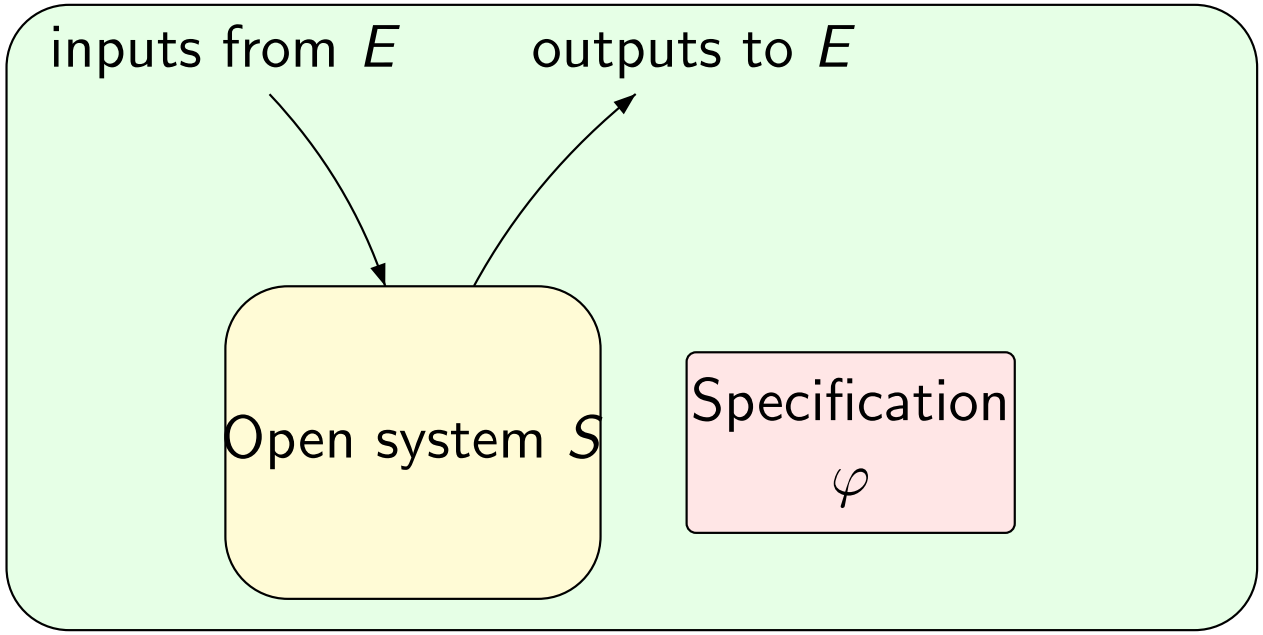
Distributed Synthesis for Well Connected Architectures

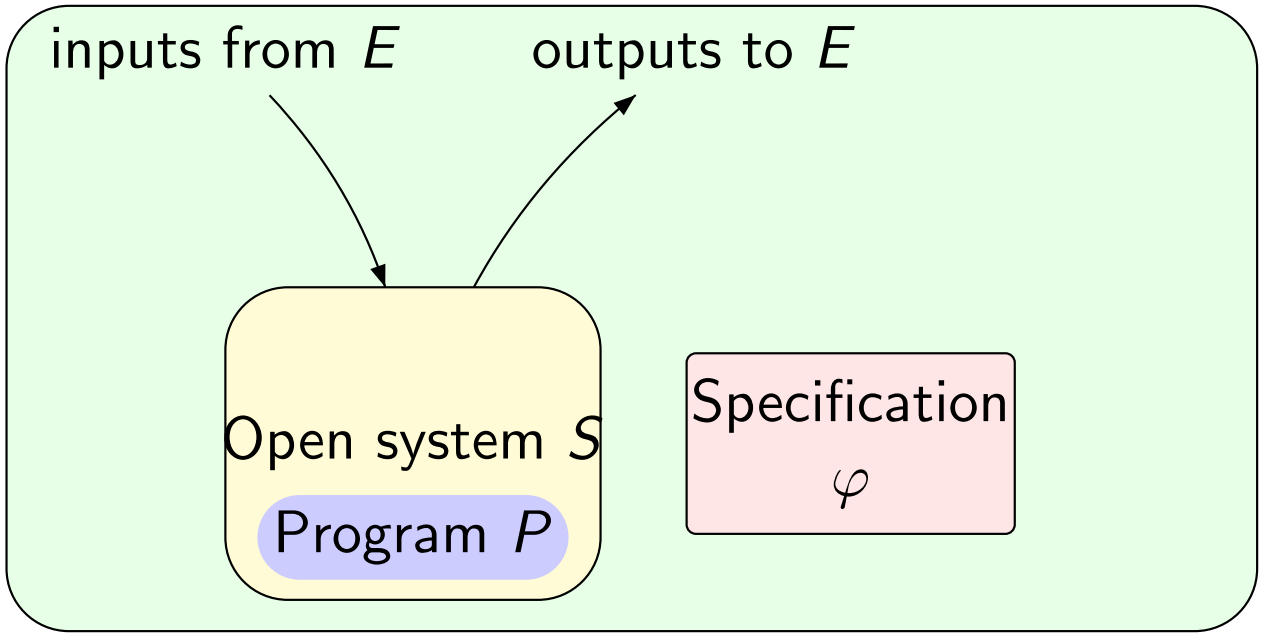
Paul Gastin, Nathalie Sznajder and Marc Zeitoun

March 13th 2006
ACI Cortos Persee Versydis

Synthesis of a reactive system



Synthesis of a reactive system

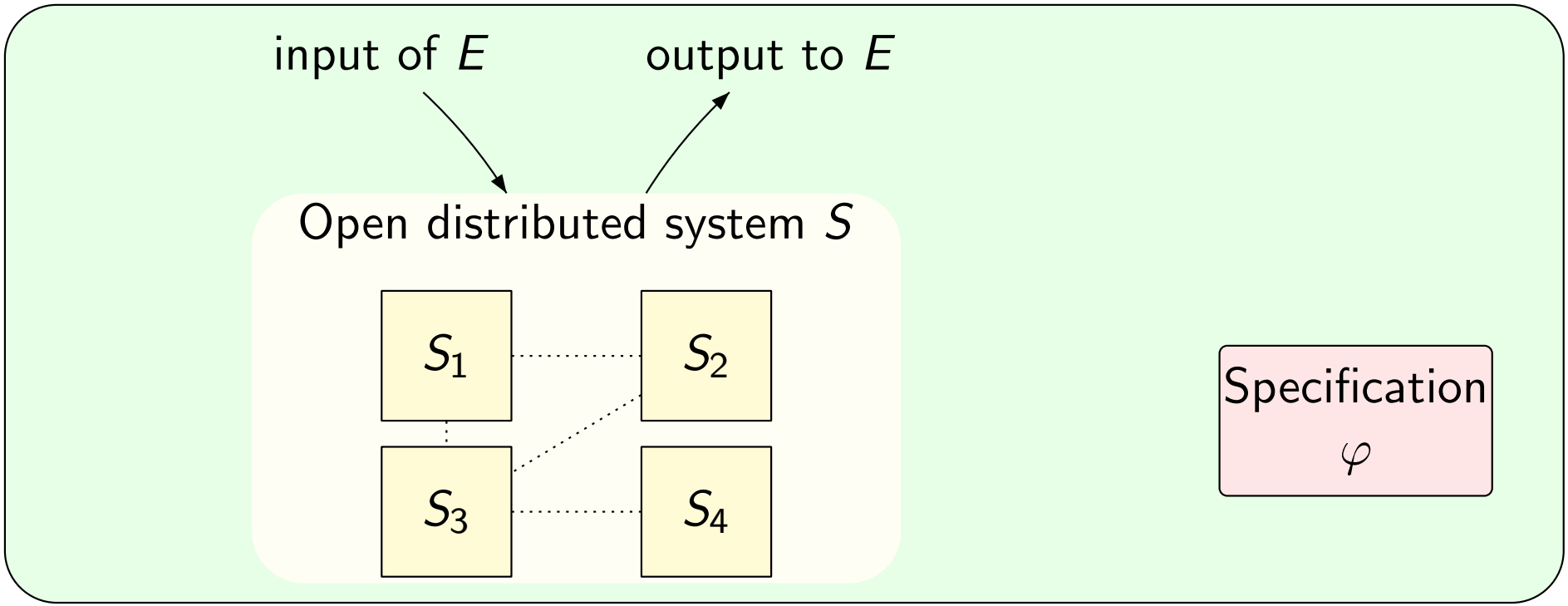


Two problems

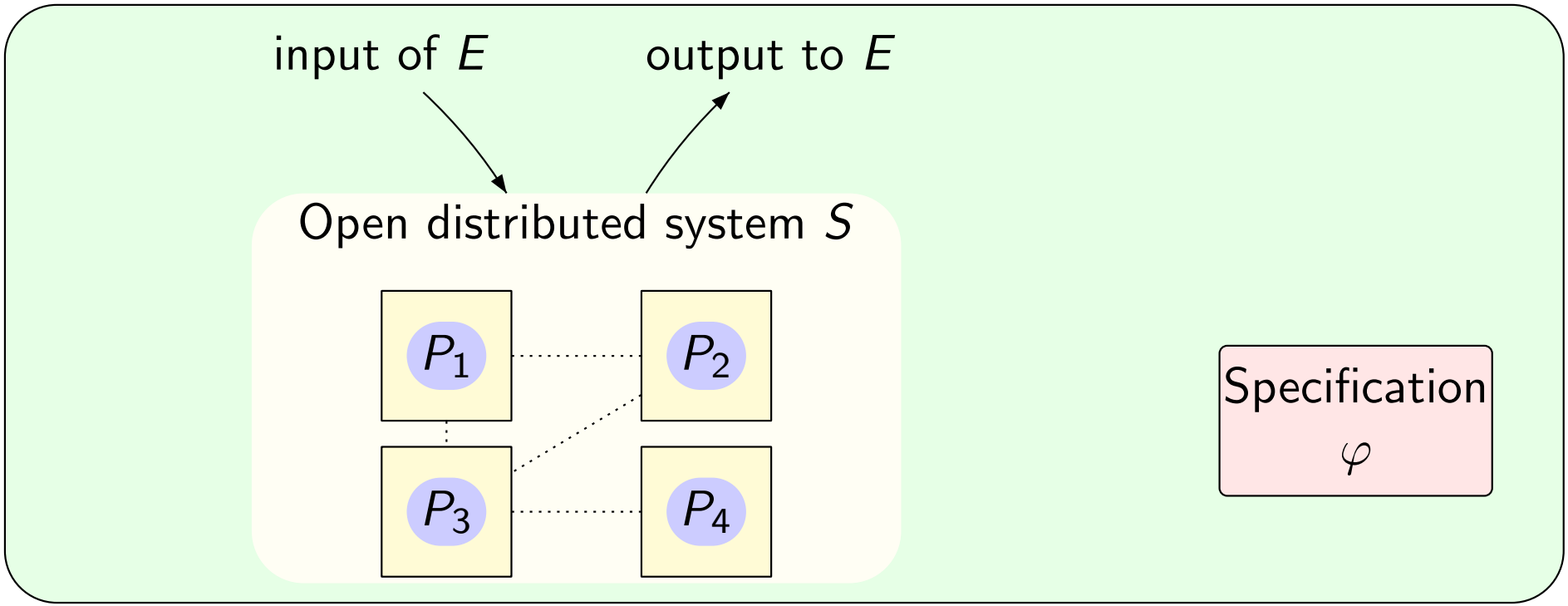
- Decide whether there exists a program st. $P||E \models \varphi, \qquad \forall E$.
- Synthesis: If so, compute such a program.

For reasonable systems and specifications, the problems are decidable.

Distributed synthesis



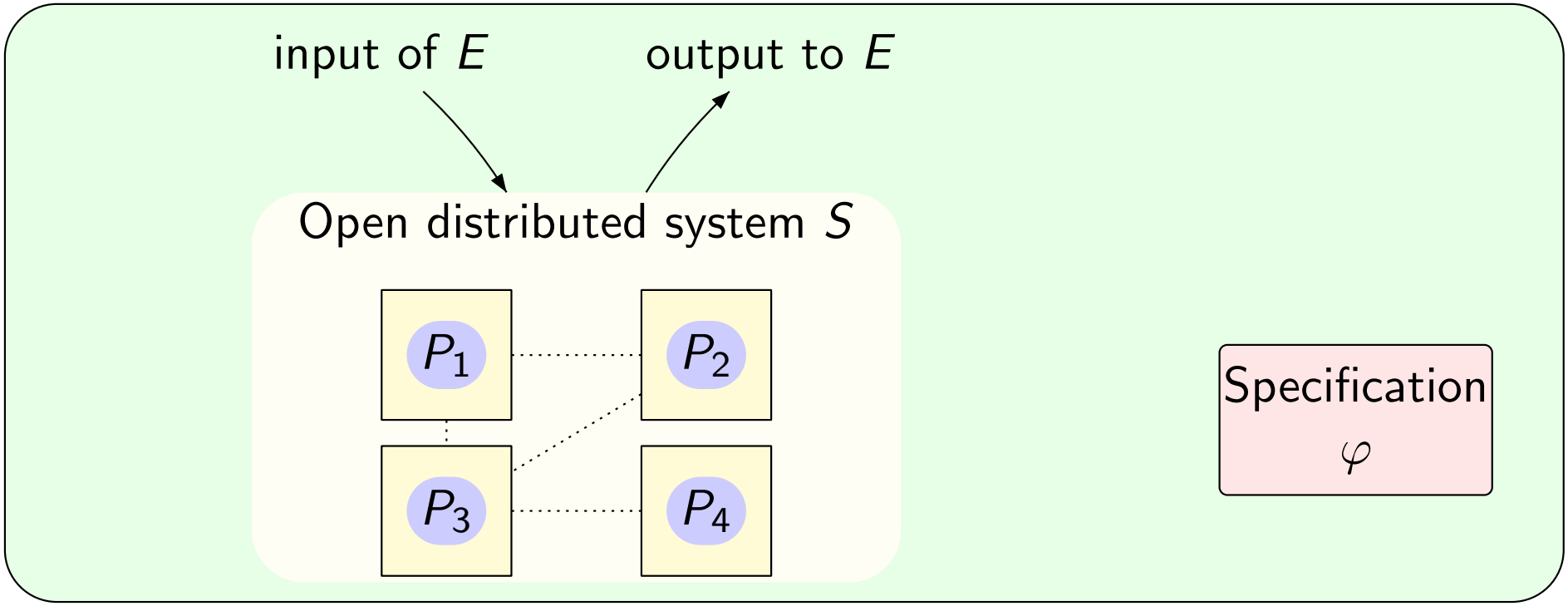
Distributed synthesis



Two problems

- Decide the existence of a distributed program such that their joint behavior $P_1||P_2||P_3||P_4||E$ satisfies φ, for all E.
- Synthesis : If it exists, compute such a distributed program.

Distributed synthesis



Two problems

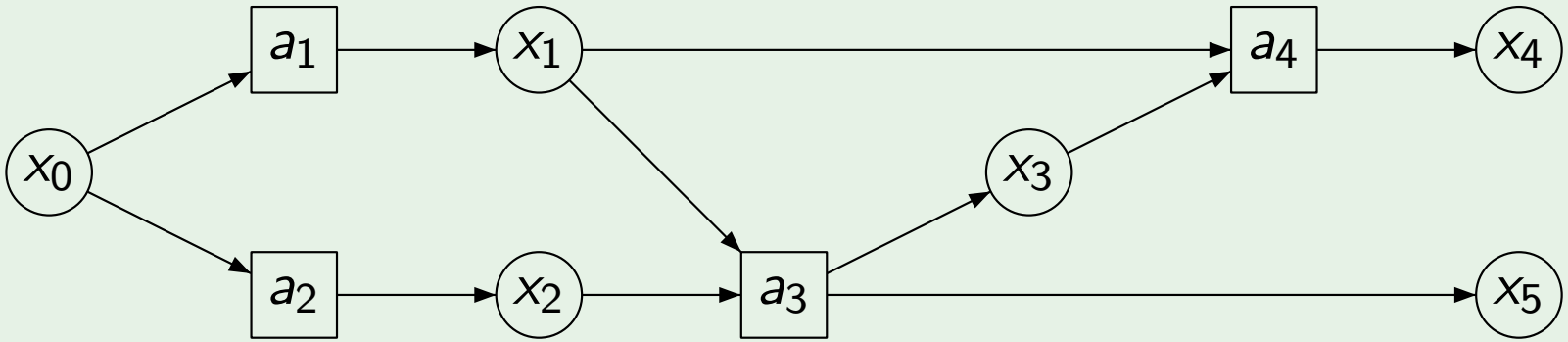
- Decide the existence of a distributed program such that their joint behavior $P_1||P_2||P_3||P_4||E$ satisfies φ, for all E.
- Synthesis : If it exists, compute such a distributed program.

Peterson-Reif 1979, Pnueli-Rosner 1990

In general, the problem is undecidable.

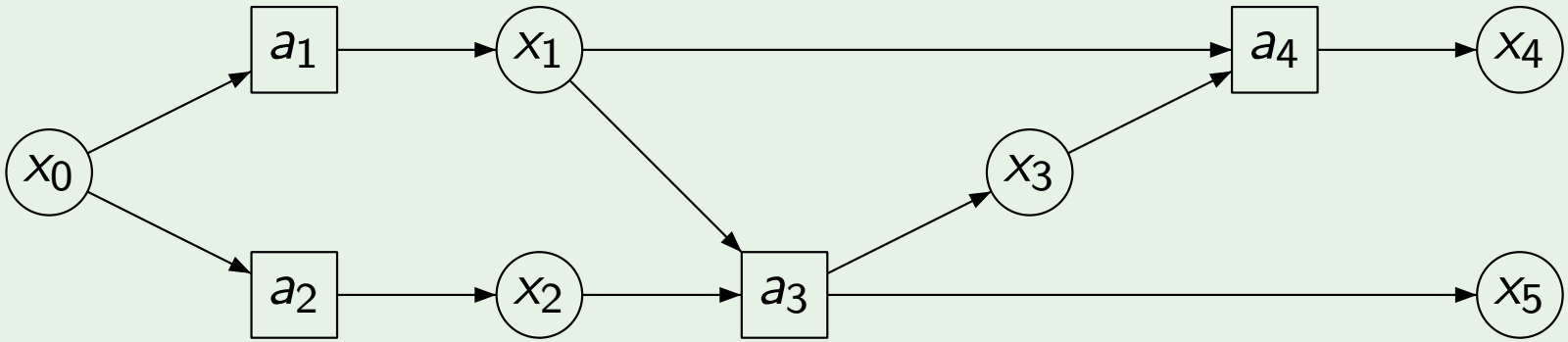
The model

Example



The model

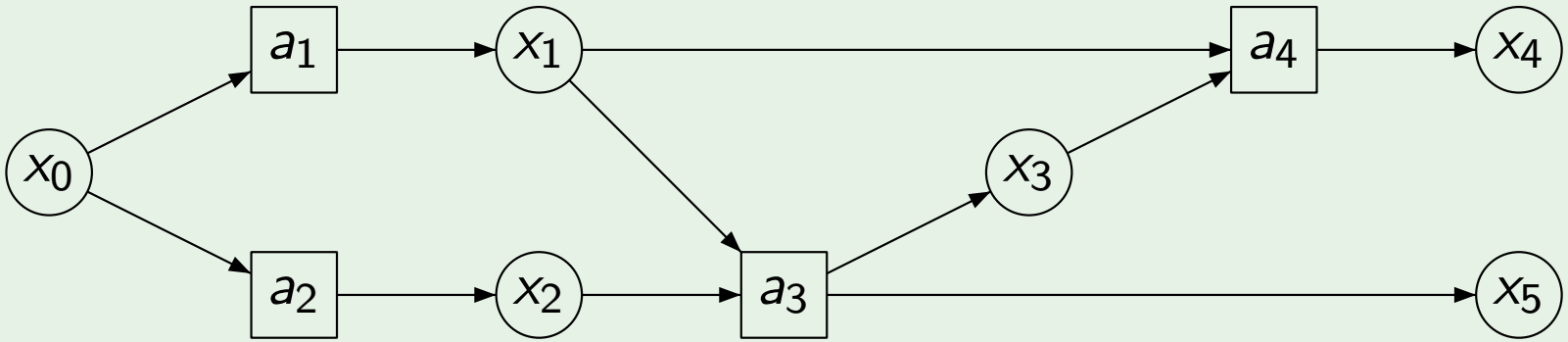
Example



- Synchronous behavior

The model

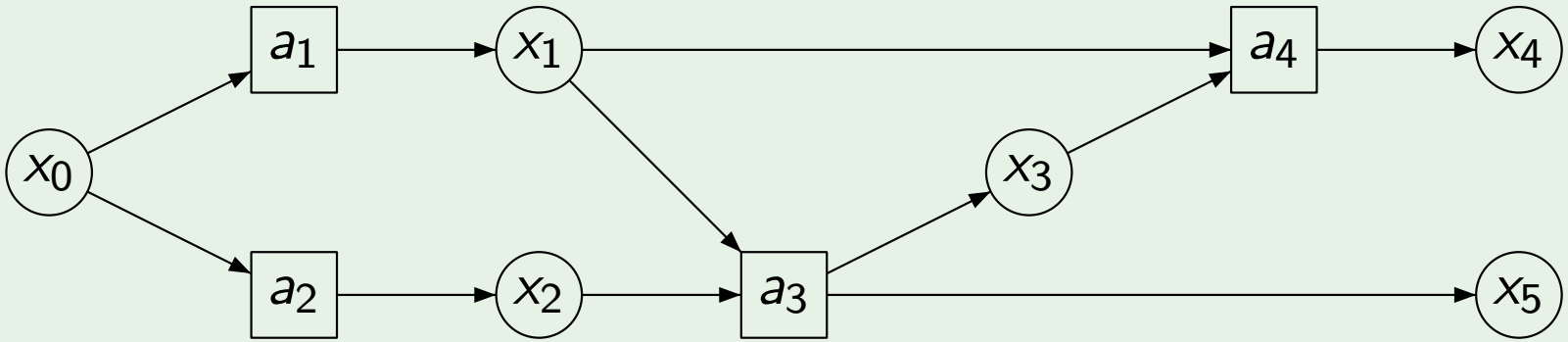
Example



- Synchronous behavior
- Strategies with local memory

The model

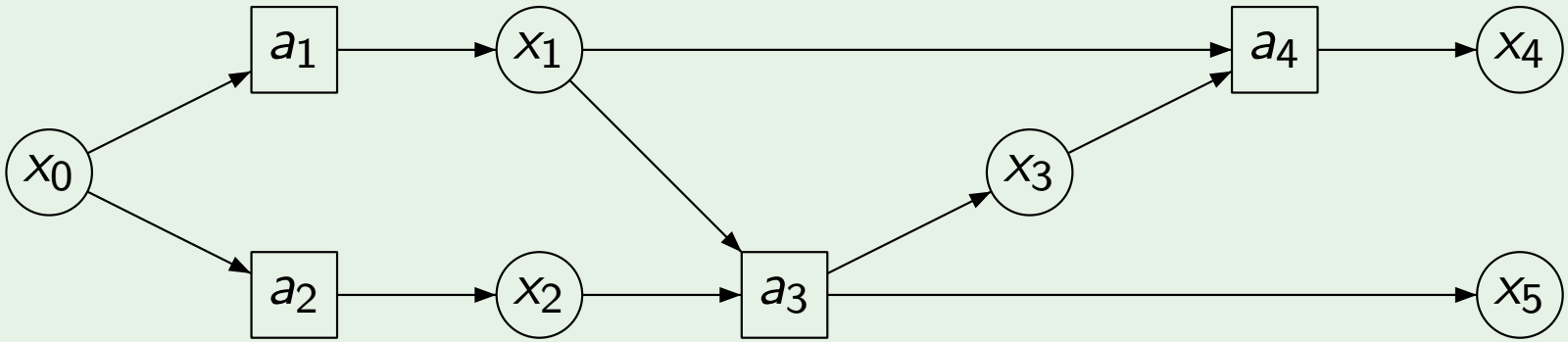
Example



- Synchronous behavior
- Strategies with local memory
- 0-delay semantics

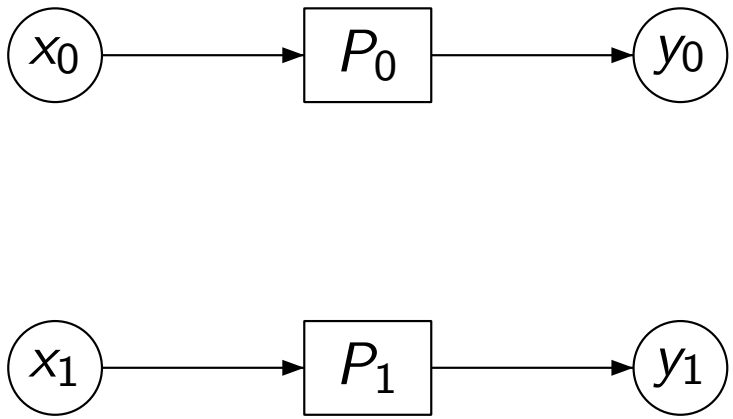
The model

Example

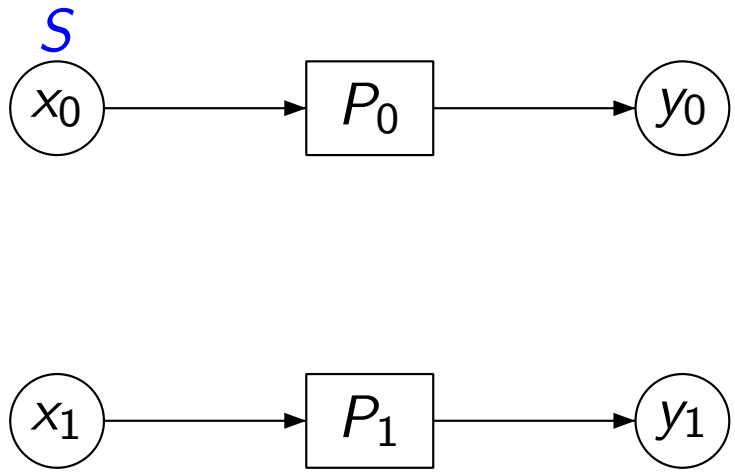


- Synchronous behavior
- Strategies with local memory
- 0-delay semantics
- Input-output specifications

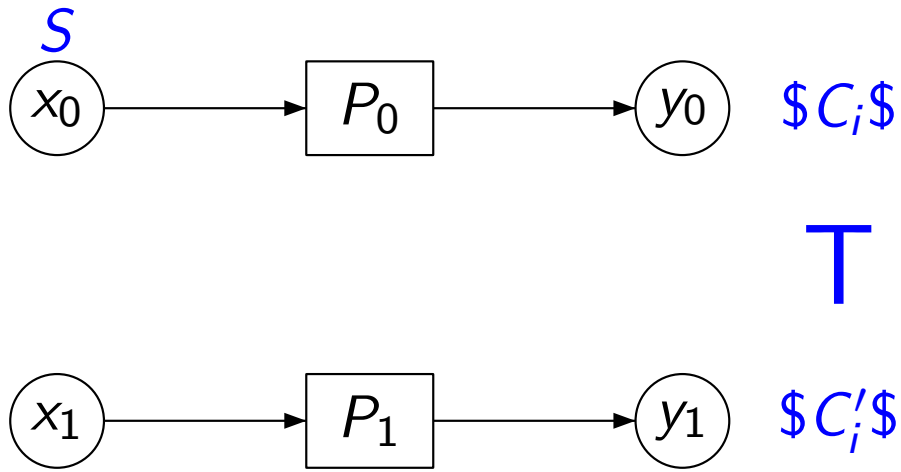
Undecidable architecture (Pnueli–Rosner '90)



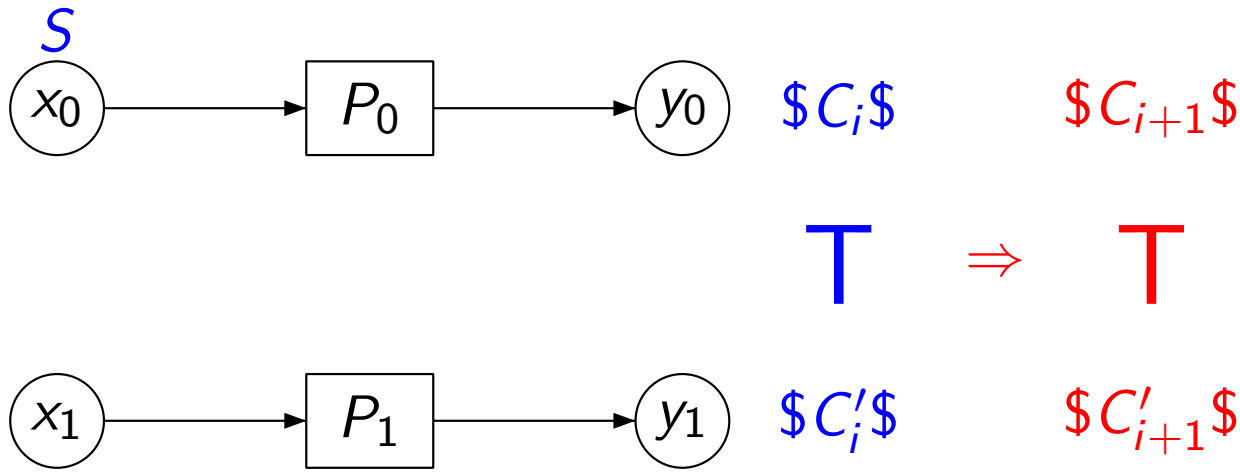
Undecidable architecture (Pnueli–Rosner '90)



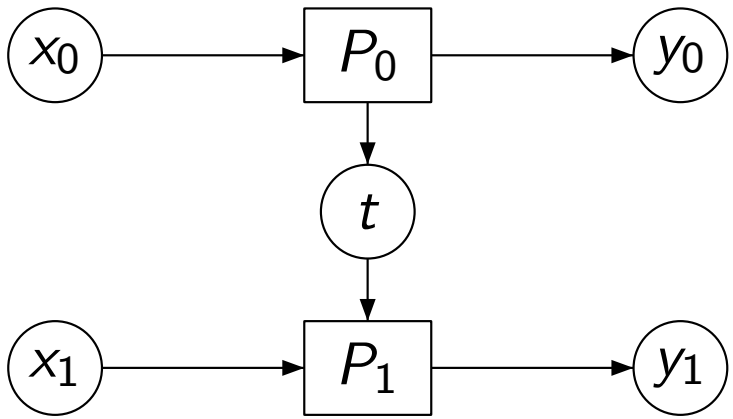
Undecidable architecture (Pnueli–Rosner '90)



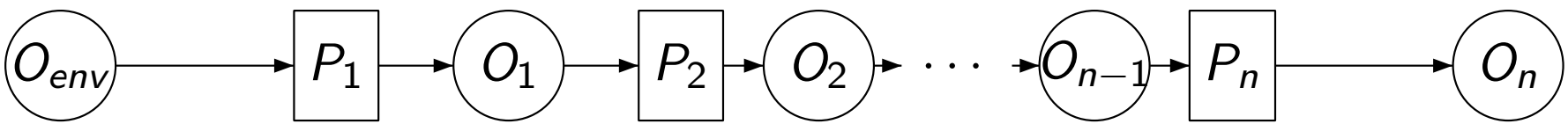
Undecidable architecture (Pnueli–Rosner '90)



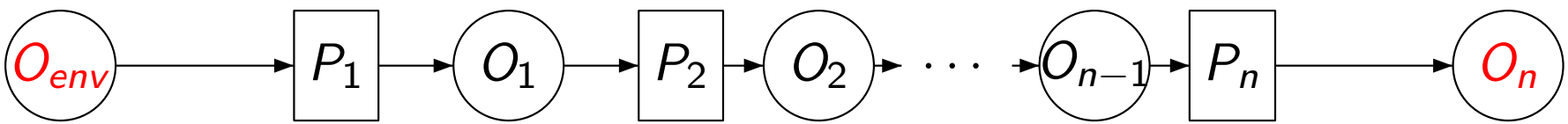
Decidable architecture



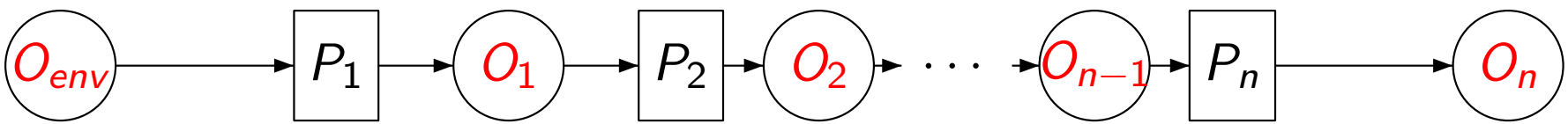
Pipe-line decidable for global specifications (Kupferman–Vardi '01)



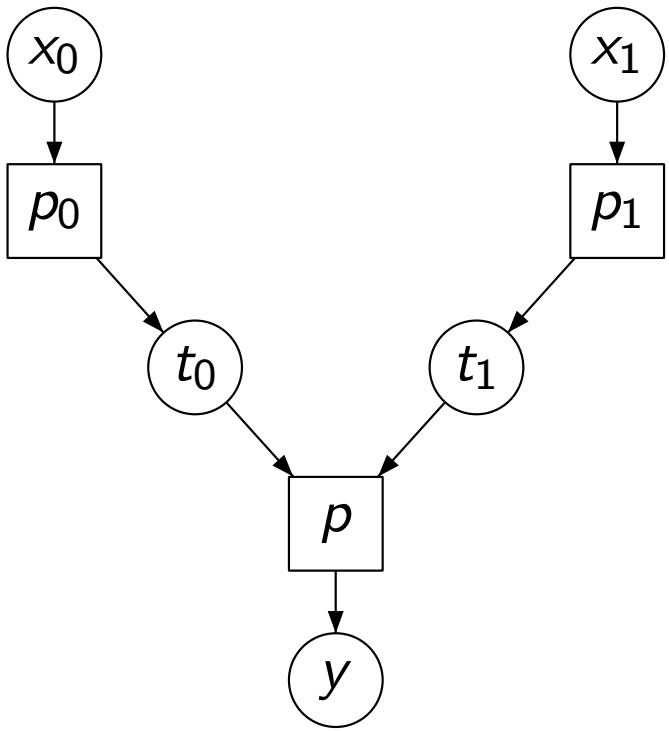
Pipe-line decidable for global specifications (Kupferman–Vardi '01)



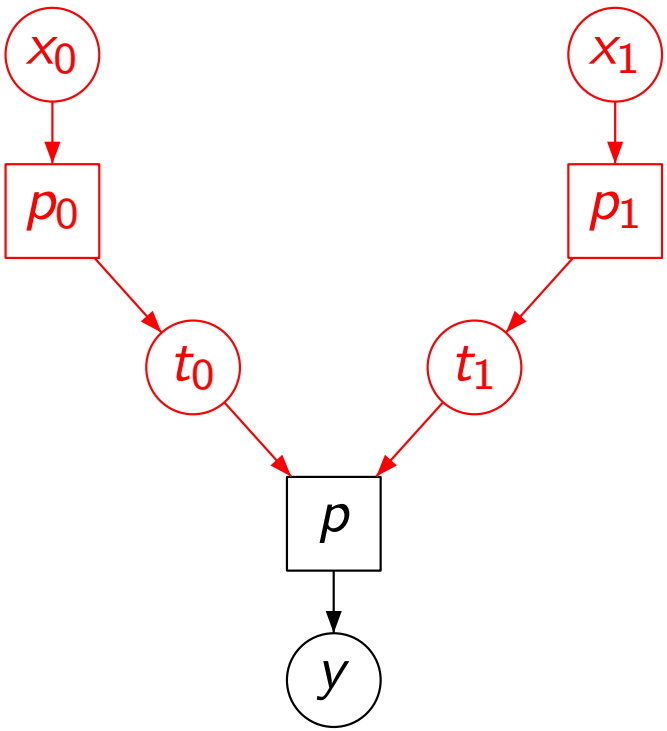
Pipe-line decidable for global specifications (Kupferman–Vardi '01)

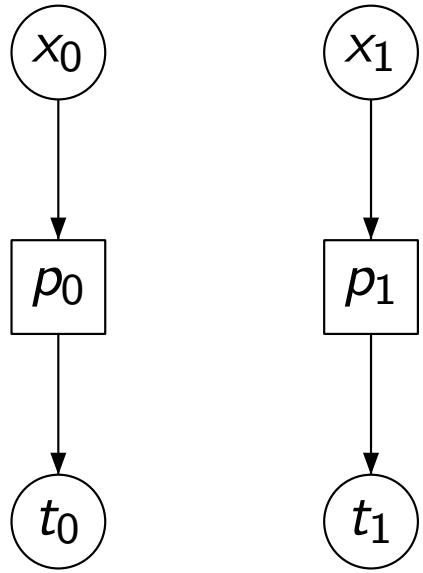


Information fork criterion (Finkbeiner–Schewe '05)



Information fork criterion (Finkbeiner–Schewe '05)





Outline

Outline

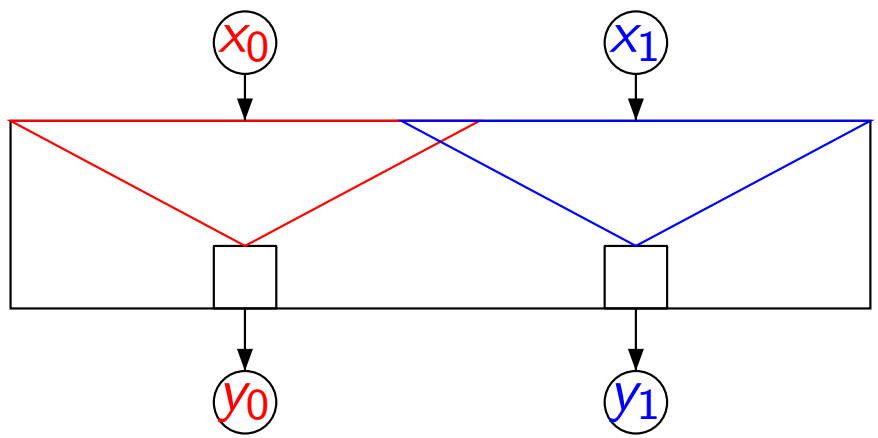
Uncomparable information yields undecidability

Definition

For an output variable v, View(v) is the set of input variables u such that v is accessible from u.

Definition

An architecture has uncomparable information if there exist x,y output variables such that $\text{View}(x) \setminus \text{View}(y) \neq \emptyset$ and $\text{View}(y) \setminus \text{View}(x) \neq \emptyset$. Otherwise it is said to have preordered information.



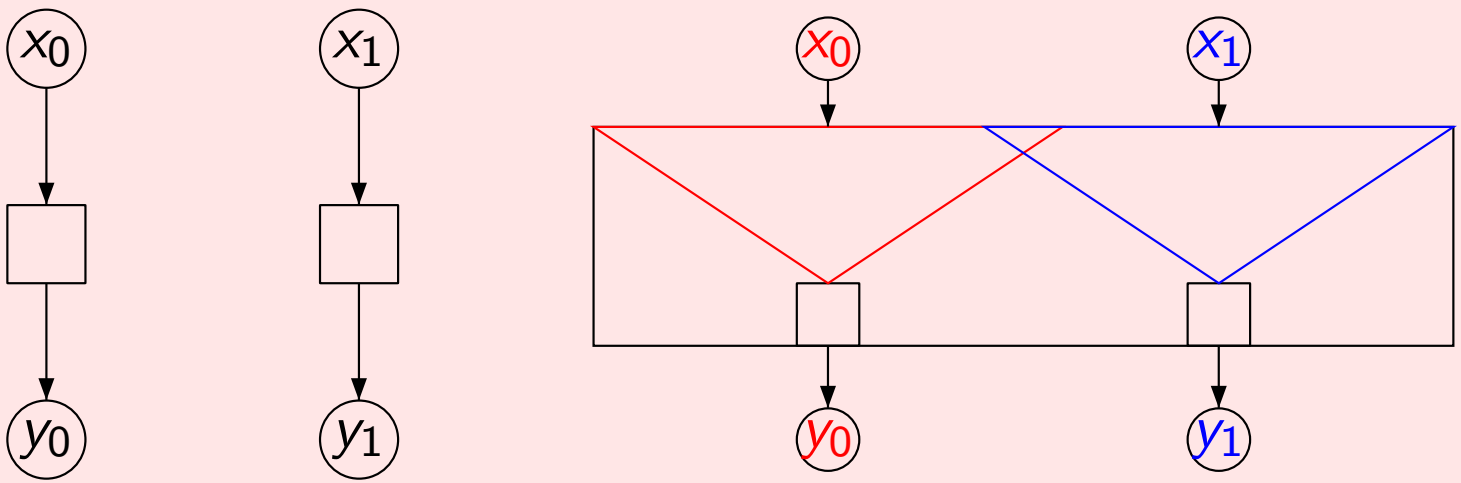
Uncomparable information yields undecidability

Theorem

Architectures with uncomparable information are undecidable for LTL or CTL input-output specifications.

Proof

LTL specifications :



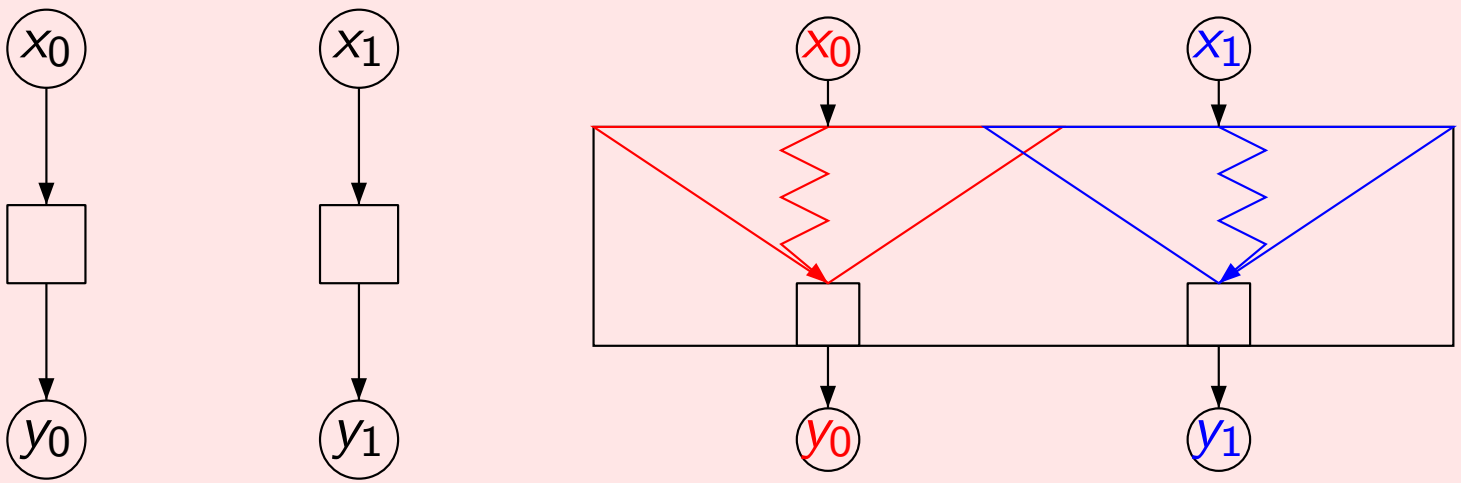
Uncomparable information yields undecidability

Theorem

Architectures with uncomparable information are undecidable for LTL or CTL input-output specifications.

Proof

LTL specifications :



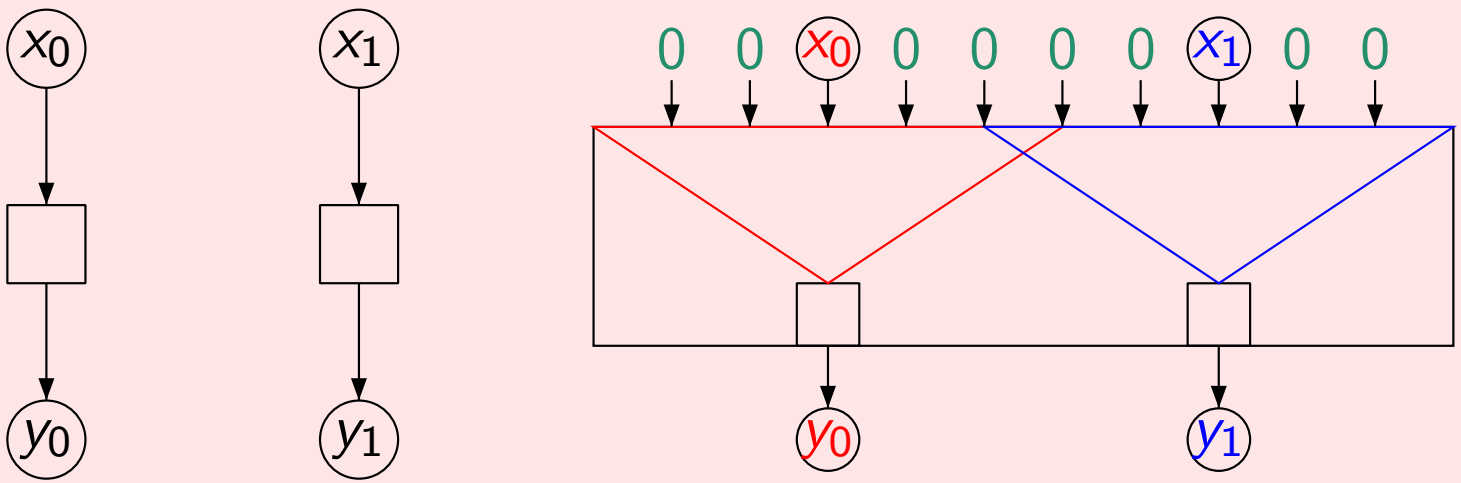
Uncomparable information yields undecidability

Theorem

Architectures with uncomparable information are undecidable for LTL or CTL input-output specifications.

Proof

LTL specifications :

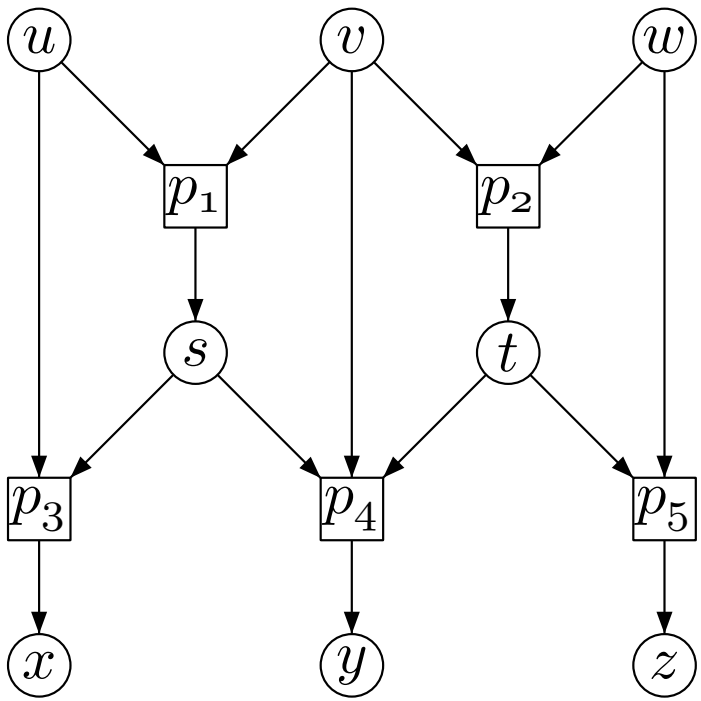


Outline

Uniformly well connected architectures

Definition

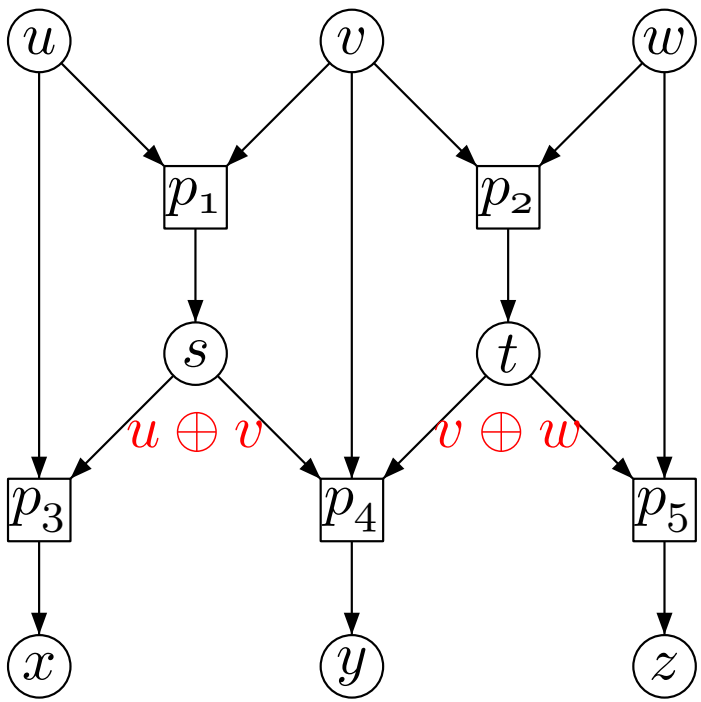
An architecture is uniformly well connected if there is a uniform way to route variables in View(v) to v for each output variable v.



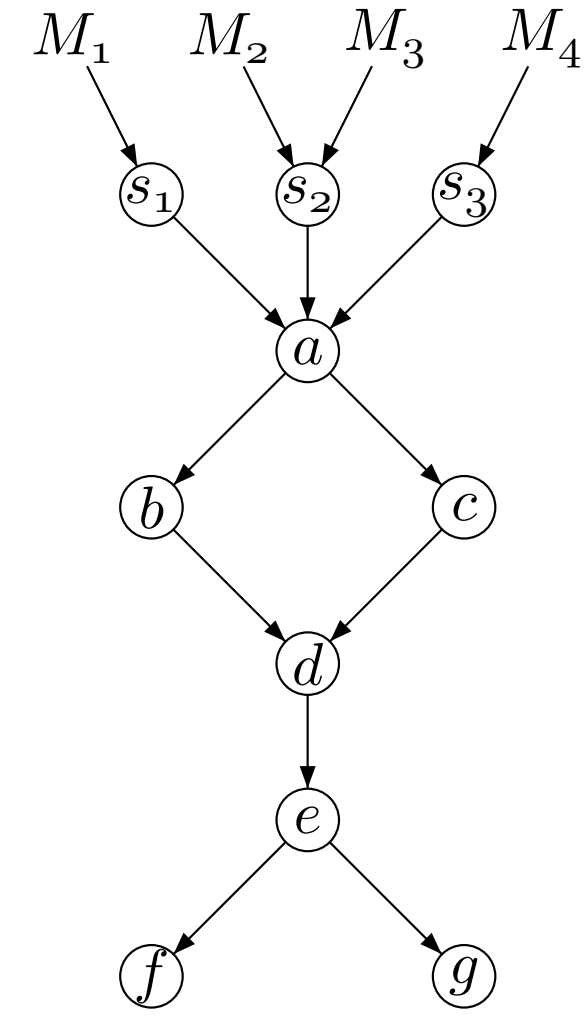
Uniformly well connected architectures

Definition

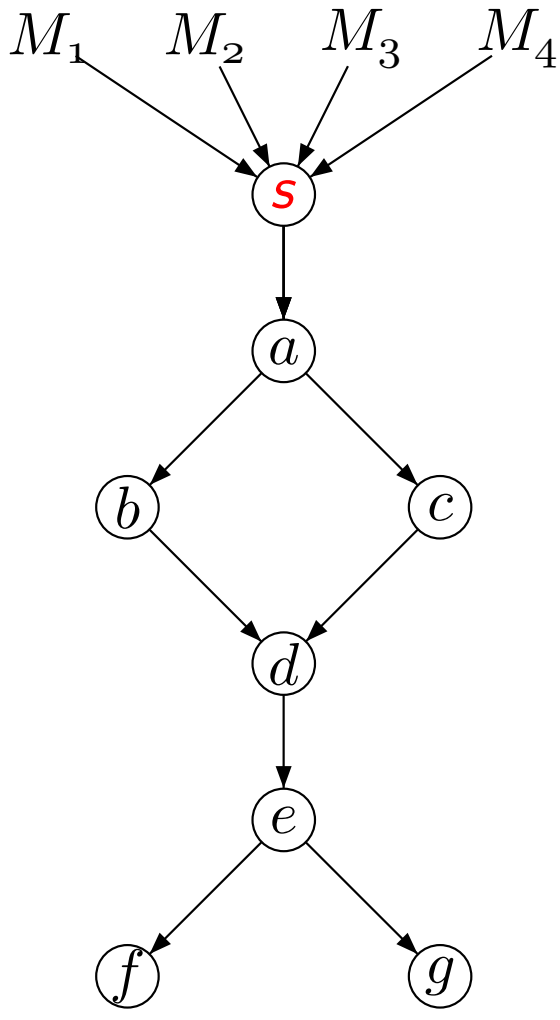
An architecture is uniformly well connected if there is a uniform way to route variables in View(v) to v for each output variable v.



Network Information Flow



Network Information Flow : Multicast



demands
M_1, M_2, M_3, M_4

demands
M_1, M_2, M_3, M_4

Relations between the two problems

Theorem

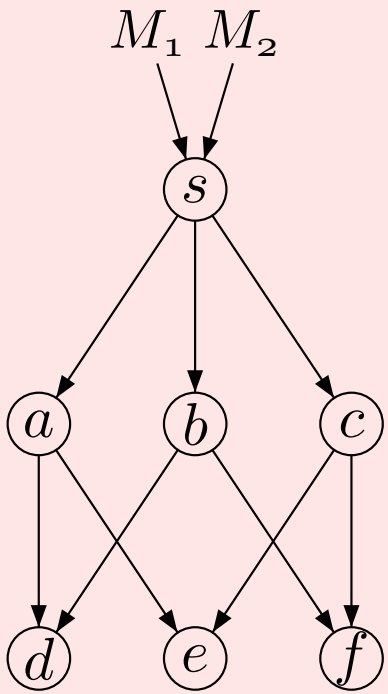
The multicast reduces to checking uniform well connectedness.

Relations between the two problems

Theorem

The multicast reduces to checking uniform well connectedness.

Proof.

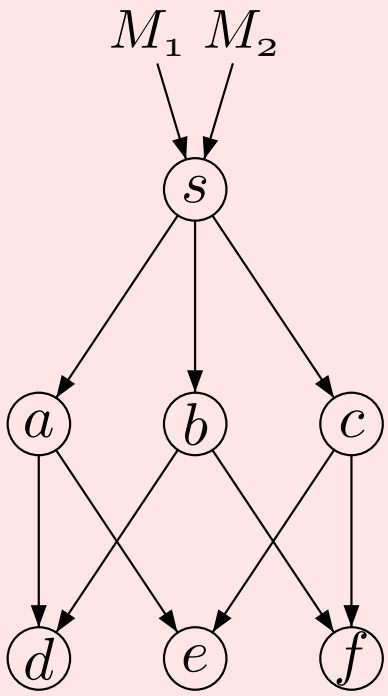


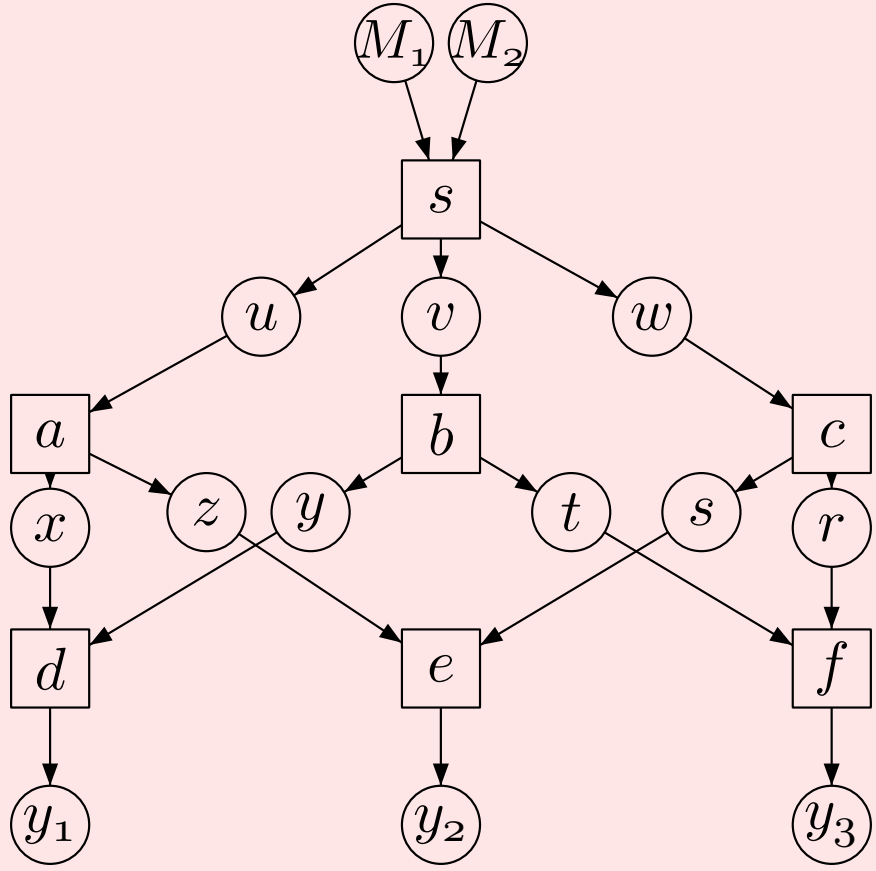
Relations between the two problems

Theorem

The multicast reduces to checking uniform well connectedness.

Proof.





Relations between the two problems

Theorem

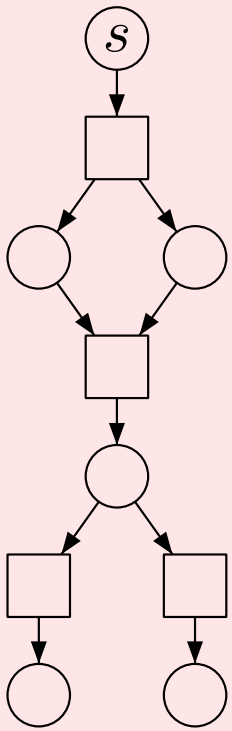
Checking uniform well connectedness reduces to the network information flow.

Relations between the two problems

Theorem

Checking uniform well connectedness reduces to the network information flow.

Proof.

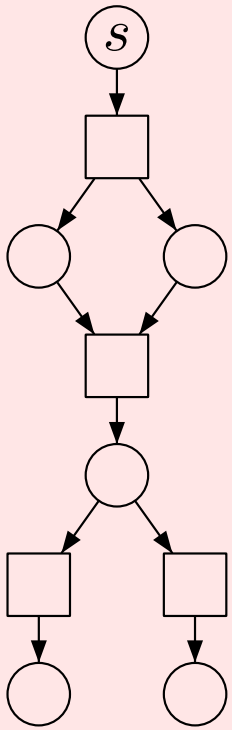


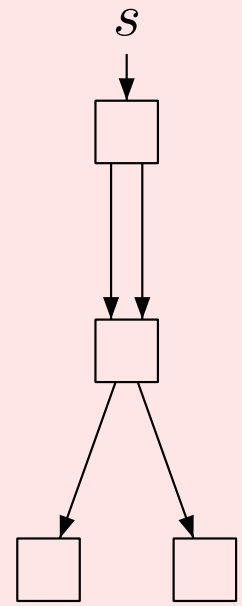
Relations between the two problems

Theorem

Checking uniform well connectedness reduces to the network information flow.

Proof.





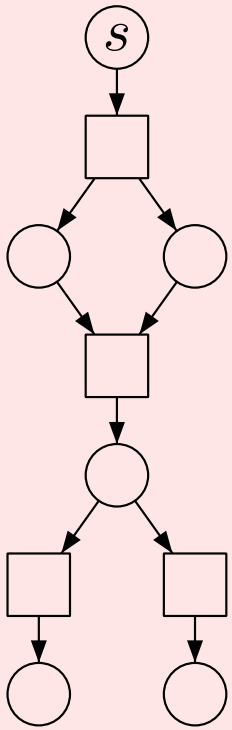
demands s demands s

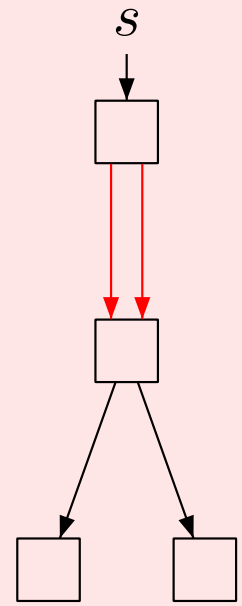
Relations between the two problems

Theorem

Checking uniform well connectedness reduces to the network information flow.

Proof.





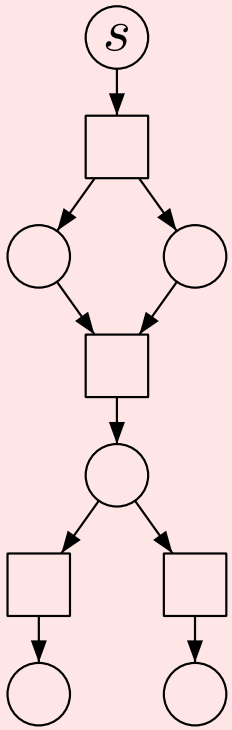
demands s demands s

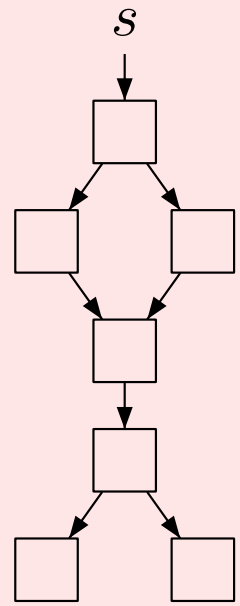
Relations between the two problems

Theorem

Checking uniform well connectedness reduces to the network information flow.

Proof.





demands s demands s

Complexity

Rasala Lehman-Lehman 2004

Multicast whith alphabet size $q = p^k$ (where p is prime) is NP-hard.

Complexity

Rasala Lehman-Lehman 2004

Multicast whith alphabet size $q = p^k$ (where p is prime) is NP-hard.

Theorem

Checking whether a given architecture is uniformly well connected is NP-complete.

Complexity

Rasala Lehman-Lehman 2004

Multicast whith alphabet size $q = p^k$ (where p is prime) is NP-hard.

Theorem

Checking whether a given architecture is uniformly well connected is NP-complete.

Proof.

It is trivially NP. Reduction from multicast gives NP-hardness.

Decidability criterion for uniformly well connected architectures

Theorem

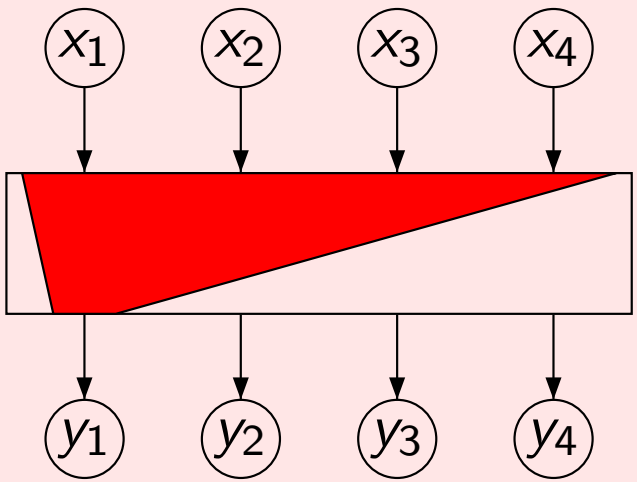
Architectures with preordered information are decidable for CTL specifications.*

Decidability criterion for uniformly well connected architectures

Theorem

Architectures with preordered information are decidable for CTL specifications.*

Proof.

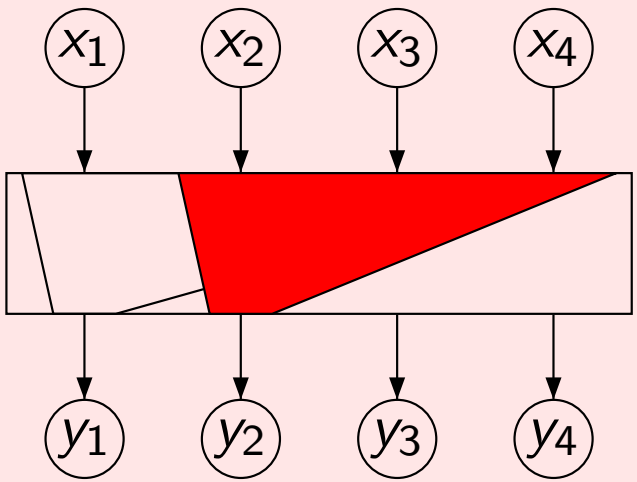


Decidability criterion for uniformly well connected architectures

Theorem

Architectures with preordered information are decidable for CTL specifications.*

Proof.

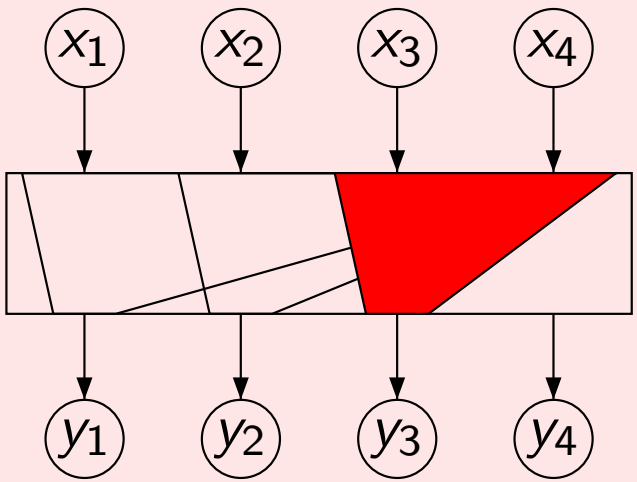


Decidability criterion for uniformly well connected architectures

Theorem

Architectures with preordered information are decidable for CTL specifications.*

Proof.

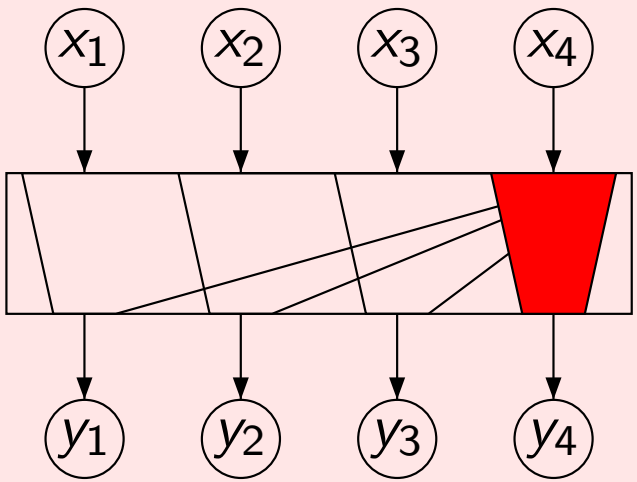


Decidability criterion for uniformly well connected architectures

Theorem

Architectures with preordered information are decidable for CTL specifications.*

Proof.

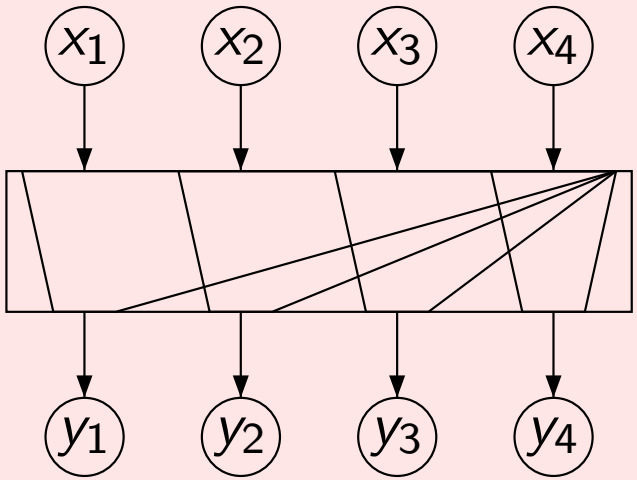


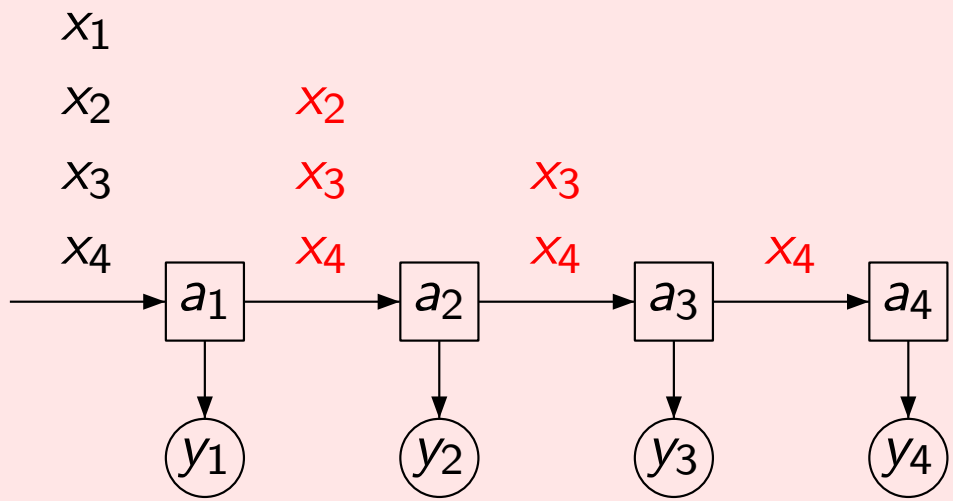
Decidability criterion for uniformly well connected architectures

Theorem

Architectures with preordered information are decidable for CTL specifications.*

Proof.





Theorem: Kupferman-Vardi (LICS'01)

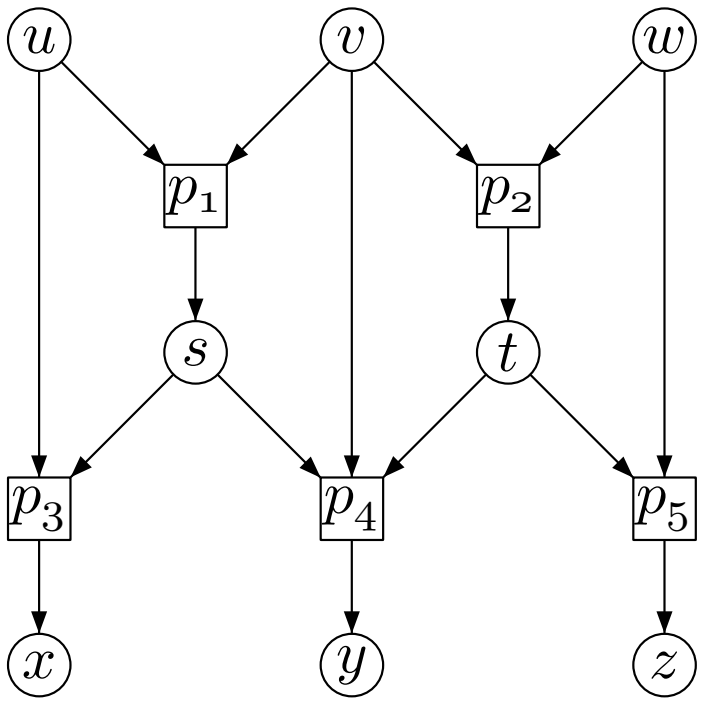
The synthesis problem with local strategies is decidable for pipeline architectures and CTL* specifications (or tree-automata specifications) on all variables.

Outline

Well connected architectures

Definition

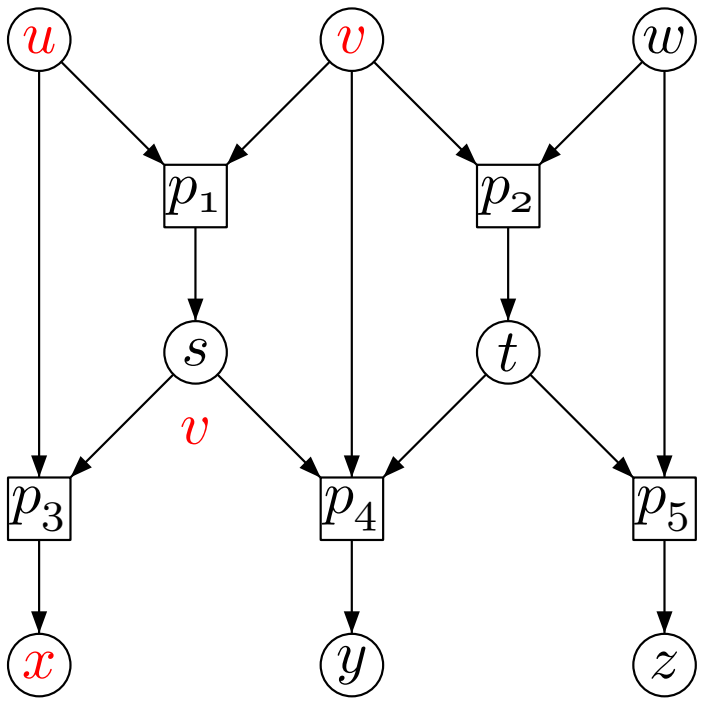
An architecture is well connected if, for each output variable y, the subarchitecture formed by $E^{*-1}(y)$ is uniformly well connected.



Well connected architectures

Definition

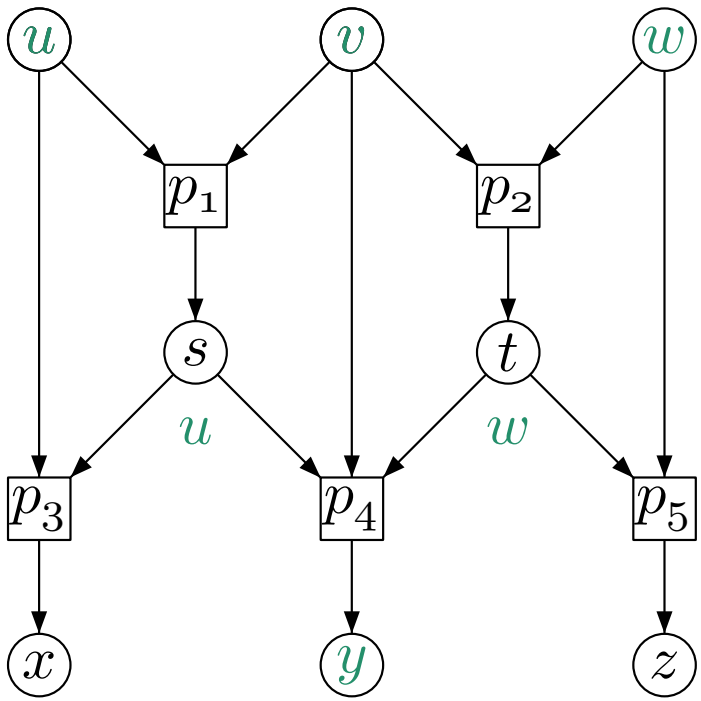
An architecture is well connected if, for each output variable y, the subarchitecture formed by $E^{*-1}(y)$ is uniformly well connected.



Well connected architectures

Definition

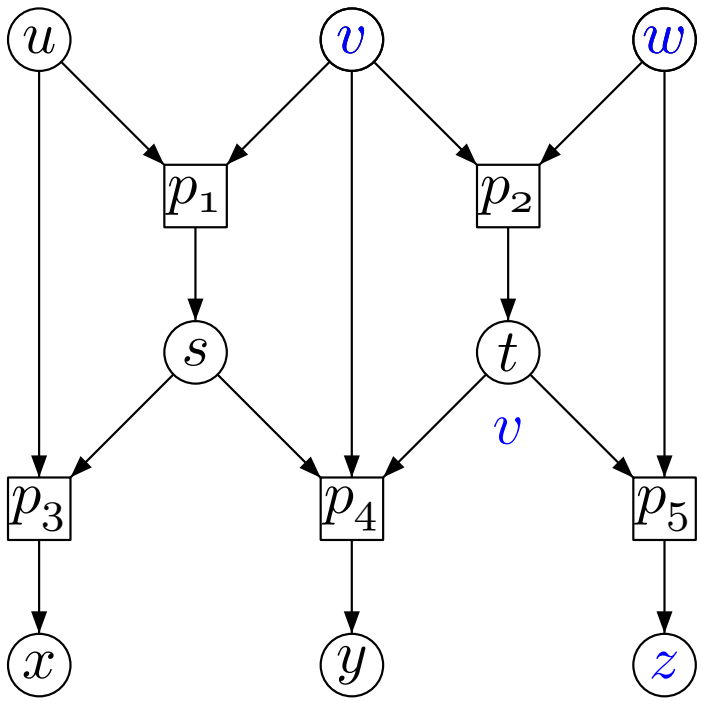
An architecture is well connected if, for each output variable y, the subarchitecture formed by $E^{*-1}(y)$ is uniformly well connected.



Well connected architectures

Definition

An architecture is well connected if, for each output variable y, the subarchitecture formed by $E^{*-1}(y)$ is uniformly well connected.



Well connected architectures

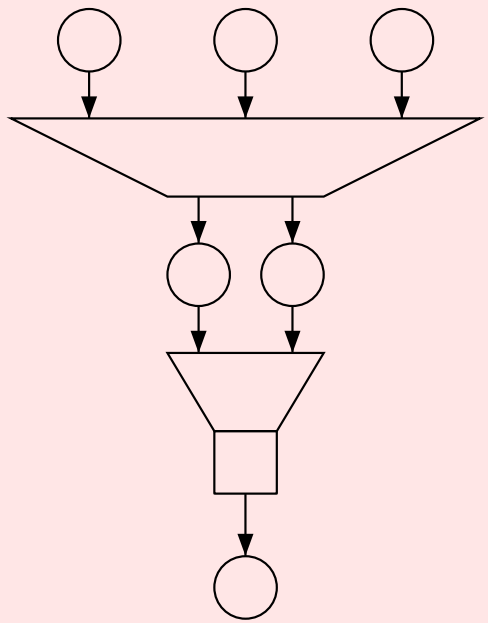
Theorem

One can decide whether an architecture is well-connected in polynomial time.

Well connected architectures

Theorem

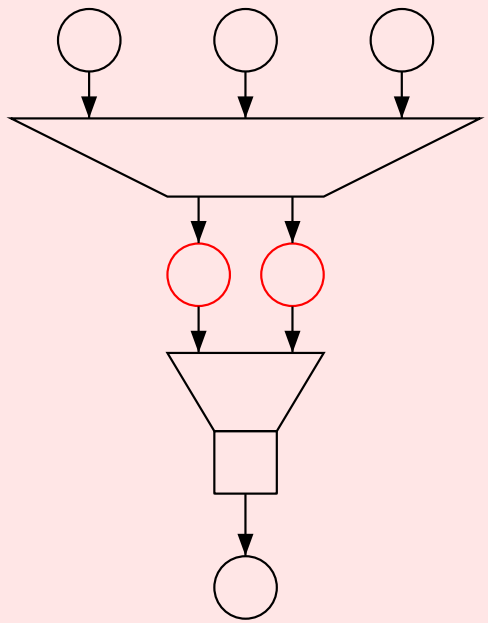
One can decide whether an architecture is well-connected in polynomial time.



Well connected architectures

Theorem

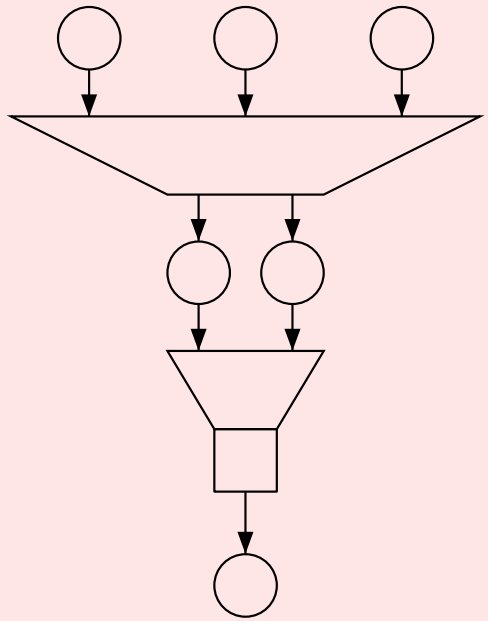
One can decide whether an architecture is well-connected in polynomial time.



Well connected architectures

Theorem

One can decide whether an architecture is well-connected in polynomial time.



Rasala Lehman–Lehman 2004

One can solve the network information flow in the special case where there is a unique sink in polynomial time.

Well connected architectures

Theorem

There exists a well connected architecture which is not uniformly well connected.

Well connected architectures

Theorem

There exists a well connected architecture which is not uniformly well connected.

Lemma (Rasala Lehman–Lehman 2004)

If $f^1, \ldots, f^n$ are pairwise independent functions of the form $S^2 \to S$, then $n \leq |S| + 1$.

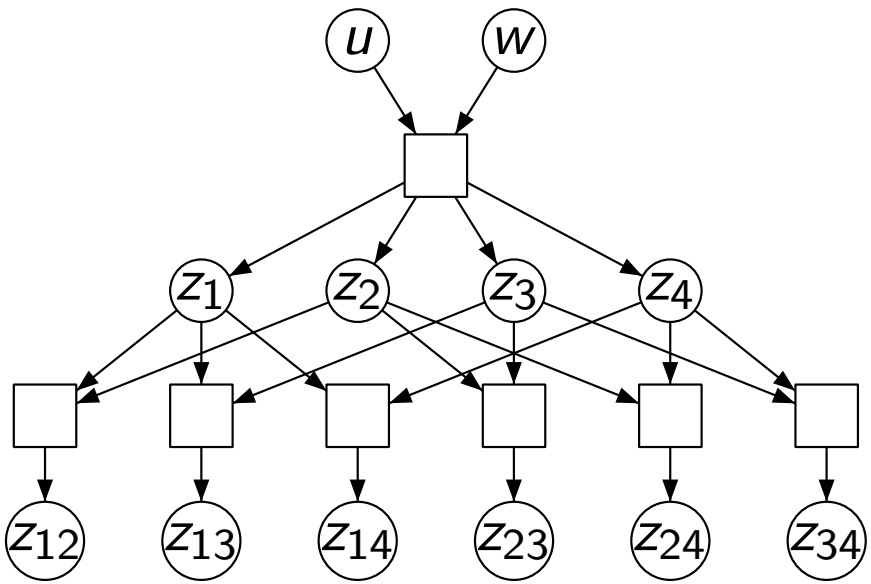
Well connected architectures

Theorem

There exists a well connected architecture which is not uniformly well connected.

Lemma (Rasala Lehman–Lehman 2004)

If $f^1, \ldots, f^n$ are pairwise independent functions of the form $S^2 \to S$, then $n \leq |S| + 1$.



Well connected preordered architectures are undecidable

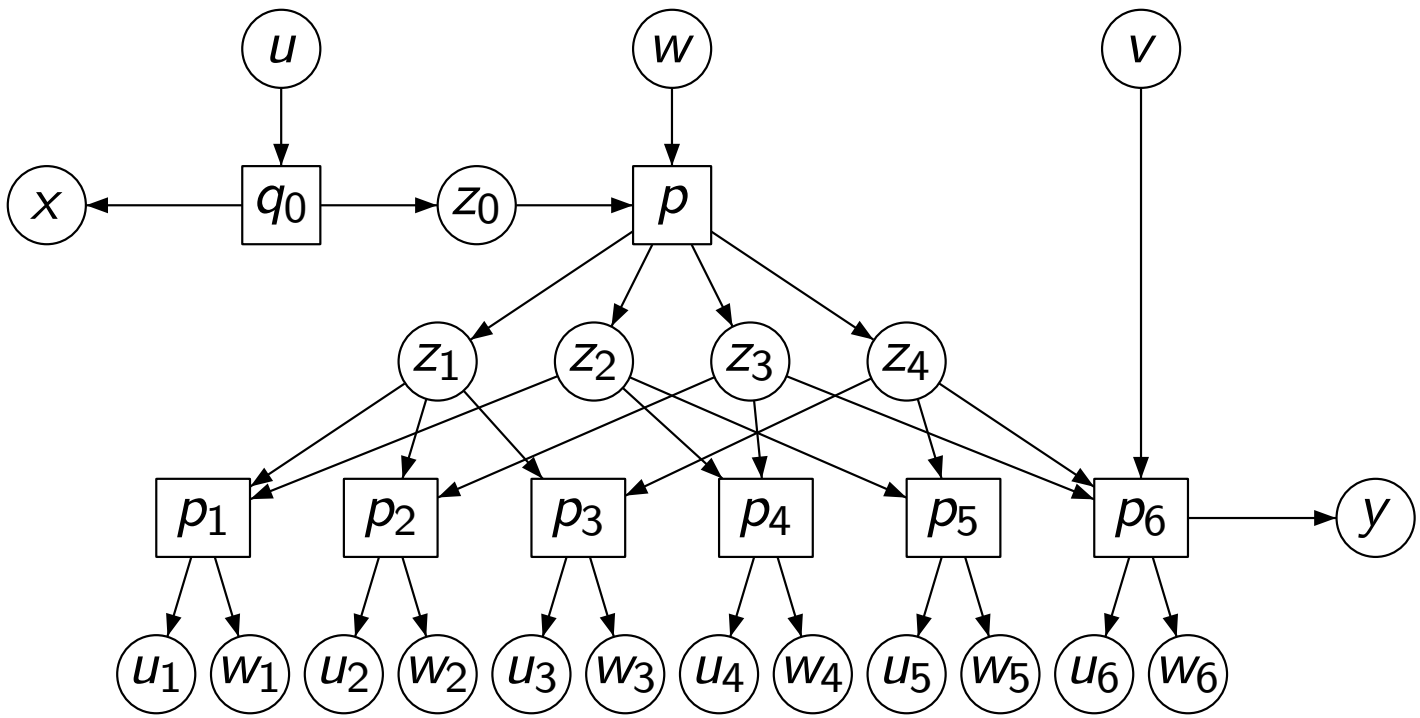
Theorem

The synthesis problem for LTL specifications and well connected, preordered information is undecidable.

Well connected preordered architectures are undecidable

Theorem

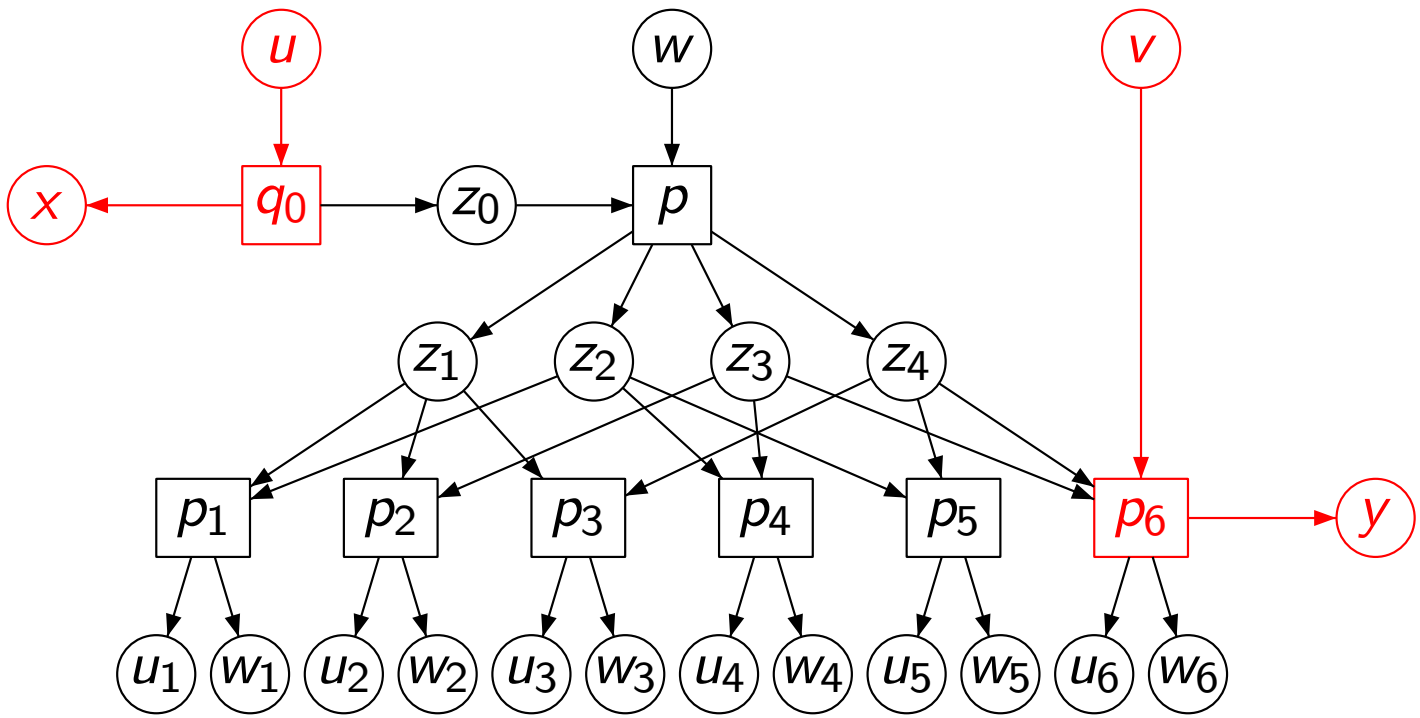
The synthesis problem for LTL specifications and well connected, preordered information is undecidable.



Well connected preordered architectures are undecidable

Theorem

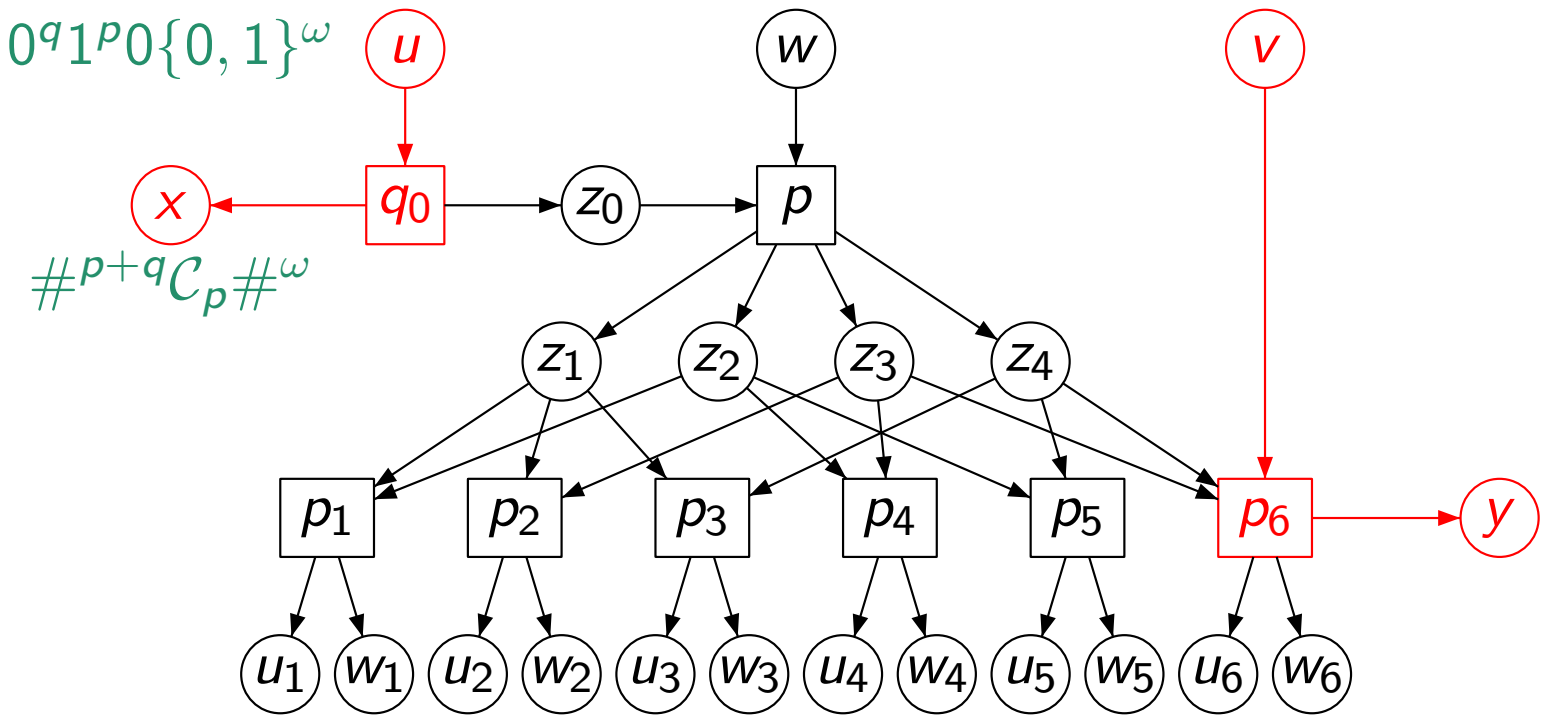
The synthesis problem for LTL specifications and well connected, preordered information is undecidable.



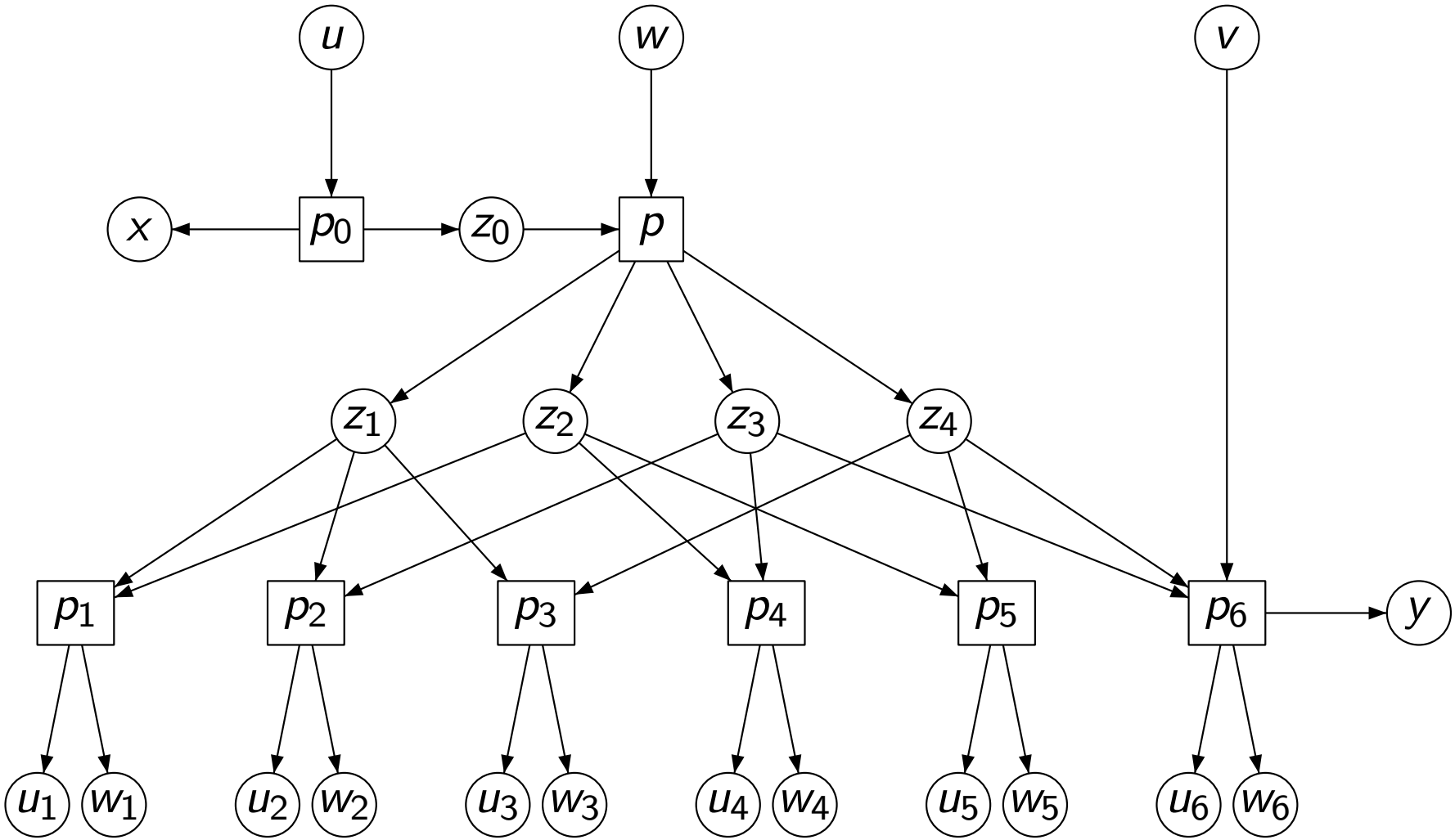
Well connected preordered architectures are undecidable

Theorem

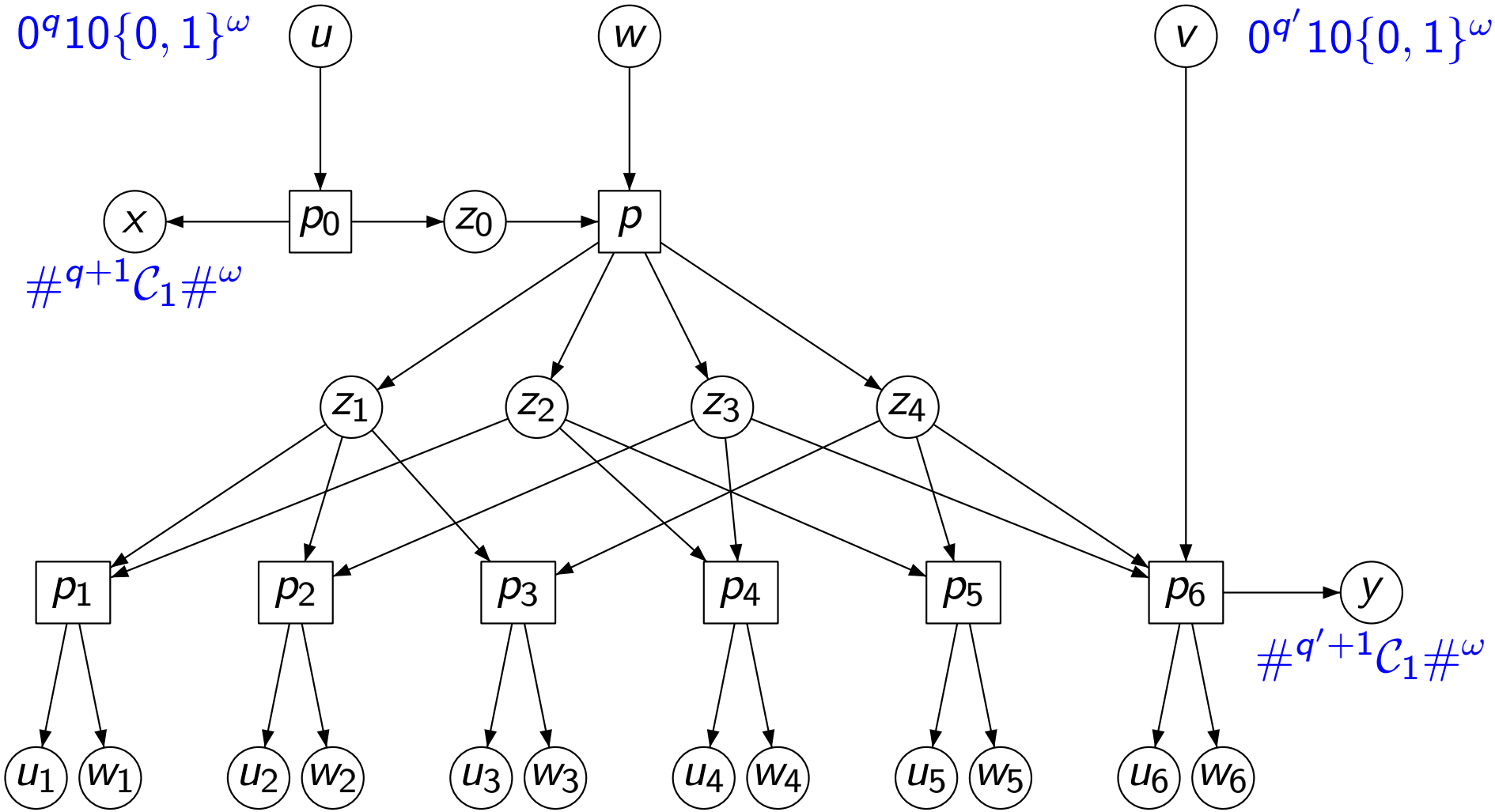
The synthesis problem for LTL specifications and well connected, preordered information is undecidable.



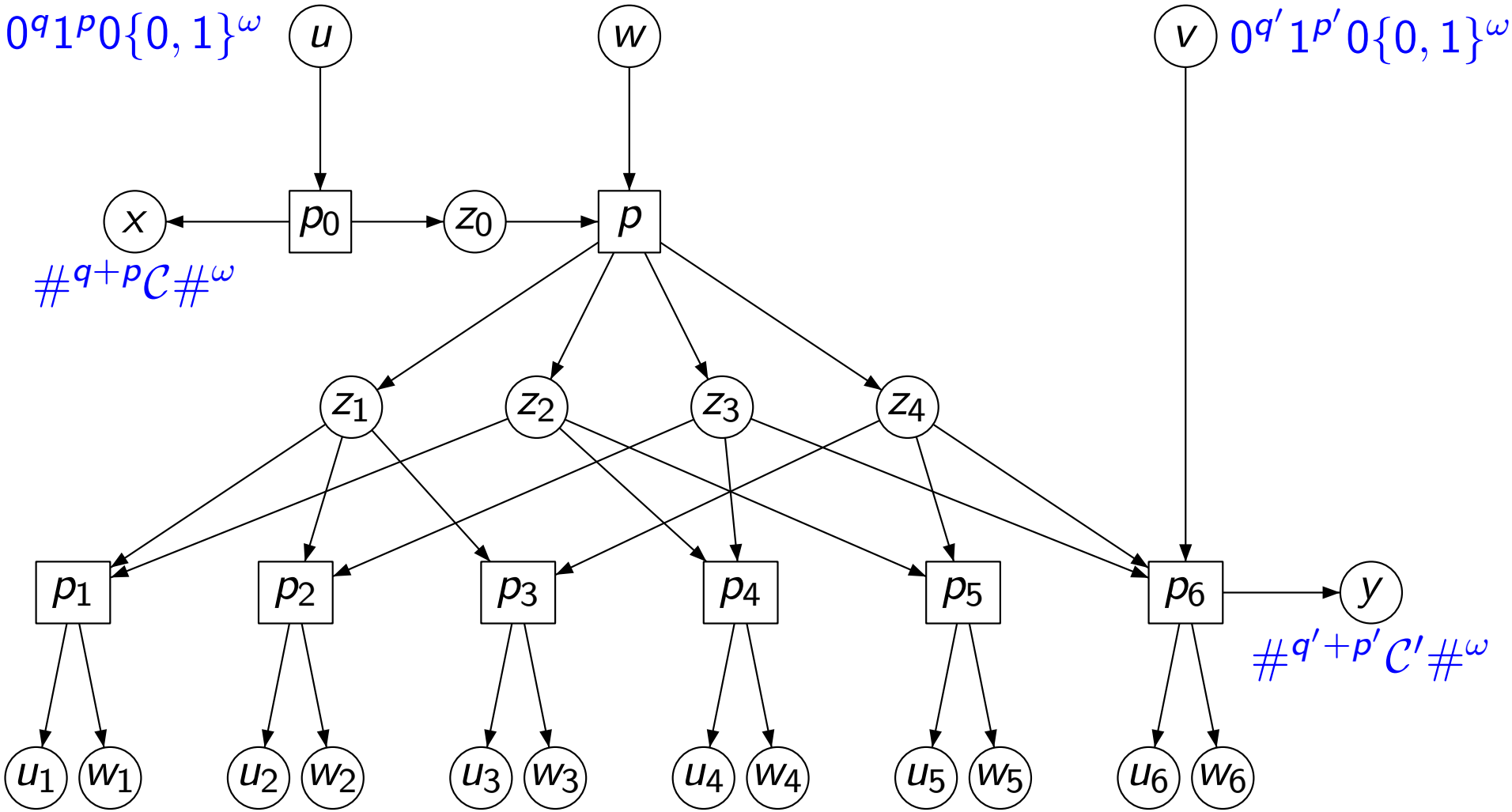
Well connected preordered architectures are undecidable : The specification



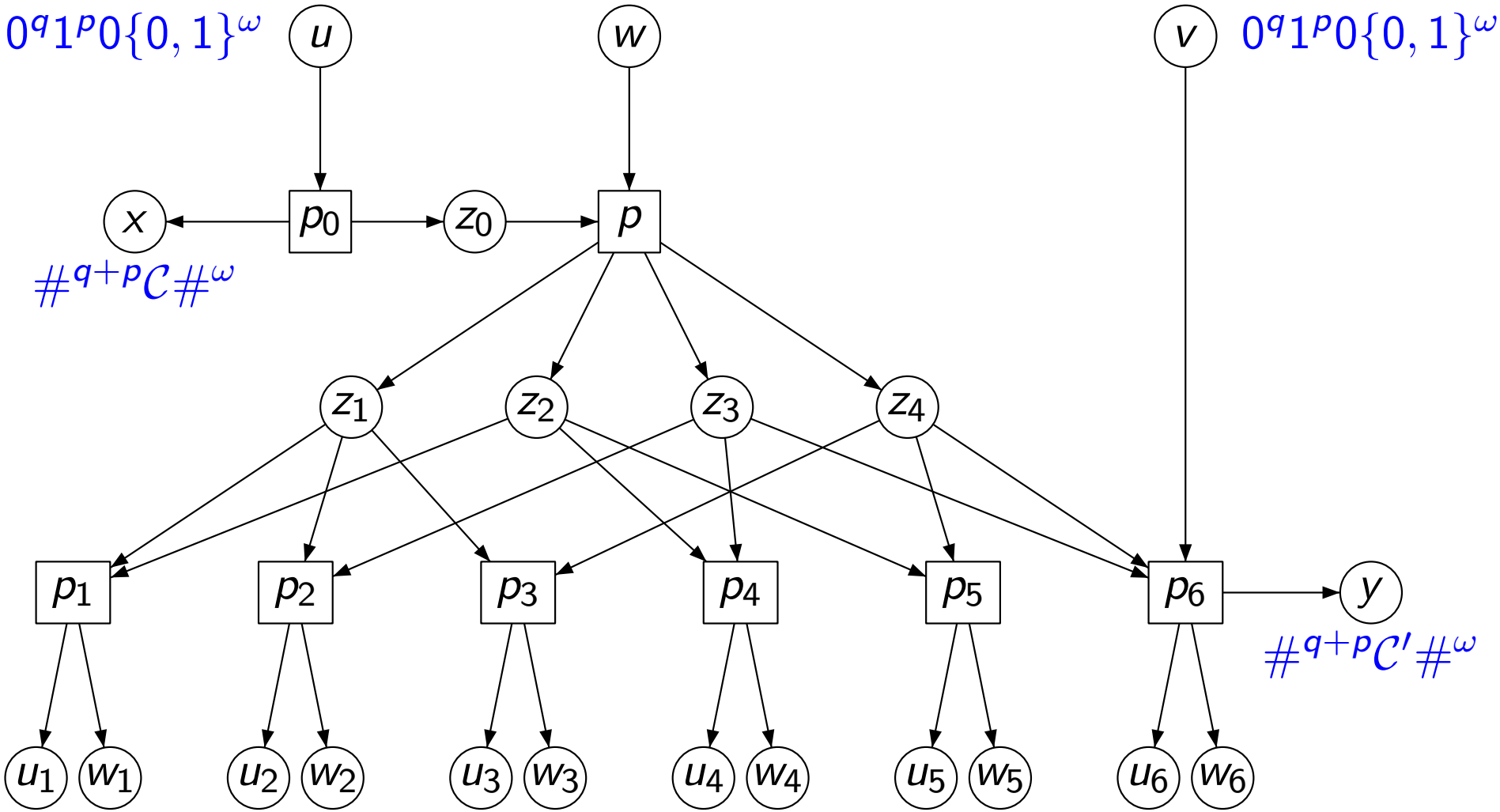
Well connected preordered architectures are undecidable : The specification



Well connected preordered architectures are undecidable : The specification

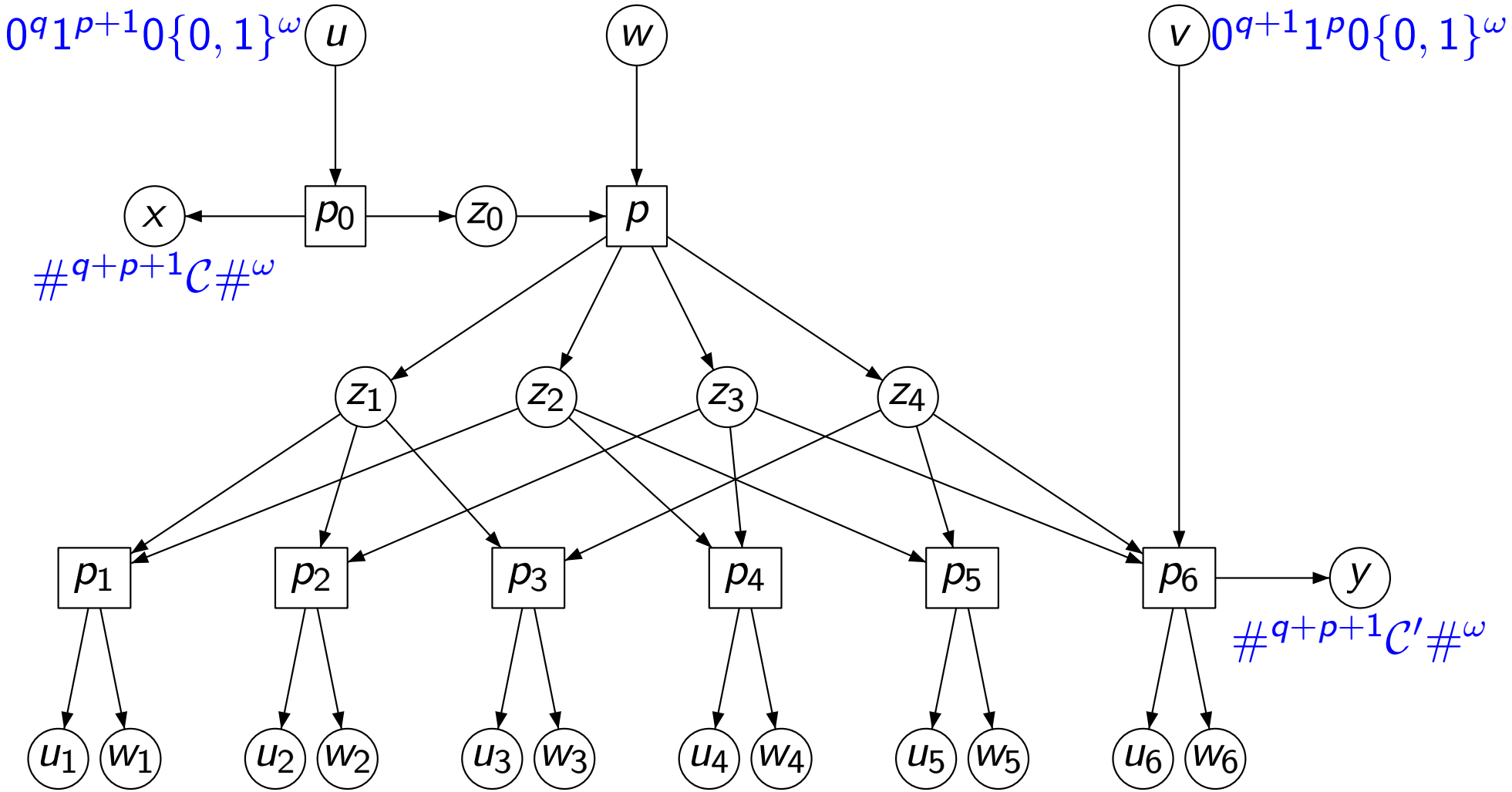


Well connected preordered architectures are undecidable : The specification



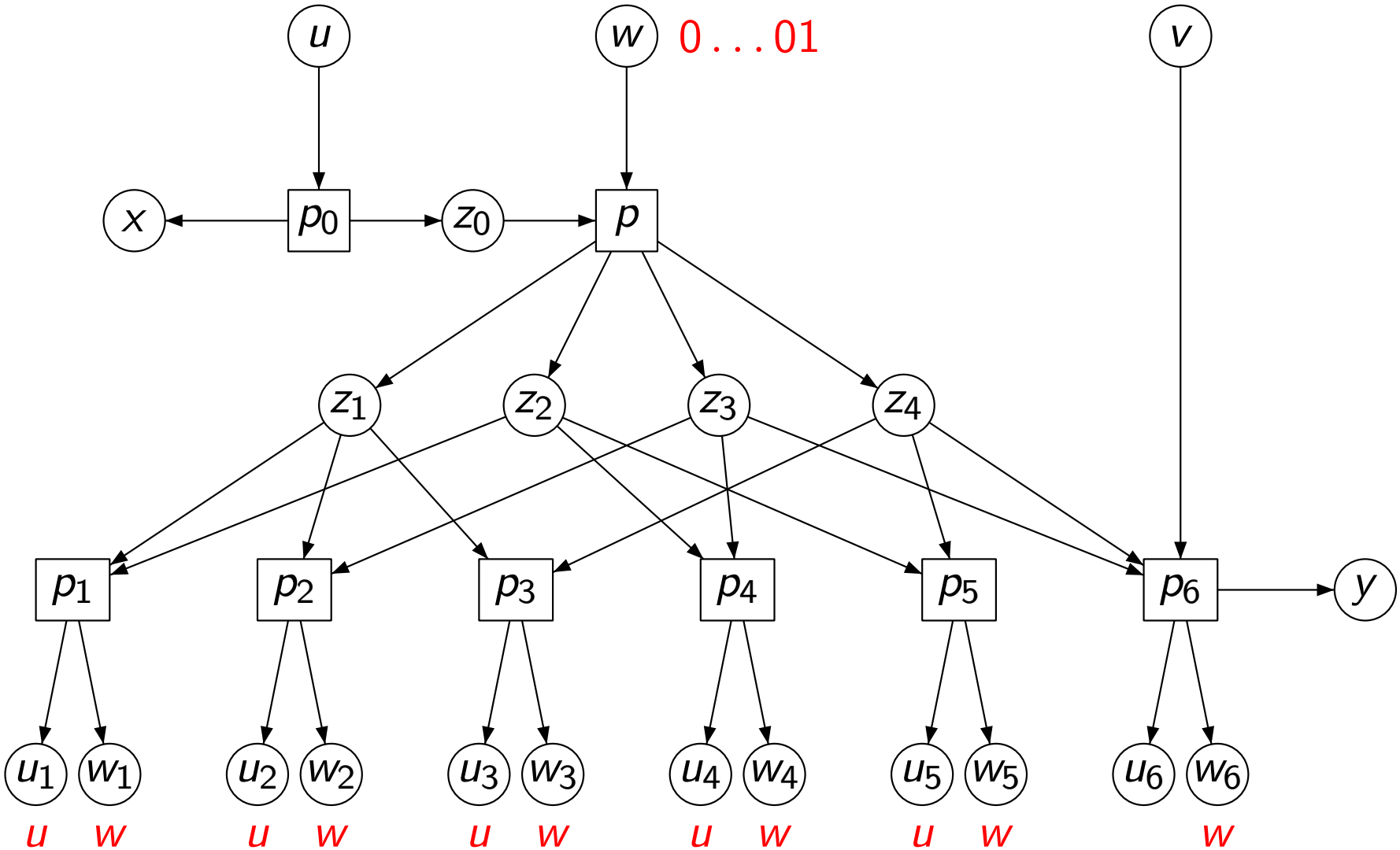
$$u = v \implies x = y$$

Well connected preordered architectures are undecidable : The specification

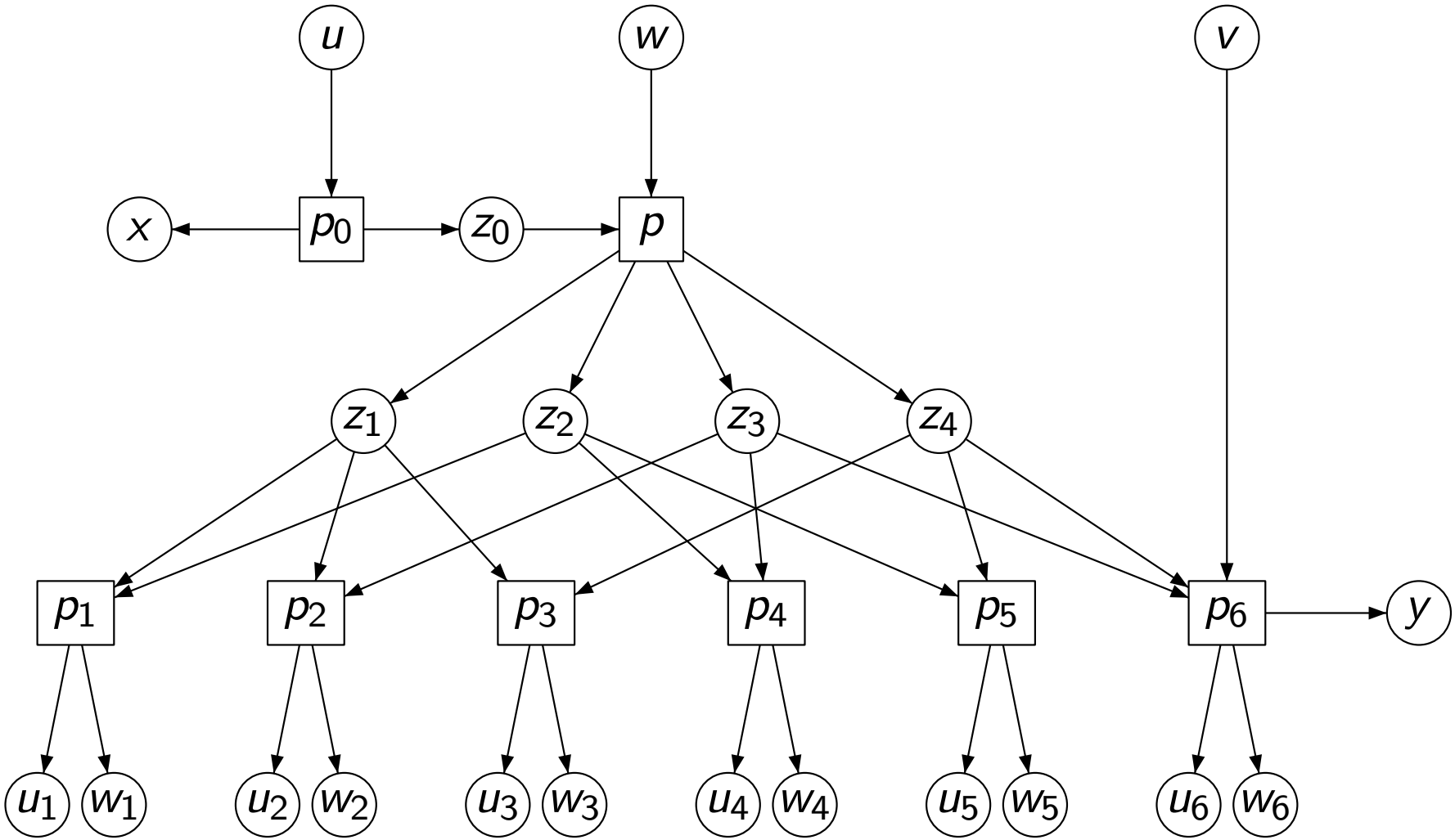


$$u = v + 1 \implies \mathcal{C}' \vdash \mathcal{C}$$

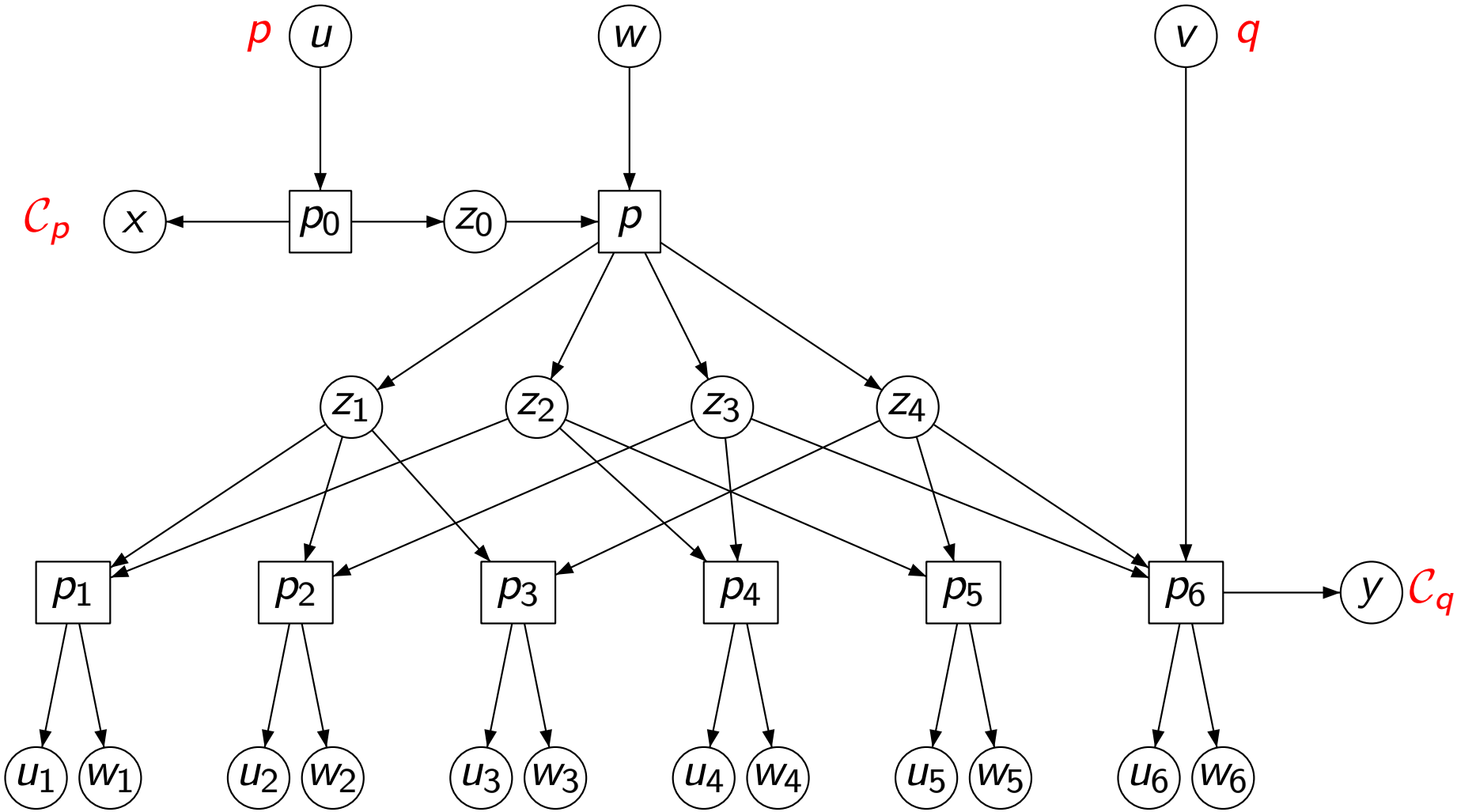
Well connected preordered architectures are undecidable : The specification



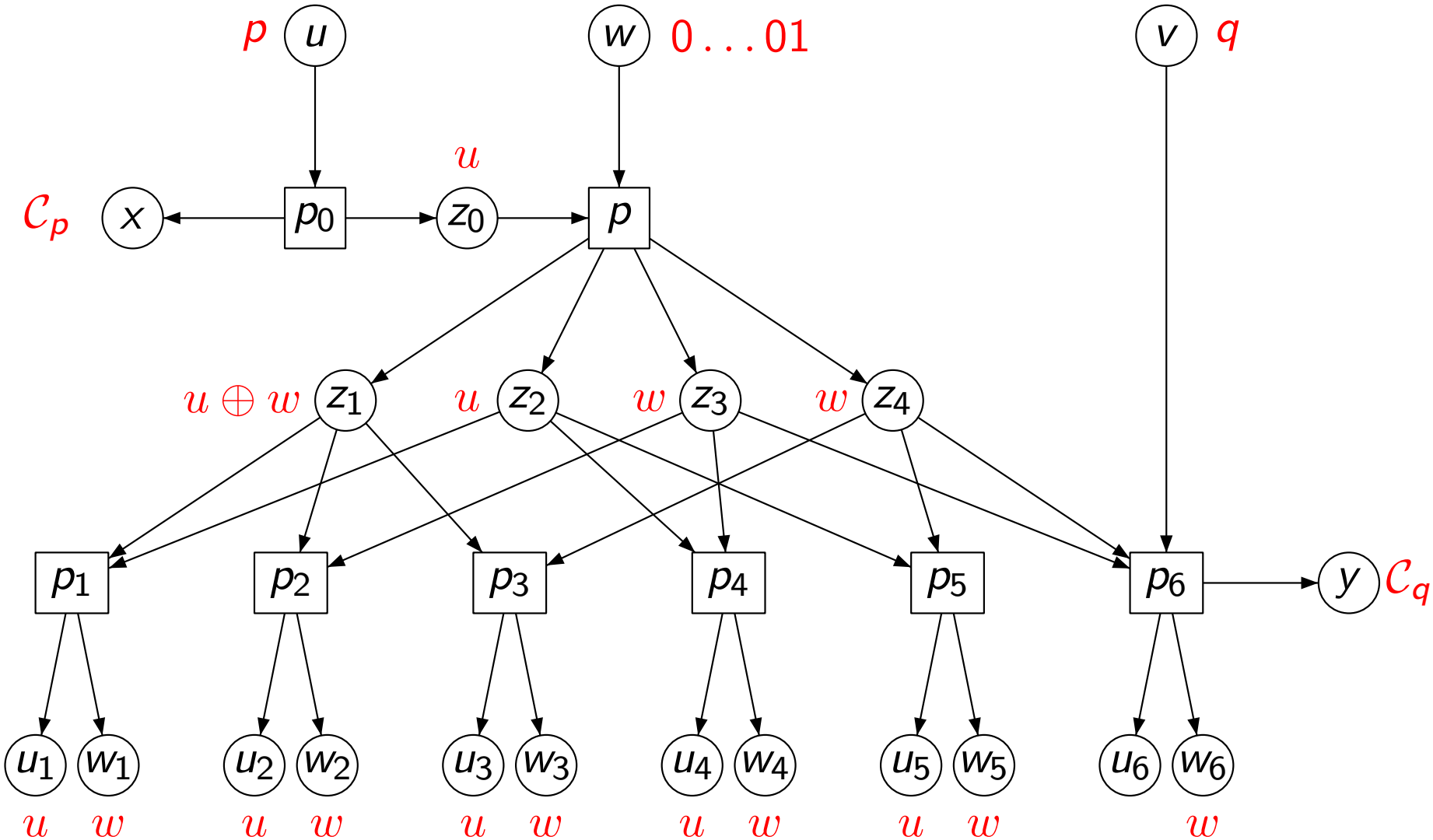
Well connected preordered architectures are undecidable : A distributed implementation



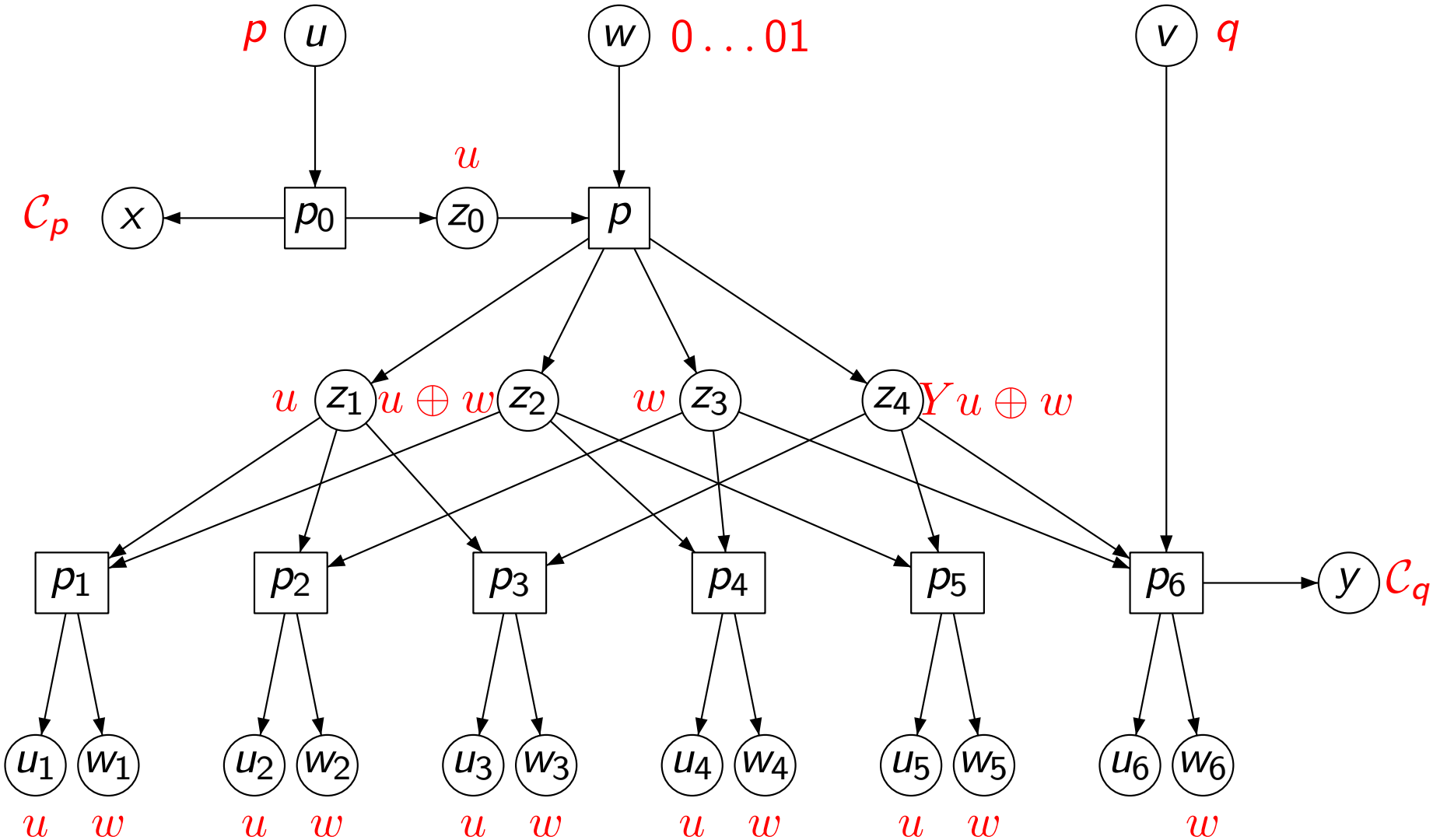
Well connected preordered architectures are undecidable : A distributed implementation



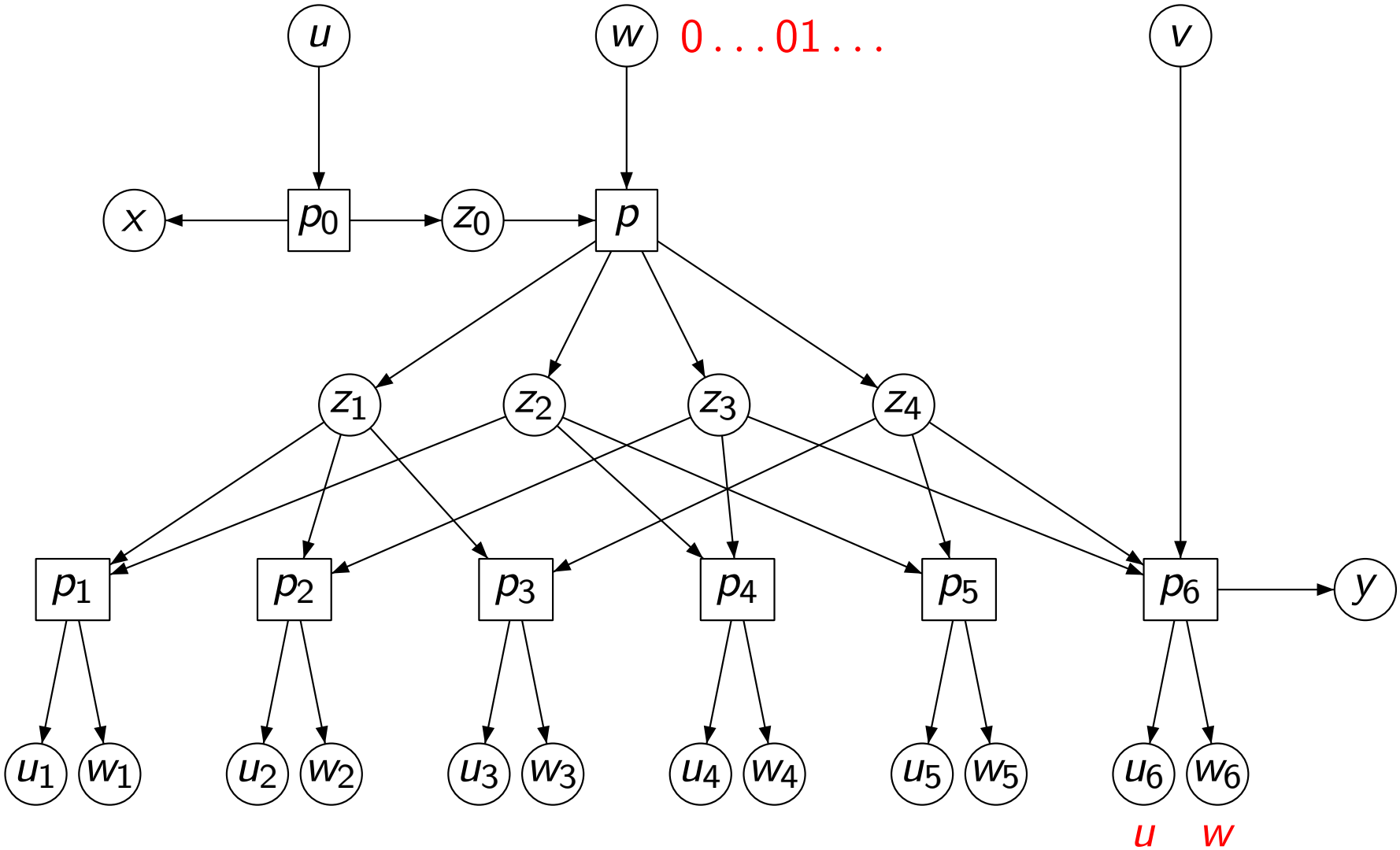
Well connected preordered architectures are undecidable : A distributed implementation



Well connected preordered architectures are undecidable : Another distributed implementation



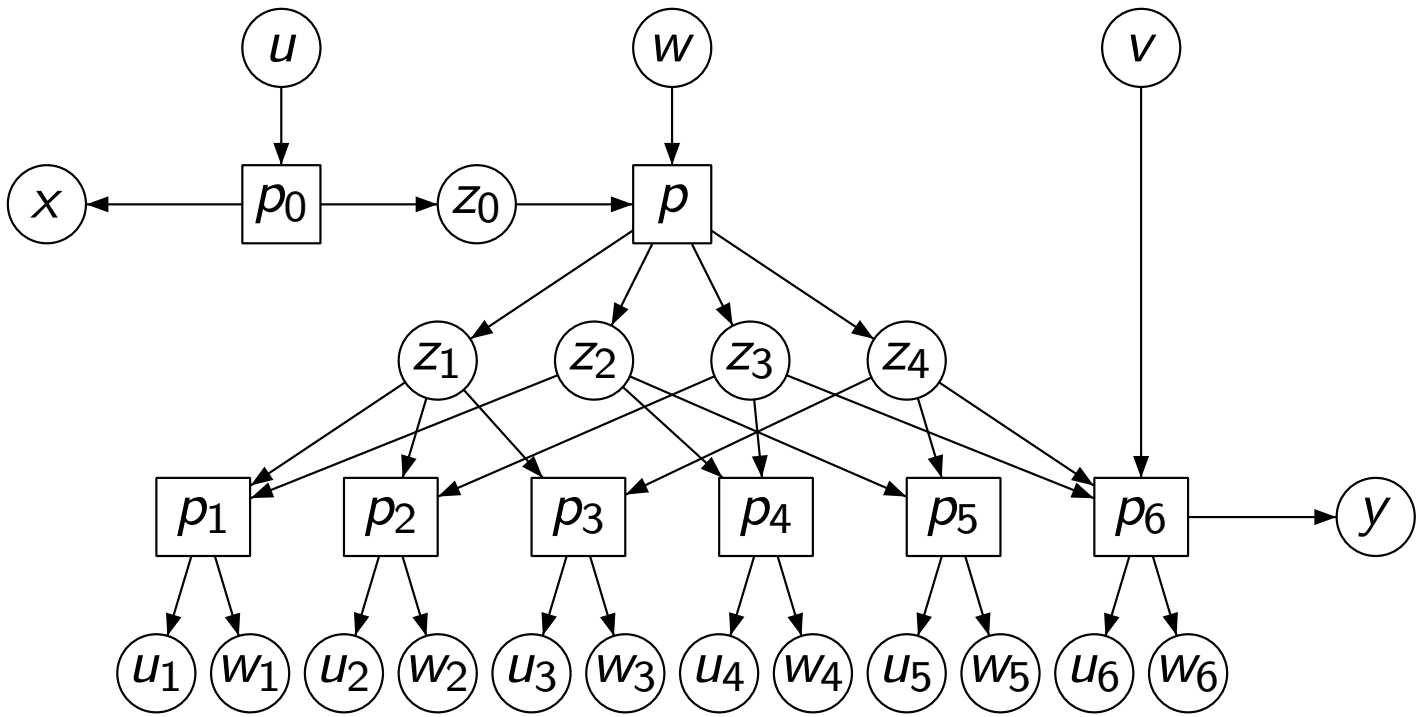
Well connected preordered architectures are undecidable : The specification (end)



Well connected preordered architectures are undecidable: Masking one bit of u to y

Lemma

Let $u^b = 0^q b 1^p 0 u'$ *and* $w = 0^q 1 w'$. *Then*
$(\hat{f}_{z_3}, \hat{f}_{z_4})(u^0[n], w[n]) = (\hat{f}_{z_3}, \hat{f}_{z_4})(u^1[n], w[n])$

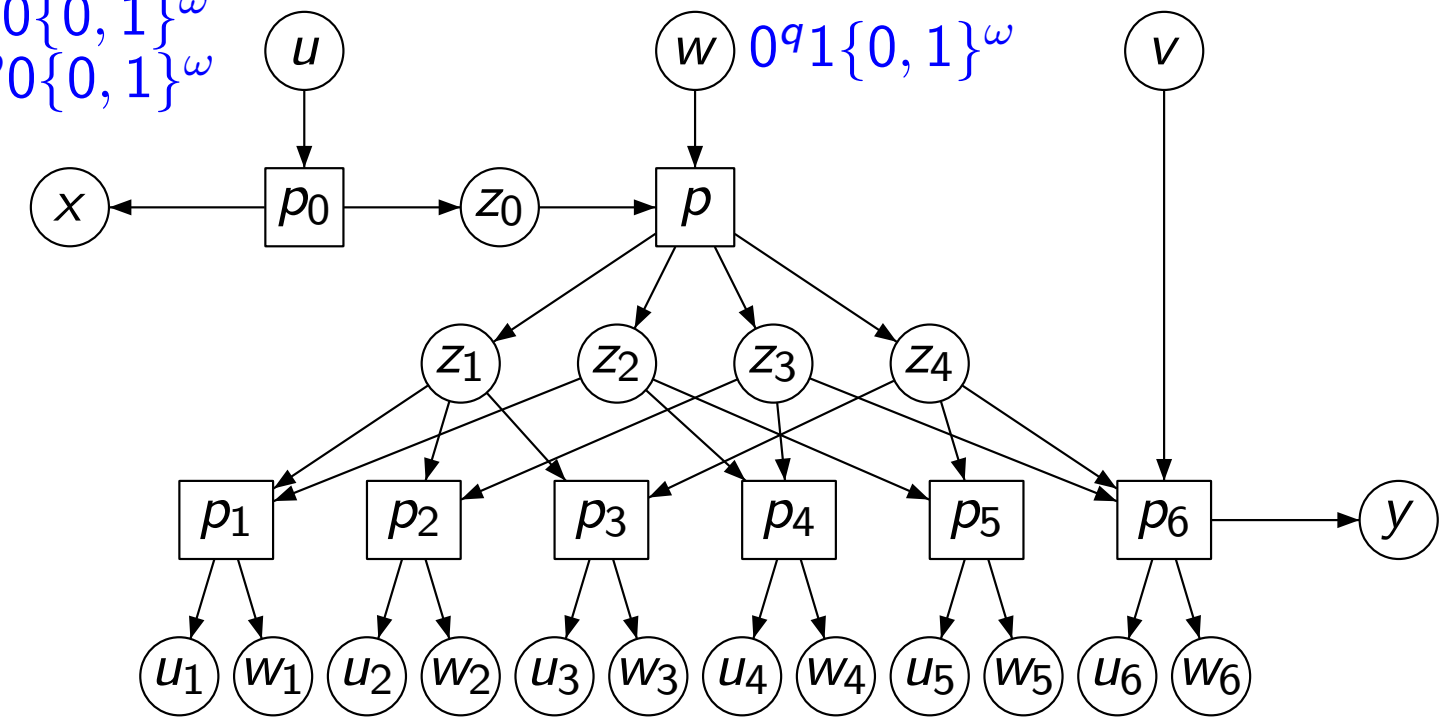


Well connected preordered architectures are undecidable: Masking one bit of u to y

Lemma

Let $u^b = 0^q b 1^p 0 u'$ *and* $w = 0^q 1 w'$. *Then*
$(\hat{f}_{z_3}, \hat{f}_{z_4})(u^0[n], w[n]) = (\hat{f}_{z_3}, \hat{f}_{z_4})(u^1[n], w[n])$

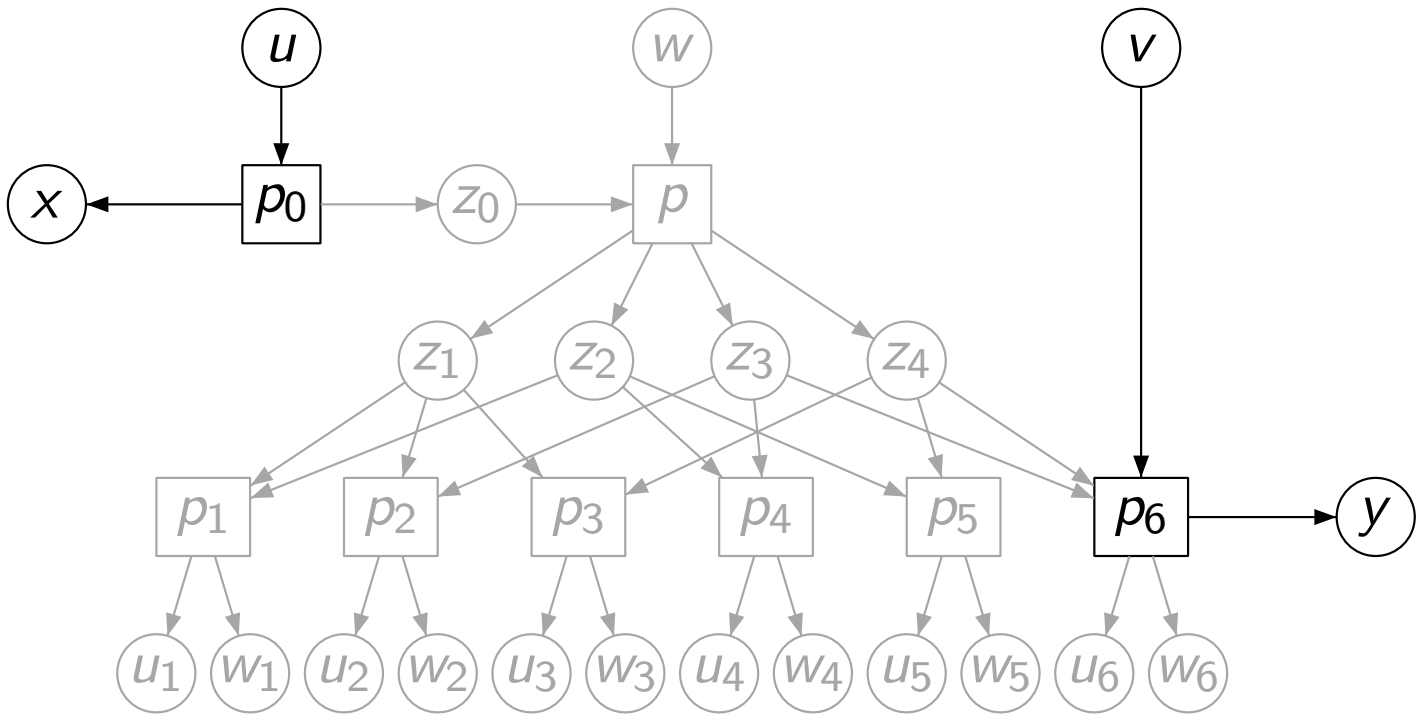
$p \sim 0^q 0 1^p 0 \{0,1\}^\omega$
$p+1 \sim 0^q 1 1^p 0 \{0,1\}^\omega$



Well connected preordered architecures are undecidable: Enforcing output of the correct configuration

Lemma

For all $q \geq 0$, $u \in 0^q 1^p 0\{0,1\}^\omega \implies x = \#^{q+p} \mathcal{C}_p \#^\omega$

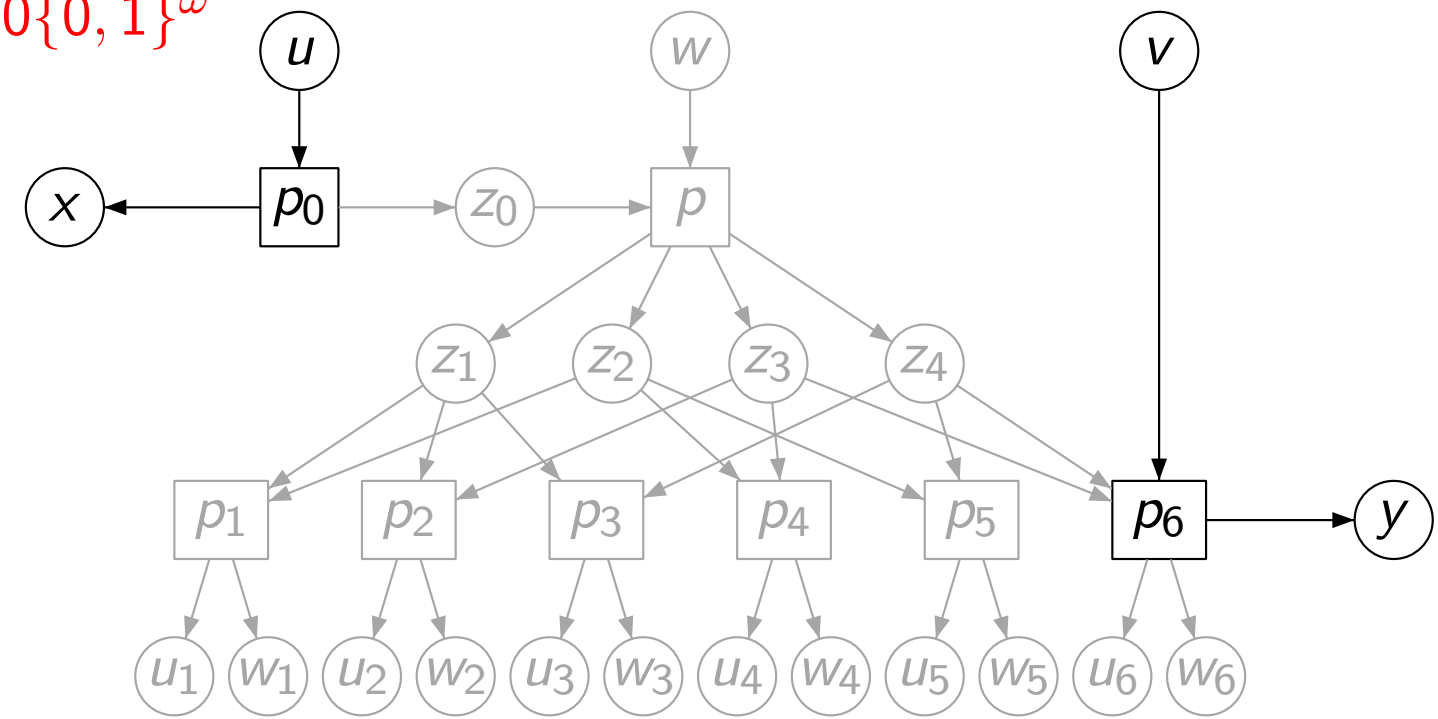


Well connected preordered architecures are undecidable: Enforcing output of the correct configuration

Lemma

For all $q \geq 0$, $u \in 0^q 1^p 0\{0,1\}^\omega \implies x = \#^{q+p} \mathcal{C}_p \#^\omega$

$p + 1 \sim 0^q 11^p 0\{0,1\}^\omega$

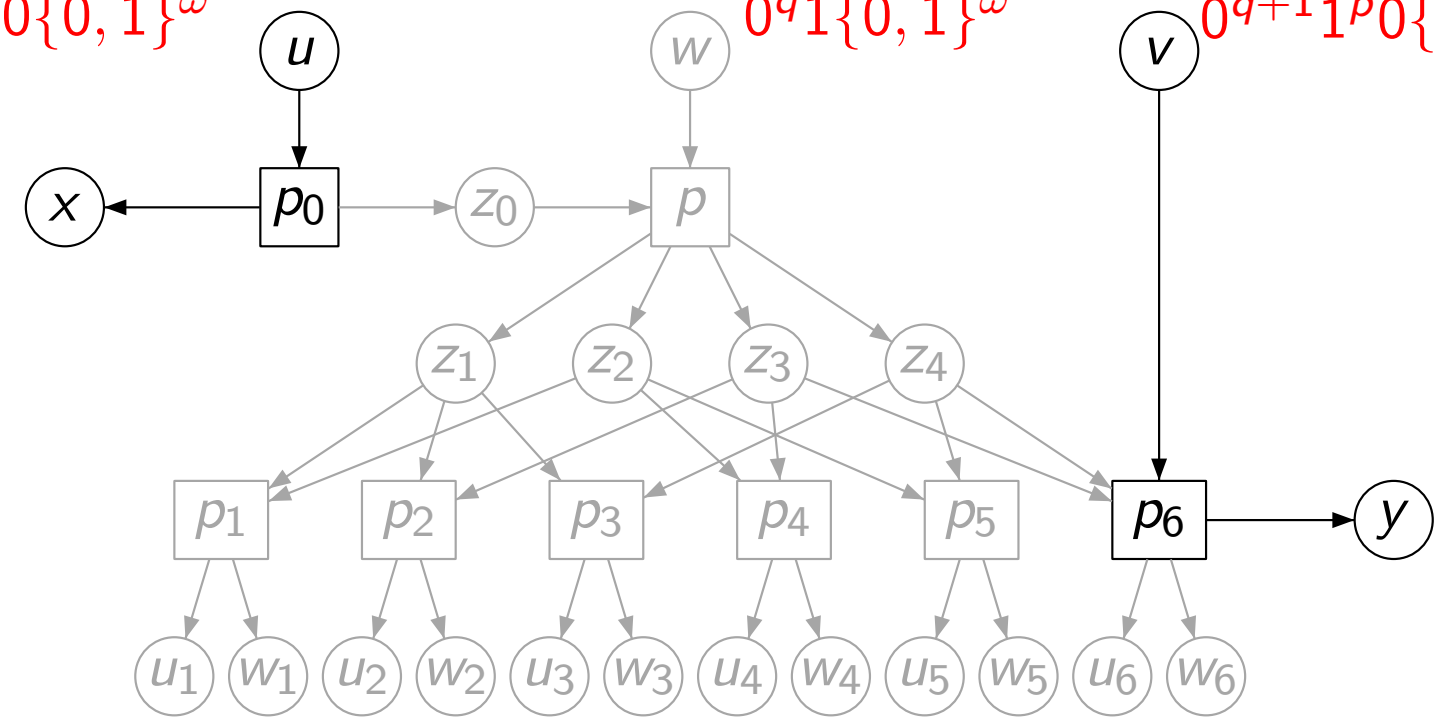


Well connected preordered architecures are undecidable: Enforcing output of the correct configuration

Lemma

For all $q \geq 0$, $u \in 0^q 1^p 0\{0,1\}^\omega \implies x = \#^{q+p} \mathcal{C}_p \#^\omega$

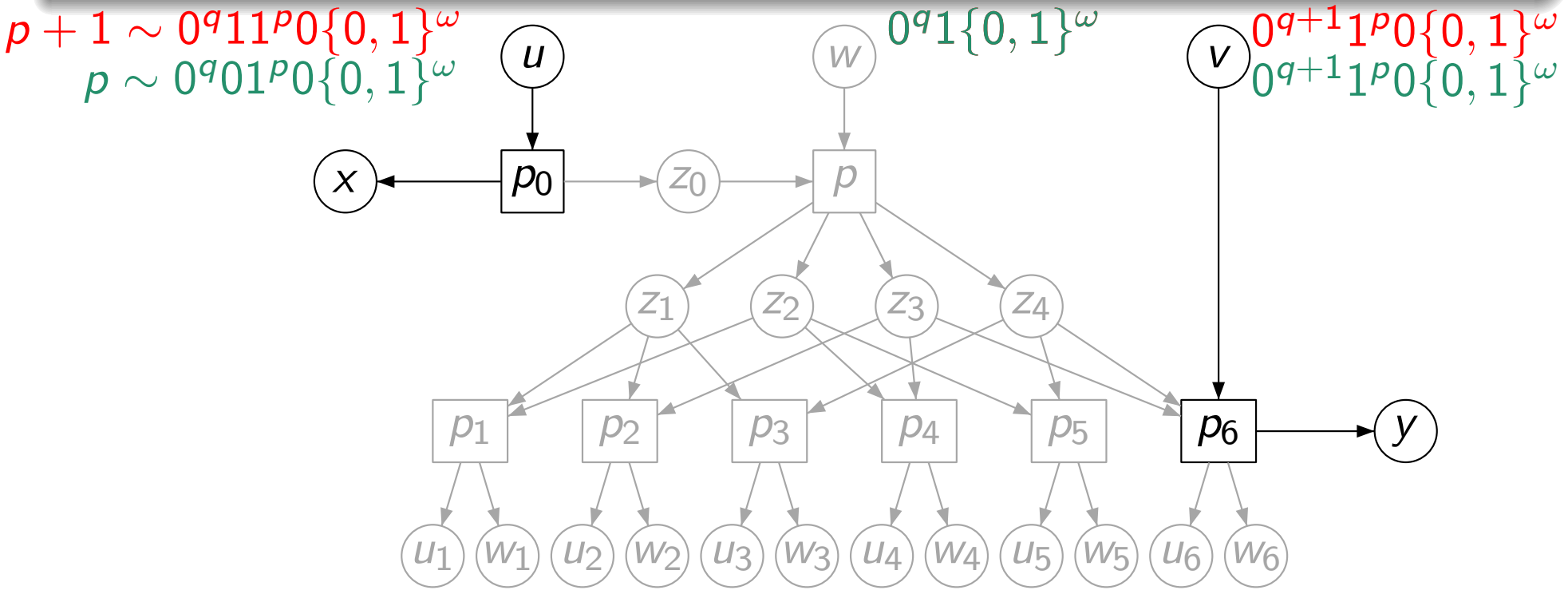
$p+1 \sim 0^q 11^p 0\{0,1\}^\omega$ $0^q 1\{0,1\}^\omega$ $0^{q+1} 1^p 0\{0,1\}^\omega$



Well connected preordered architecures are undecidable: Enforcing output of the correct configuration

Lemma

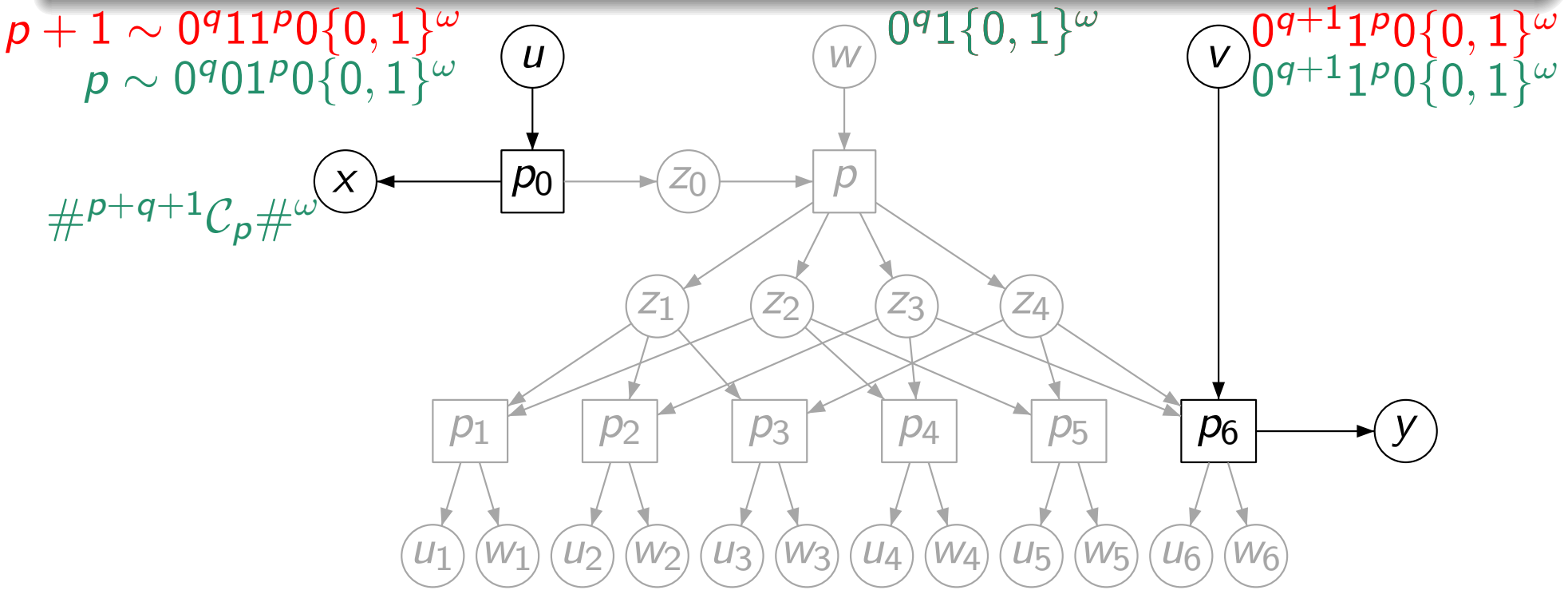
For all $q \geq 0$, $u \in 0^q 1^p 0\{0,1\}^\omega \implies x = \#^{q+p} \mathcal{C}_p \#^\omega$



Well connected preordered architecures are undecidable: Enforcing output of the correct configuration

Lemma

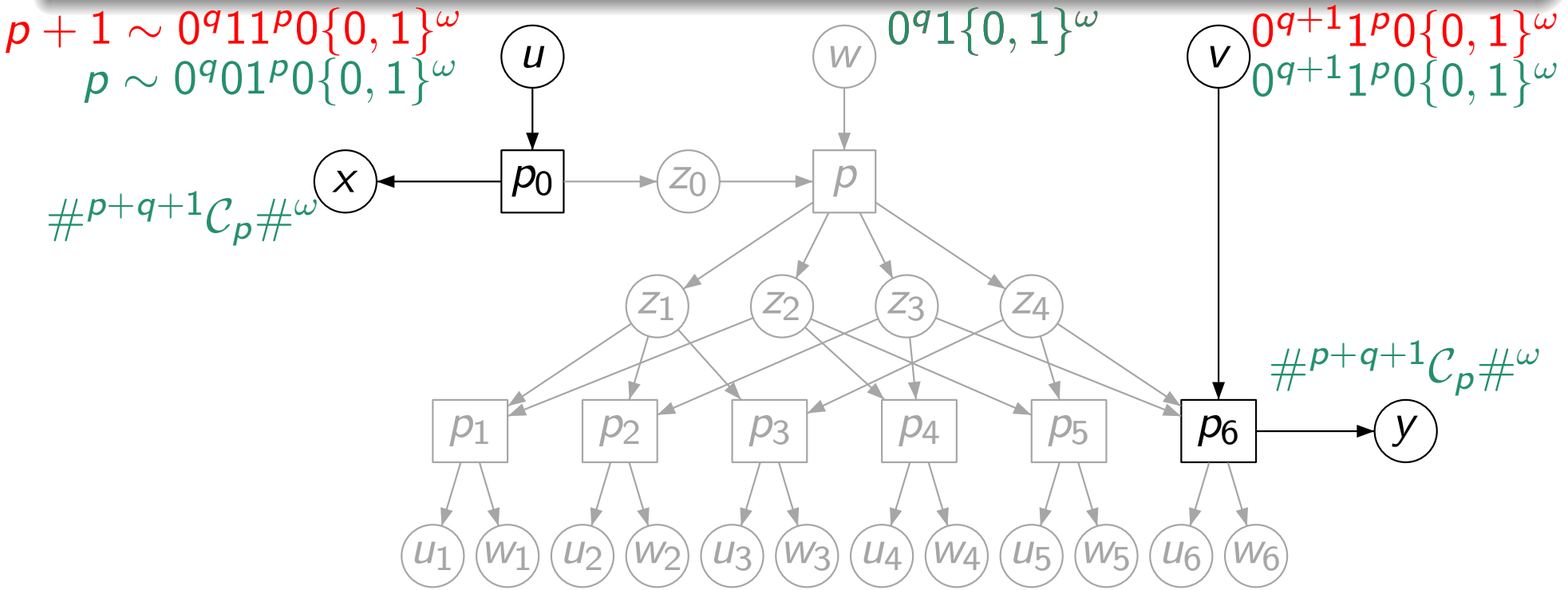
For all $q \geq 0$, $u \in 0^q 1^p 0\{0,1\}^\omega \implies x = \#^{q+p} \mathcal{C}_p \#^\omega$



Well connected preordered architecures are undecidable: Enforcing output of the correct configuration

Lemma

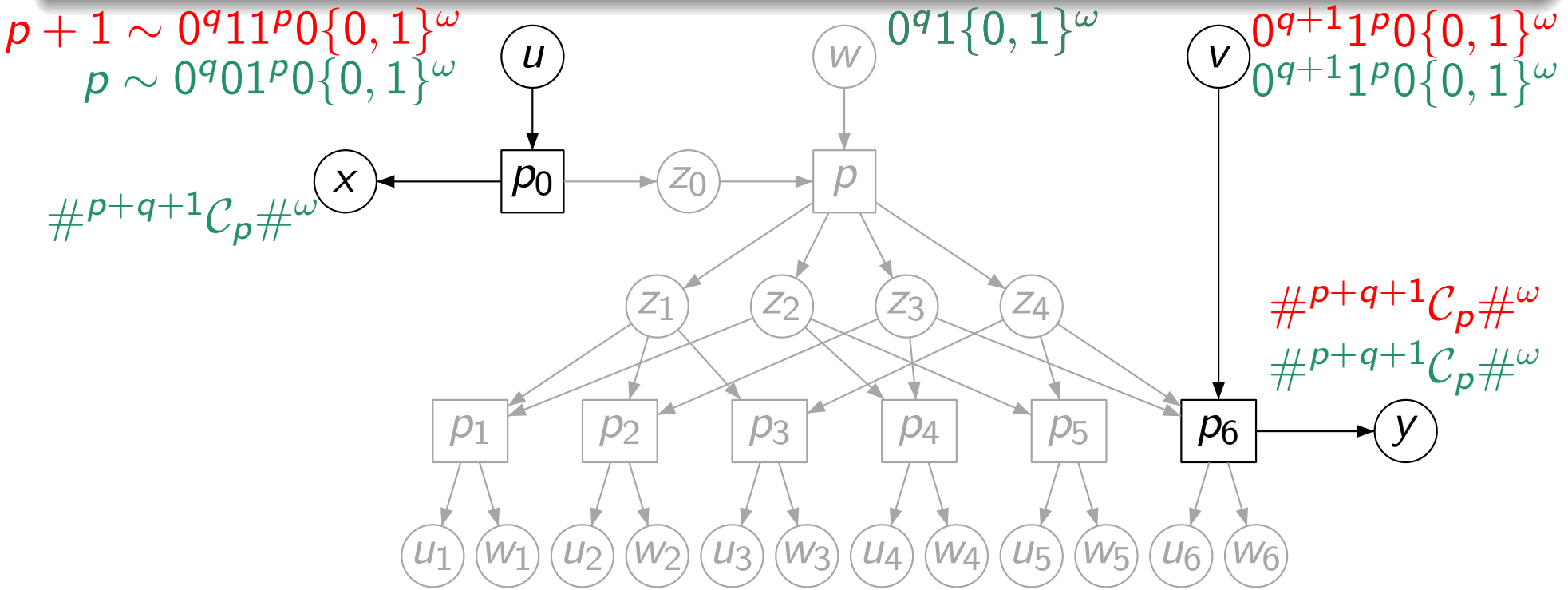
For all $q \geq 0$, $u \in 0^q 1^p 0\{0,1\}^\omega \implies x = \#^{q+p}\mathcal{C}_p\#^\omega$



Well connected preordered architecures are undecidable: Enforcing output of the correct configuration

Lemma

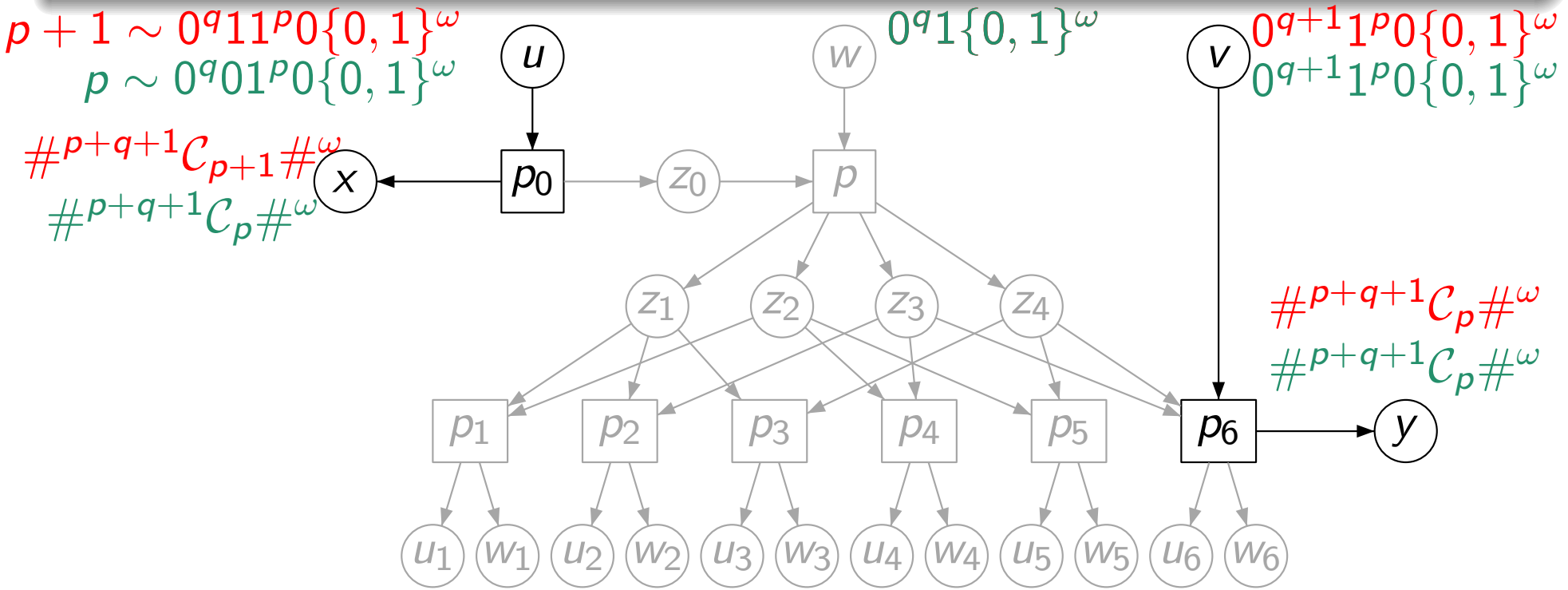
For all $q \geq 0$, $u \in 0^q 1^p 0\{0,1\}^\omega \implies x = \#^{q+p} \mathcal{C}_p \#^\omega$



Well connected preordered architecures are undecidable: Enforcing output of the correct configuration

Lemma

For all $q \geq 0$, $u \in 0^q 1^p 0\{0,1\}^\omega \implies x = \#^{q+p}\mathcal{C}_p\#^\omega$




April Rasala Lehman and Eric Lehman.
Complexity classification of network information flow problems.
In *SODA*, pages 142–150, 2004.