Coalgebraic Bisimulation-up-to

Jurriaan Rot [1,2] Marcello Bonsangue [1,2] Jan Rutten [2,3]

1. LIACS – Leiden University

2. Centrum Wiskunde en Informatica

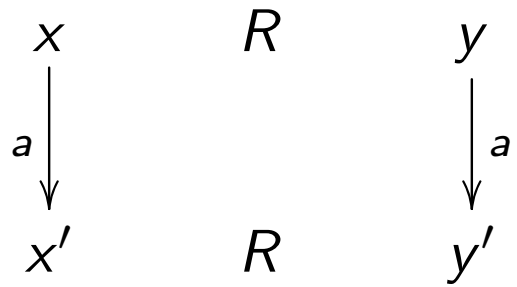
3. Radboud University Nijmegen

SOFSEM 2013

Bisimilarity

A *bisimulation* is a relation R on the states of a labelled transition system such that for all $(x, y) \in R$:

- if $x \xrightarrow{a} x'$ then there is some y' s.t. $y \xrightarrow{a} y'$ and $(x', y') \in R$, and
- vice versa.

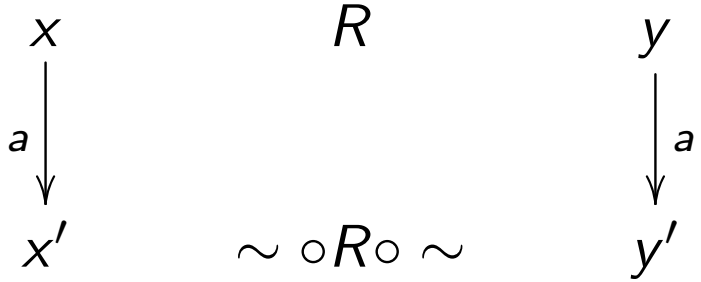


In order to prove *bisimilarity* of x and y, denoted $x \sim y$, it suffices to construct a bisimulation containing the pair (x, y).

Bisimilarity

A *bisimulation up to bisimilarity* (Milner 1983) is a relation R such that for all $(x, y) \in R$:

- if $x \xrightarrow{a} x'$ then there is some y' s.t. $y \xrightarrow{a} y'$ and $x' \sim u \; R \; v \sim y'$, and
- vice versa.

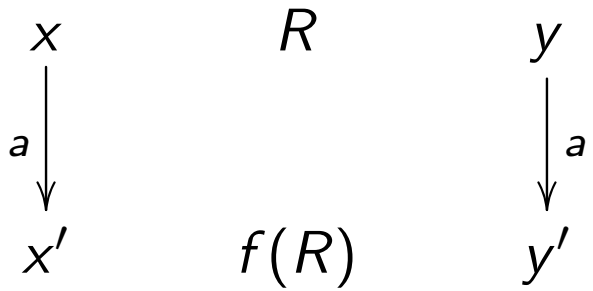


This is *sound*; every bisimulation up to bisimilarity R is contained in $\sim$ (bisimilarity).

Bisimilarity

A *bisimulation up to f* (Sangiorgi 1998) is a relation R such that for all $(x, y) \in R$:

- if $x \xrightarrow{a} x'$ then there is some y' s.t. $y \xrightarrow{a} y'$ and $(x', y') \in f(R)$, and
- vice versa.



- up to bisimilarity, up to context, up to union, combinations, ...
- smaller, easier, more elegant proofs of bisimilarity.

Bisimulation(-up-to) on other systems

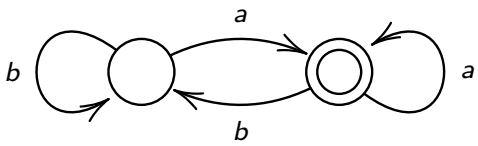
Notions of bisimulation for other types of systems exist as well; e.g. for deterministic automata, and for streams.

Bisimulation up-to techniques may also be very useful.

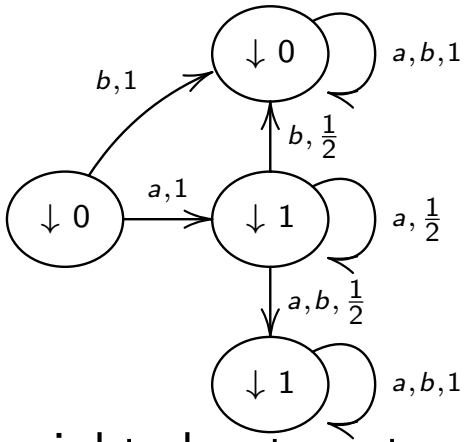
- (Bonchi, Pous – POPL 2013)
- Stream calculus

Coalgebra, bisimulations, coinduction

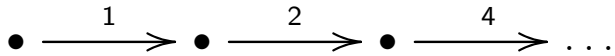
Coalgebra: mathematical theory for the uniform study of many different types of state-based systems and their (infinite) behaviour.



deterministic automata



weighted automata



stream systems

The type of coalgebra determines a canonical notion of behaviour (*final coalgebra*) and of equivalence (*bisimilarity*).

Coinduction: if two states are bisimilar, then they are behaviourally equivalent.

Our contribution

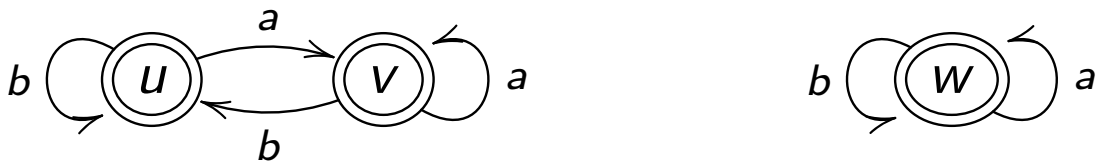
Generalization of bisimulation-up-to from labelled transition systems to arbitrary types of coalgebras.

Allows for smaller, easier, more elegant proofs of bisimilarity (and thus behavioural equivalence) for many kinds of state-based systems.

Related:

- (Lenisa 1999) some *results*, some *problems*
- (Bartels 2004) bisimulation up to context

Deterministic automata, bisimulation

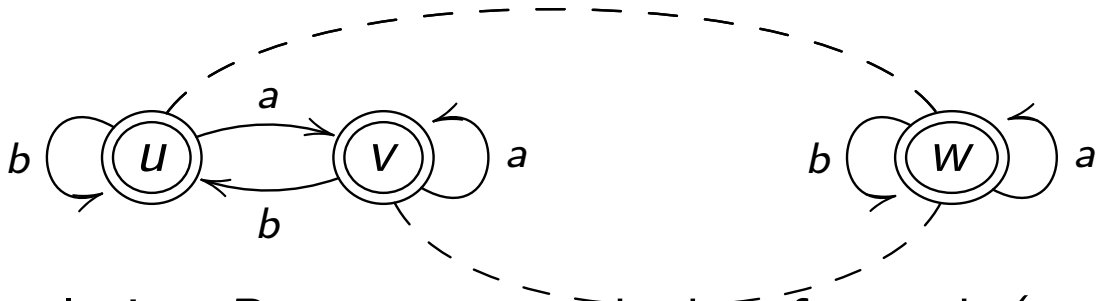


Bisimulation: a relation R on states such that for each $(x, y) \in R$:

- x accepts iff y accepts
- for each alphabet letter a, let $x \xrightarrow{a} x'$ and $y \xrightarrow{a} y'$; then $(x', y') \in R$.

Coinduction: if $(x, y) \in R$ for a bisimulation R then $L(x) = L(y)$.

Deterministic automata, bisimulation

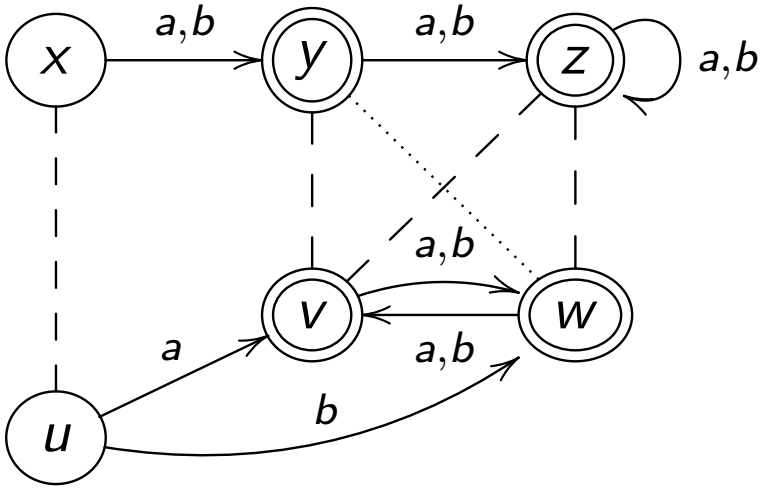


Bisimulation: a relation R on states such that for each $(x, y) \in R$:

- x accepts iff y accepts
- for each alphabet letter a, let $x \xrightarrow{a} x'$ and $y \xrightarrow{a} y'$; then $(x', y') \in R$.

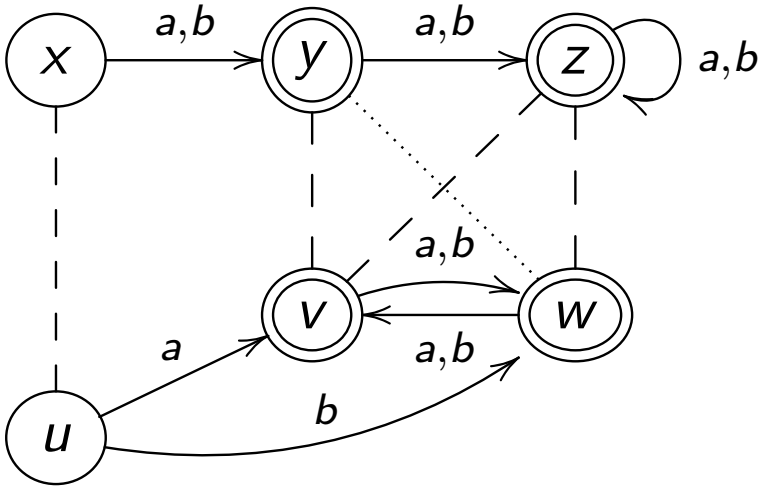
Coinduction: if $(x, y) \in R$ for a bisimulation R then $L(x) = L(y)$.

Deterministic automata: bisimulation-up-to



Dashed lines + dotted line: bisimulation.

Deterministic automata: bisimulation-up-to

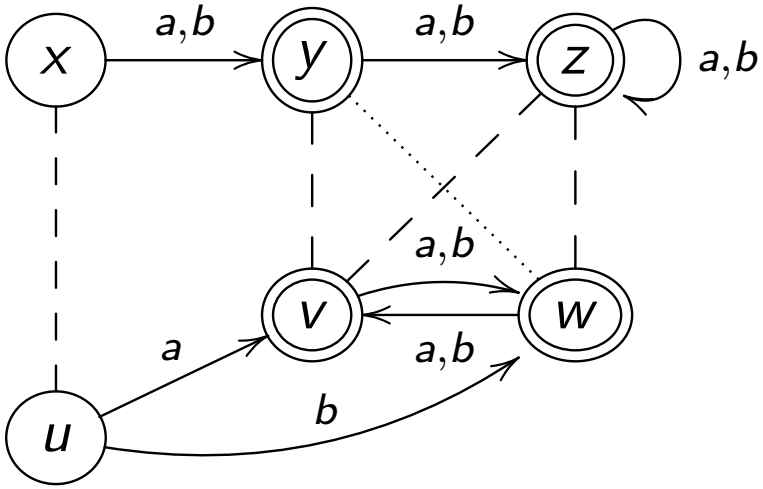


Dashed lines + dotted line: bisimulation.

Bisimulation up to equivalence: a relation R on states such that for each $(x, y) \in R$:

- x accepts iff y accepts
- for each alphabet letter a, let $x \xrightarrow{a} x'$ and $y \xrightarrow{a} y'$; then $(x', y') \in eq(R)$, where $eq(R)$ is the least *equivalence* containing R.

Deterministic automata: bisimulation-up-to



Dashed lines + dotted line: bisimulation.

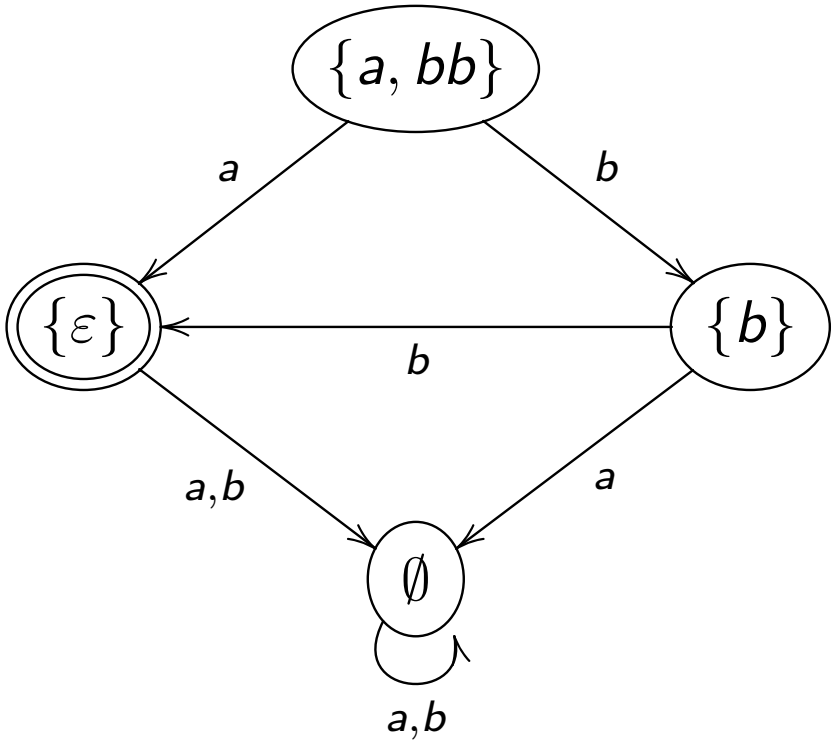
Only dashed lines: bisimulation up to equivalence.

Bisimulation up to equivalence: a relation R on states such that for each $(x, y) \in R$:

- x accepts iff y accepts
- for each alphabet letter a, let $x \xrightarrow{a} x'$ and $y \xrightarrow{a} y'$; then $(x', y') \in eq(R)$, where $eq(R)$ is the least *equivalence* containing R.

A special deterministic automaton

The set of all languages $\mathcal{P}(\Sigma^*)$ is *itself* a deterministic automaton.



$L \xrightarrow{a} L_a$ language derivative
L accepts iff $\varepsilon \in L$

The language accepted by a state L is precisely L itself.

So prove language equality by bisimulation(-up-to) and coinduction.

Arden's rule

If $L = KL + M$ and $\varepsilon \notin K$ then $L = K^*M$.

Arden's rule

If $L = KL + M$ and $\varepsilon \notin K$ then $L = K^*M$.

Proof: Let L, K, M be as above.

$$R = \{(L, K^*M)\}$$

Arden's rule

If $L = KL + M$ and $\varepsilon \notin K$ then $L = K^*M$.

Proof: Let L, K, M be as above.

$$R = \{(L, K^*M)\}$$

L accepts iff K^*M accepts (easy).

Arden's rule

If $L = KL + M$ and $\varepsilon \notin K$ then $L = K^*M$.

Proof: Let L, K, M be as above.

$$R = \{(L, K^*M)\}$$

L accepts iff K^*M accepts (easy). For any alphabet letter a:

$$\begin{array}{lll} L_a & = (KL + M)_a & \text{assumption} \\ & = K_aL + M_a & \text{Brzozowski, and assumption } \varepsilon \notin K \\ \\ (K^*M)_a & = K_aK^*M + M_a & \text{Brzozowski} \end{array}$$

Arden's rule

If $L = KL + M$ and $\varepsilon \notin K$ then $L = K^*M$.

Proof: Let L, K, M be as above.

$$R = \{(L, K^*M)\}$$

L accepts iff K^*M accepts (easy). For any alphabet letter a:

$$\begin{array}{lll} L_a & = (KL + M)_a & \text{assumption} \\ & = K_a L + M_a & \text{Brzozowski, and assumption } \varepsilon \notin K \\ \\ (K^*M)_a & = K_a K^*M + M_a & \text{Brzozowski} \end{array}$$

The derivatives are related by the *least congruence* containing $R \Longrightarrow$ bisimulation *up to congruence*.

Soundness

Let $\alpha : X \to FX$ be a coalgebra; bisimulation up to f is *sound* if $R \subseteq \sim$ whenever R is a bisimulation up to f.

- Bisimulation up to *union* with S is sound, if S contains only bisimilar pairs.
- Bisimulation up to *context* is sound, if the coalgebra is part of a λ-bialgebra [Bartels, PhD thesis, 2004].
- Bisimulation up to *bisimilarity* is ...

Soundness

Let $\alpha : X \to FX$ be a coalgebra; bisimulation up to f is *sound* if $R \subseteq \sim$ whenever R is a bisimulation up to f.

- Bisimulation up to *union* with S is sound, if S contains only bisimilar pairs.
- Bisimulation up to *context* is sound, if the coalgebra is part of a λ-bialgebra [Bartels, PhD thesis, 2004].
- Bisimulation up to *bisimilarity* is ... *not* sound in general!
- Bisimulation up to *equivalence* is ... not sound either.

Soundness

Let $\alpha : X \to FX$ be a coalgebra; bisimulation up to f is *sound* if $R \subseteq \sim$ whenever R is a bisimulation up to f.

- Bisimulation up to *union* with S is sound, if S contains only bisimilar pairs.
- Bisimulation up to *context* is sound, if the coalgebra is part of a λ-bialgebra [Bartels, PhD thesis, 2004].
- Bisimulation up to *bisimilarity* is ... *not* sound in general!
- Bisimulation up to *equivalence* is ... not sound either.

Solution: via *behavioural equivalence(-up-to)* these are again sound under a mild condition.

Concluding remarks

- Bisimulation-up-to often yields simpler, easier and more elegant coinductive proofs.
- Our contribution: generalize to the level of coalgebras, so applicable to many different types of systems: deterministic automata and languages, weighted automata, streams, ...
- Applied to language equations in (Rot, Bonsangue, Rutten LATA 2013).
- Current extension in joint work with Bonchi, Bonsangue, Pous, Rutten, Silva: *compatible* functions: compositionality.