i| \leq g^i(n)$ for all i has at most $F(n)$ terms.[a][b]

[a] Cichon, Tahan Bittar, Theoretical Computer Science, '98
[b] Schmitz, Schnoebelen, ICALP'11

- We can bound the tree size by a function of $\mathcal{F}_{\omega^\omega}$.

Theorem

$\textsc{Cover}$ in BNRA is $\mathbf{F}_{\omega^\omega}$-complete, even for a fixed number of registers $r \geq 2$.

Complexity bounds

Length function theorem

For all finite Σ, for all $g : \mathbb{N} \to \mathbb{N}$ primitive recursive, there exists a function $F : \mathbb{N} \to \mathbb{N}$ of $\mathcal{F}_{\omega^{|\Sigma|-1}}$ such that for all $n \in \mathbb{N}$, every sequence $w_0, w_1, \ldots \in \Sigma^*$ such that $|w_i| \leq g^i(n)$ for all i has at most $F(n)$ terms.[a][b]

[a] Cichon, Tahan Bittar, Theoretical Computer Science, '98
[b] Schmitz, Schnoebelen, ICALP'11

- We can bound the tree size by a function of $\mathcal{F}_{\omega^\omega}$.

Theorem

Cover in BNRA is $\mathbf{F}_{\omega^\omega}$-complete, even for a fixed number of registers $r \geq 2$.

Theorem

Cover in BNRA with one register is **NP**-complete.

Perspectives

- Add mustard!
  - Pushdown processes
  - Inequality tests $\leq$
- Less permissive communication topology?
- Bounded number of alternations between broadcast and receive?

Thank you for your attention!