Coalgebras and Coalgebra Automata

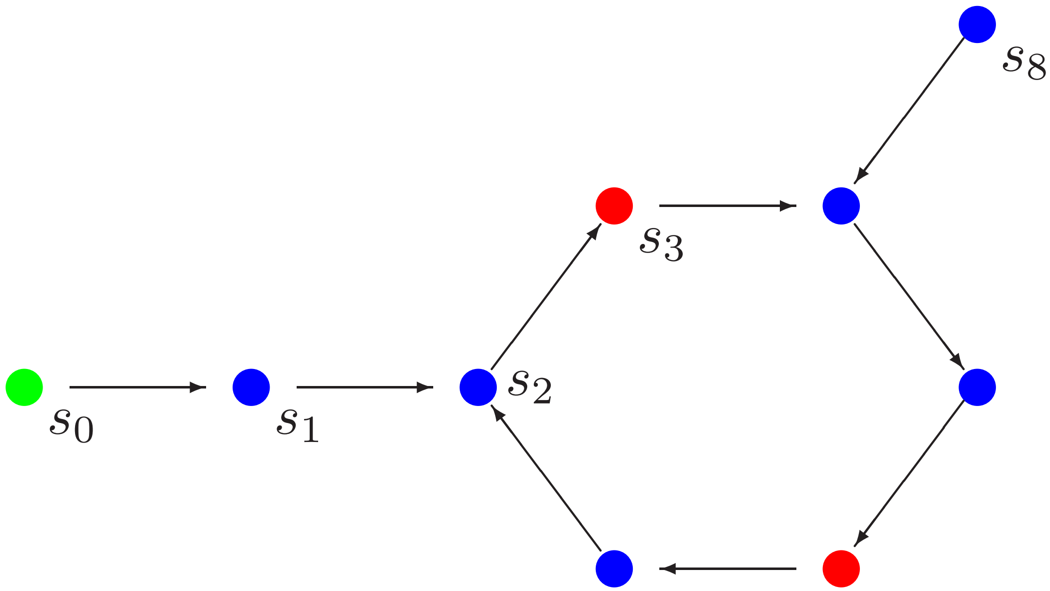
Yde Venema
ILLC-UvA
http://staff.science.uva.nl/~yde

October 13, 2006
Japan Advanced Institute of Science and Technology

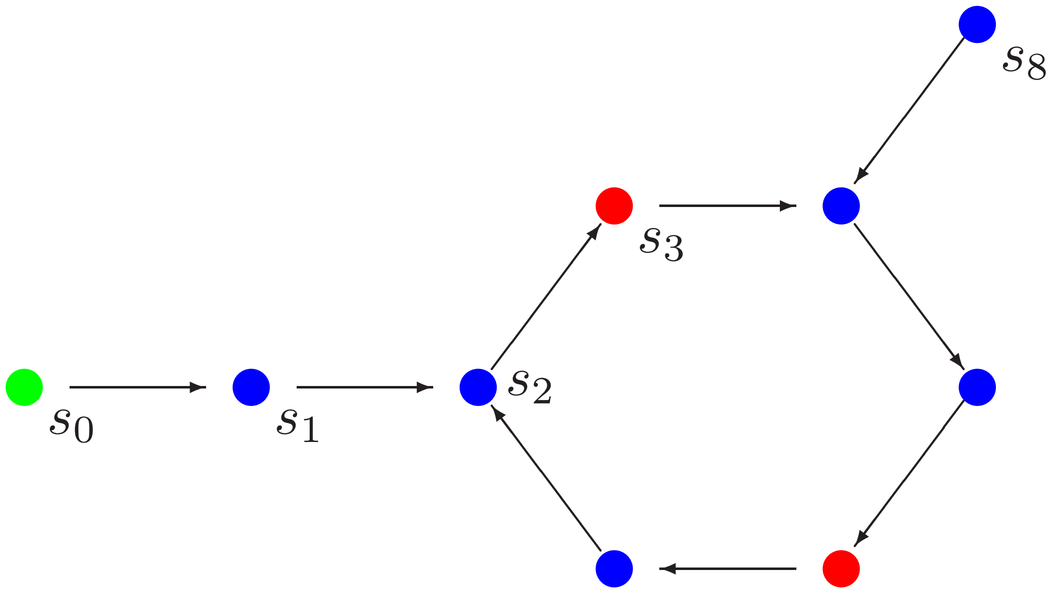
Overview of talk

- Examples
- Coalgebra
- Automata for coalgebras
- Finally, . . .

Streams

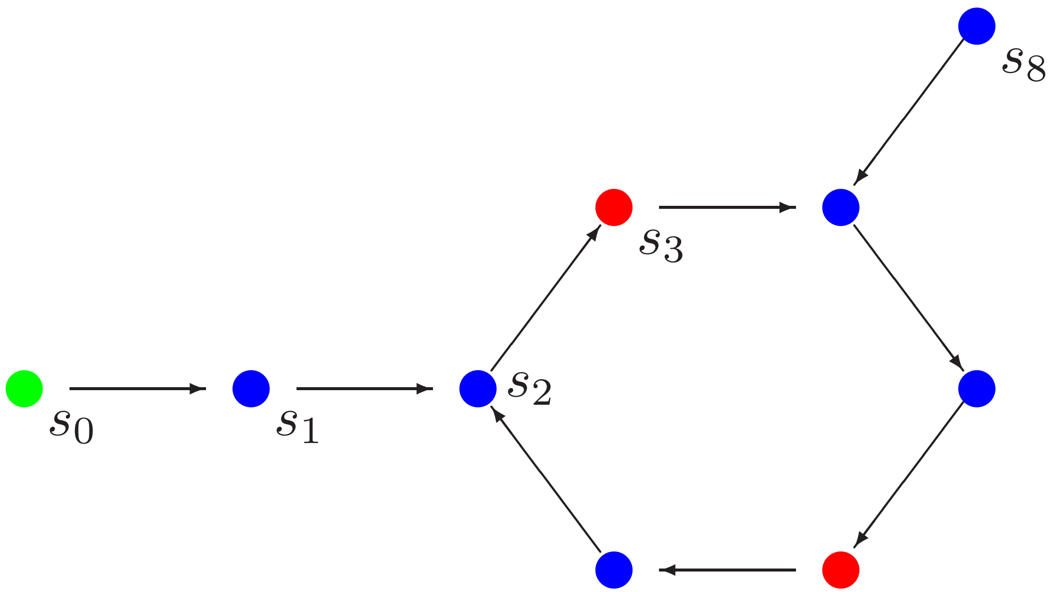


Streams



$\mathsf{Beh}(s_0) =$ ●●●●●●● $\cdots$

Streams

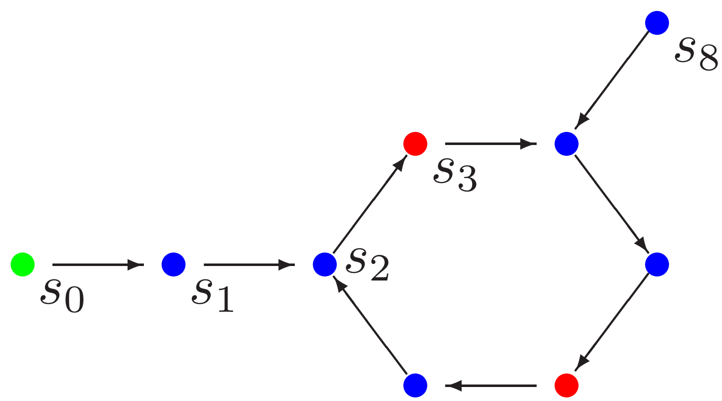


$\mathrm{Beh}(s_0) = {\color{green}\bullet}{\color{blue}\bullet}{\color{blue}\bullet}{\color{red}\bullet}{\color{blue}\bullet}{\color{blue}\bullet}{\color{red}\bullet}\cdots$

$\mathrm{Beh}(s_1) = {\color{blue}\bullet}{\color{blue}\bullet}{\color{red}\bullet}{\color{blue}\bullet}{\color{blue}\bullet}{\color{red}\bullet}{\color{blue}\bullet}\cdots$

$\mathrm{Beh}(s_8) = {\color{blue}\bullet}{\color{blue}\bullet}{\color{blue}\bullet}{\color{red}\bullet}{\color{blue}\bullet}{\color{blue}\bullet}{\color{red}\bullet}\cdots$

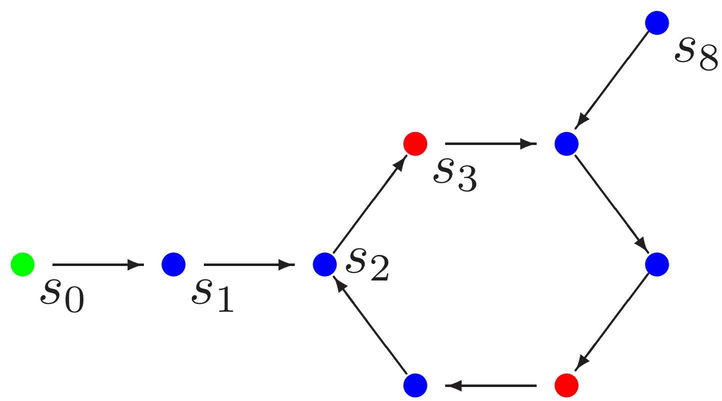
Streams as Coalgebras



A stream (stream coalgebra) is a pair $\mathbb{S} = \langle S, \sigma : S \to C \times S \rangle$.

E.g. model an infinite word $c_0 c_1 c_2 \ldots$ as $\langle \omega, \lambda n.(c_n, n+1) \rangle$.

Streams as Coalgebras

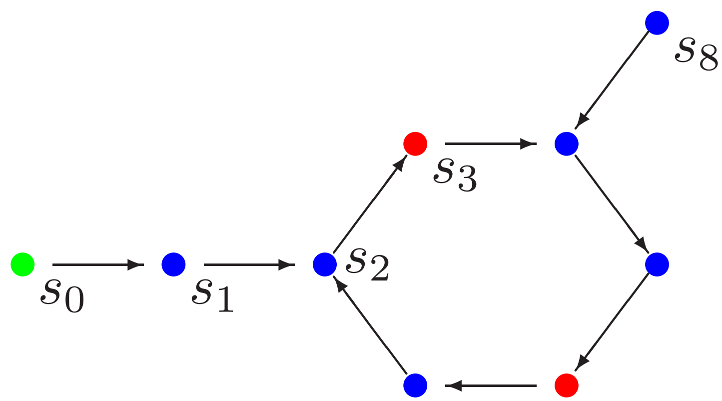


A stream (stream coalgebra) is a pair $\mathbb{S} = \langle S, \sigma : S \to C \times S \rangle$.

E.g. model an infinite word $c_0c_1c_2\ldots$ as $\langle \omega, \lambda n.(c_n, n+1) \rangle$.

Write $\sigma(s) = (\mathsf{col}(s), \mathsf{nxt}(s))$.

Streams as Coalgebras



A stream (stream coalgebra) is a pair $\mathbb{S} = \langle S, \sigma : S \to C \times S \rangle$.

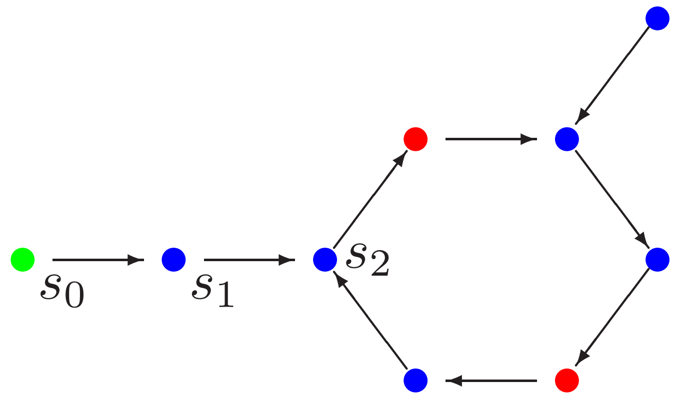
E.g. model an infinite word $c_0 c_1 c_2 \ldots$ as $\langle \omega, \lambda n.(c_n, n+1) \rangle$.

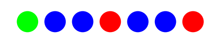
Write $\sigma(s) = (\mathsf{col}(s), \mathsf{nxt}(s))$.

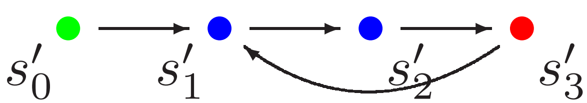
Behaviour: $\mathsf{Beh}(s) = \mathsf{col}(s)\mathsf{col}(\mathsf{nxt}(s))\mathsf{col}(\mathsf{nxt}^2(s)) \ldots$

Behavioral equivalence

Behavioral equivalence

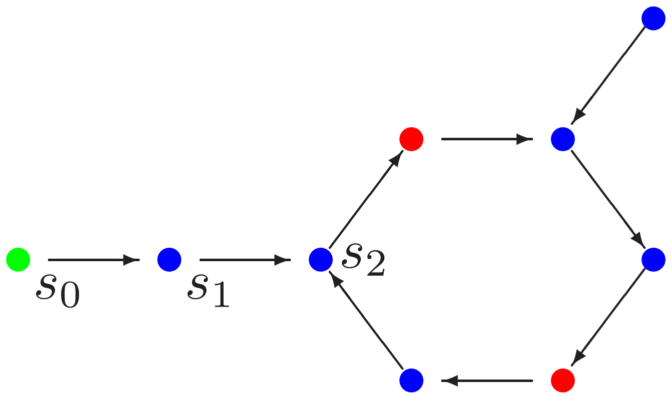


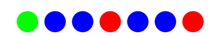
Beh(s_0) =  $\cdots$

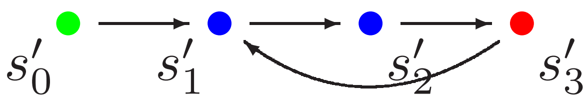


= Beh(s'_0)

Behavioral equivalence



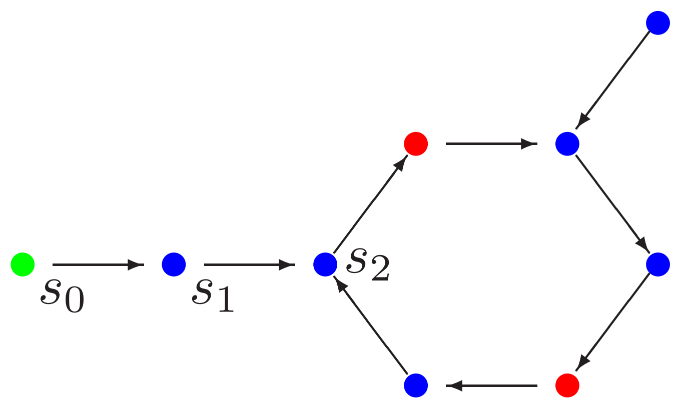
Beh(s_0) =  $\cdots$



= Beh(s'_0)

Definition: s in $\mathbb{S}$ is behaviorally equivalent to s' in $\mathbb{S}'$ if Beh(s) = Beh(s').

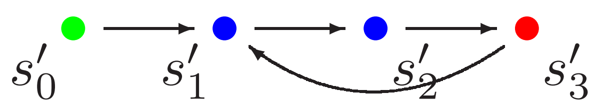
Bisimilarity



$Z \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in Z$:

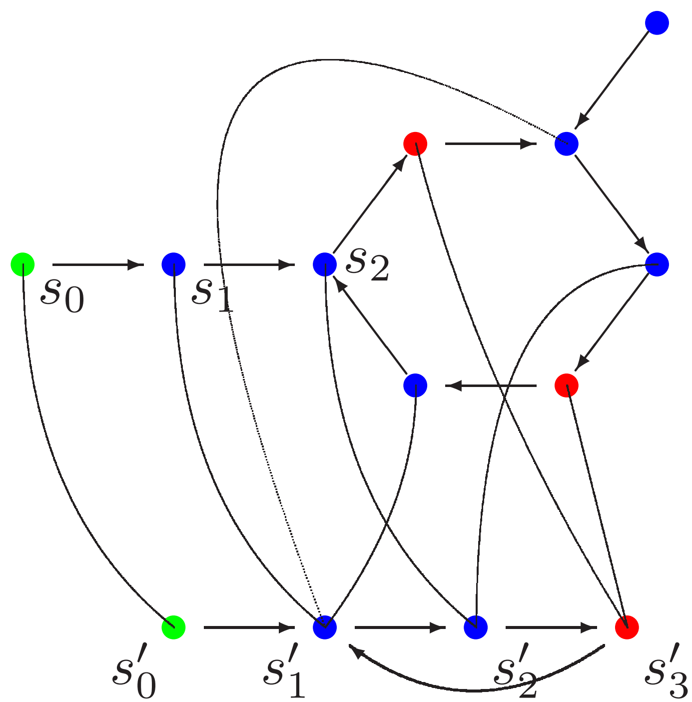
1. $\mathsf{col}(s) = \mathsf{col}(s')$,

2. $(\mathsf{nxt}(s), \mathsf{nxt}(s') \in Z$.



s in $\mathbb{S}$ and s' in $\mathbb{S}'$ are bisimilar if linked by some bisimulation Z.

Bisimilarity ct'd



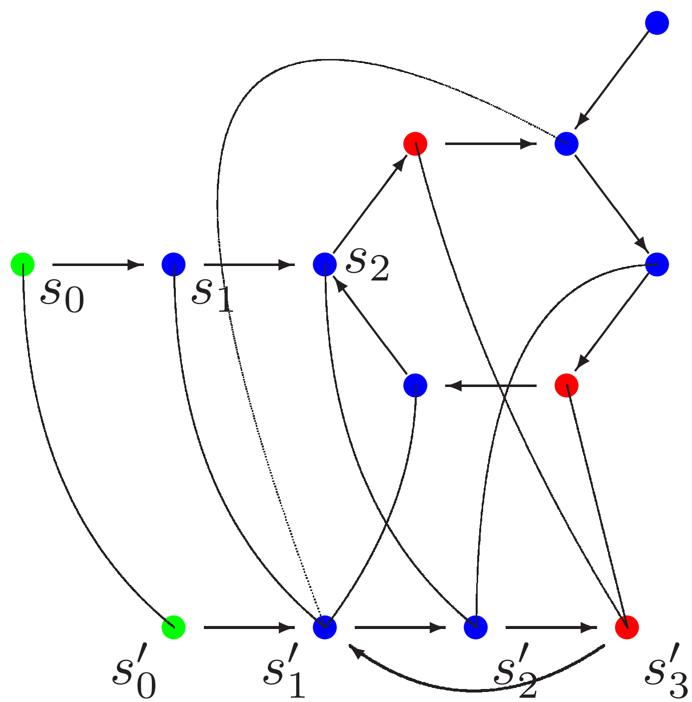
$Z \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in Z$:

1. $\mathsf{col}(s) = \mathsf{col}(s')$,

2. $(\mathsf{nxt}(s), \mathsf{nxt}(s') \in Z$.

s in $\mathbb{S}$ and s' in $\mathbb{S}'$ are bisimilar if linked by some bisimulation Z.

Bisimilarity ct'd



$Z \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in Z$:

1. $\mathsf{col}(s) = \mathsf{col}(s')$,

2. $(\mathsf{nxt}(s), \mathsf{nxt}(s') \in Z$.

s in $\mathbb{S}$ and s' in $\mathbb{S}'$ are bisimilar if linked by some bisimulation Z.

Theorem: bisimilarity = behavioral equivalence

Bistreams

Suppose now every state has two successors: a left and a right one.

Bistreams

Suppose now every state has two successors: a left and a right one.

A bistream is a pair $\mathbb{S} = \langle S, \sigma : S \to C \times S \times S \rangle$.
Write $\sigma(s) = (\mathsf{col}(s), \mathsf{lft}(s), \mathsf{rgt}(s))$.

Bistreams

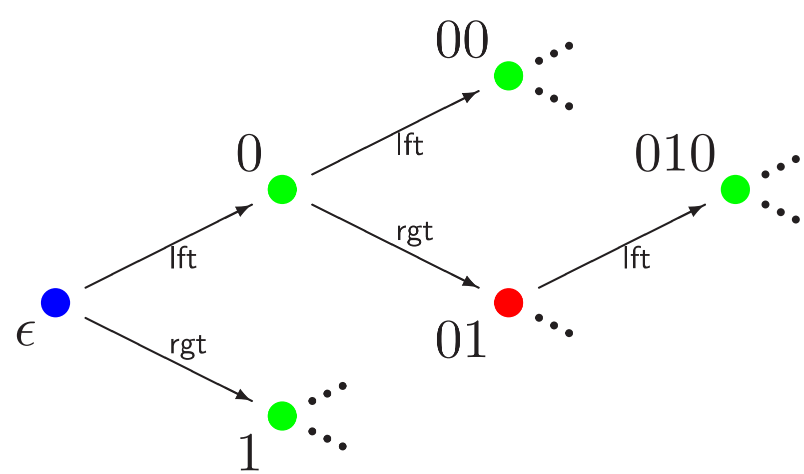
Suppose now every state has two successors: a left and a right one.

A bistream is a pair $\mathbb{S} = \langle S, \sigma : S \to C \times S \times S \rangle$.
Write $\sigma(s) = (\mathsf{col}(s), \mathsf{lft}(s), \mathsf{rgt}(s))$.

E.g. model an infinite C-labeled binary tree

$\gamma : \{0,1\}^* \to C$ as

$\langle \{0,1\}^*, \lambda s.(\gamma(s), s0, s1) \rangle$.



Bistreams bisimulations

Definition: Let $\mathbb{S}$ and $\mathbb{S}'$ be two bistreams.

$Z \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in Z$:

1. $\mathsf{col}(s) = \mathsf{col}(s')$,
2. $(\mathsf{lft}(s), \mathsf{lft}(s')) \in Z$ and $(\mathsf{rgt}(s), \mathsf{rgt}(s')) \in Z$,

Definition: An infinite C-labeled binary tree is regular iff it is bisimilar to a finite bistream

Kripke Models

Kripke structure: pair $\mathbb{S} = \langle S, \sigma : S \to \wp\mathsf{Prop} \times \wp S \rangle$, with

- $\sigma(s) = (\sigma_V(s), \sigma_R(s))$ given by

Kripke Models

Kripke structure: pair $\mathbb{S} = \langle S, \sigma : S \to \wp \text{Prop} \times \wp S \rangle$, with

- $\sigma(s) = (\sigma_V(s), \sigma_R(s))$ given by

  $\sigma_V(s) \subseteq \text{Prop}$ is the set of variables true at s,

Kripke Models

Kripke structure: pair $\mathbb{S} = \langle S, \sigma : S \to \wp\text{Prop} \times \wp S \rangle$, with

- $\sigma(s) = (\sigma_V(s), \sigma_R(s))$ given by

  $\sigma_V(s) \subseteq$ Prop is the set of variables true at s,

  $\sigma_R(s) \subseteq S$ is the set of successors of s.

Kripke Models

Kripke structure: pair $\mathbb{S} = \langle S, \sigma : S \to \wp\text{Prop} \times \wp S \rangle$, with

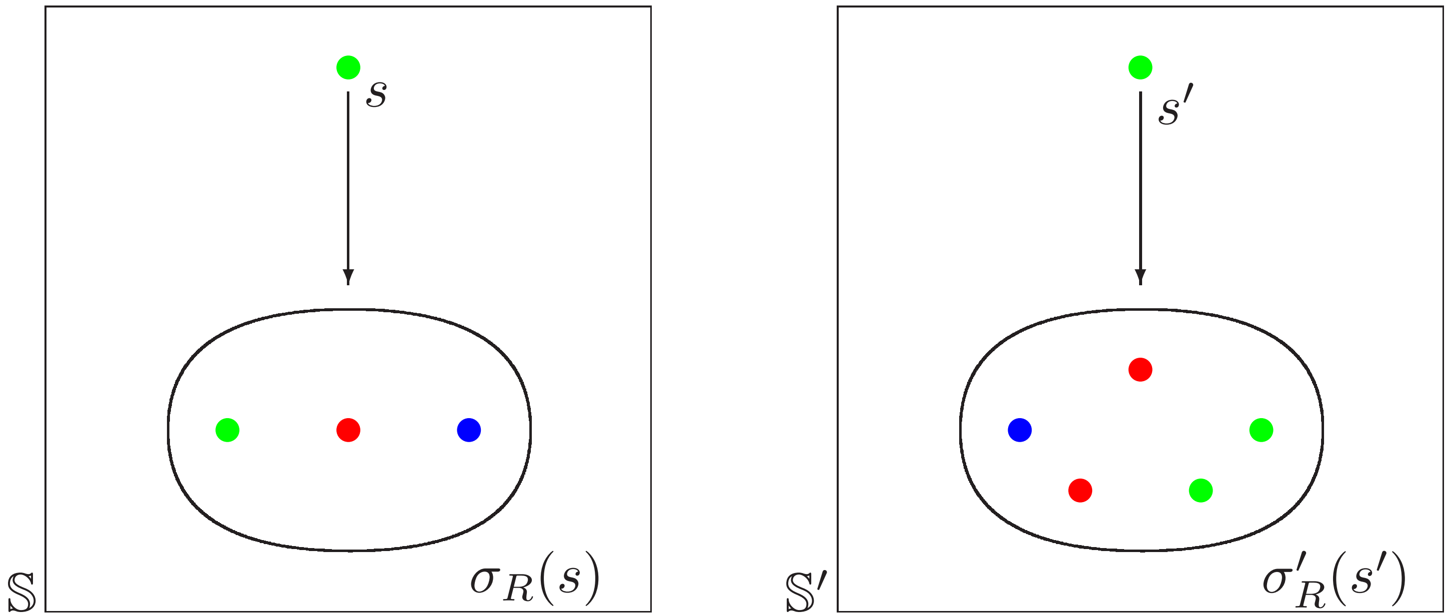
- $\sigma(s) = (\sigma_V(s), \sigma_R(s))$ given by

  $\sigma_V(s) \subseteq \text{Prop}$ is the set of variables true at s,

  $\sigma_R(s) \subseteq S$ is the set of successors of s.

Abbreviate $\mathsf{K}S := \wp\text{Prop} \times \wp S$.

Bisimulations



Bisimulations

Definition: Let $\mathbb{S}$ and $\mathbb{S}'$ be two Kripke structures.

- $Z \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in Z$:

Bisimulations

Definition: Let $\mathbb{S}$ and $\mathbb{S}'$ be two Kripke structures.

- $Z \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in Z$:

  1. $\sigma_V(s) = \sigma'_V(s')$

Bisimulations

Definition: Let $\mathbb{S}$ and $\mathbb{S}'$ be two Kripke structures.

- $Z \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in Z$:
  1. $\sigma_V(s) = \sigma'_V(s')$
  2. $(\sigma_R(s), \sigma'_R(s')) \in \overline{\wp}(Z)$, i.e.

Bisimulations

Definition: Let $\mathbb{S}$ and $\mathbb{S}'$ be two Kripke structures.

- $Z \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in Z$:

  1. $\sigma_V(s) = \sigma'_V(s')$
  2. $(\sigma_R(s), \sigma'_R(s')) \in \overline{\wp}(Z)$, i.e.
     - $\forall t \in \sigma_2(s) \exists t' \in \sigma'_2(s').Ztt'$ and
     - $\forall t' \in \sigma'_2(s') \exists t \in \sigma_2(s).Ztt'$.

Bisimulations

Definition: Let $\mathbb{S}$ and $\mathbb{S}'$ be two Kripke structures.

- $Z \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in Z$:
  1. $\sigma_V(s) = \sigma'_V(s')$
  2. $(\sigma_R(s), \sigma'_R(s')) \in \overline{\wp}(Z)$, i.e.
     - $\forall t \in \sigma_2(s) \exists t' \in \sigma'_2(s').Ztt'$ and
     - $\forall t' \in \sigma'_2(s') \exists t \in \sigma_2(s).Ztt'$.
- s and s' are bisimilar, notation: $\mathbb{S}, s \leftrightarrows \mathbb{S}', s'$
  if there is a bisimulation Z with $(s, s') \in Z$.

Bisimulations

Definition: Let $\mathbb{S}$ and $\mathbb{S}'$ be two Kripke structures.

- $Z \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in Z$:

  1. $\sigma_V(s) = \sigma'_V(s')$
  2. $(\sigma_R(s), \sigma'_R(s')) \in \overline{\wp}(Z)$, i.e.
     - $\forall t \in \sigma_2(s) \exists t' \in \sigma'_2(s').Ztt'$ and
     - $\forall t' \in \sigma'_2(s') \exists t \in \sigma_2(s).Ztt'$.

- s and s' are bisimilar, notation: $\mathbb{S}, s \leftrightarrows \mathbb{S}', s'$ if there is a bisimulation Z with $(s, s') \in Z$.

- $Z \subseteq S \times S'$ is a local bisimulation for $\sigma(s) \in \mathsf{K}S$ and $\sigma'(s') \in \mathsf{K}S'$ if (1) and (2) hold

Bisimulations

Definition: Let $\mathbb{S}$ and $\mathbb{S}'$ be two Kripke structures.

- $Z \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in Z$:

  1. $\sigma_V(s) = \sigma'_V(s')$
  2. $(\sigma_R(s), \sigma'_R(s')) \in \overline{\wp}(Z)$, i.e.
     - $\forall t \in \sigma_2(s) \exists t' \in \sigma'_2(s').Ztt'$ and
     - $\forall t' \in \sigma'_2(s') \exists t \in \sigma_2(s).Ztt'$.

- s and s' are bisimilar, notation: $\mathbb{S}, s \leftrightarroweq \mathbb{S}', s'$ if there is a bisimulation Z with $(s, s') \in Z$.

- $Z \subseteq S \times S'$ is a local bisimulation for $\sigma(s) \in \mathsf{K}S$ and $\sigma'(s') \in \mathsf{K}S'$ if (1) and (2) hold

Proposition: *Let $Z \subseteq S \times S'$ for two Kripke structures $\mathbb{S}$ and $\mathbb{S}'$.*
Z is a bisimulation iff it is a local bisimulation for $(\sigma(s), \sigma'(s'))$ whenever $(s, s') \in Z$.

Bisimilarity game

With Kripke structures $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{S}, \mathbb{S}')$ is played by $\exists$ and $\forall$:

Bisimilarity game

With Kripke structures $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{S}, \mathbb{S}')$ is played by $\exists$ and $\forall$:

- at position (s, s'), $\exists$ chooses a local bisimulation Z for $\sigma(s)$ and $\sigma'(s')$;

Bisimilarity game

With Kripke structures $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{S}, \mathbb{S}')$ is played by $\exists$ and $\forall$:

- at position (s, s'), $\exists$ chooses a local bisimulation Z for $\sigma(s)$ and $\sigma'(s')$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(t, t') \in Z$.

Bisimilarity game

With Kripke structures $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{S}, \mathbb{S}')$ is played by $\exists$ and $\forall$:

- at position (s, s'), $\exists$ chooses a local bisimulation Z for $\sigma(s)$ and $\sigma'(s')$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(t, t') \in Z$.

Winning conditions:

Bisimilarity game

With Kripke structures $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{S}, \mathbb{S}')$ is played by $\exists$ and $\forall$:

- at position (s, s'), $\exists$ chooses a local bisimulation Z for $\sigma(s)$ and $\sigma'(s')$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(t, t') \in Z$.

Winning conditions:

- finite matches are lost by the player who gets stuck,

Bisimilarity game

With Kripke structures $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{S}, \mathbb{S}')$ is played by $\exists$ and $\forall$:

- at position (s, s'), $\exists$ chooses a local bisimulation Z for $\sigma(s)$ and $\sigma'(s')$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(t, t') \in Z$.

Winning conditions:

- finite matches are lost by the player who gets stuck,
- infinite matches are won by $\exists$.

Bisimilarity game

With Kripke structures $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{S}, \mathbb{S}')$ is played by $\exists$ and $\forall$:

- at position (s, s'), $\exists$ chooses a local bisimulation Z for $\sigma(s)$ and $\sigma'(s')$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(t, t') \in Z$.

Winning conditions:

- finite matches are lost by the player who gets stuck,
- infinite matches are won by $\exists$.

Theorem: For all s, s': $(s, s') \in Win_{\exists}(\mathcal{B})$ iff $\mathbb{S}, s \leftrightarrows \mathbb{S}', s'$.

Overview

- Examples
- Coalgebra
- Automata for coalgebras
- Finally, . . .

Coalgebra

- Universal Coalgebra (Rutten, 2000) is
  a general mathematical theory for evolving systems

Coalgebra

- Universal Coalgebra (Rutten, 2000) is
  a general mathematical theory for evolving systems
- It provides a natural framework for notions like
  - behavior

Coalgebra

- Universal Coalgebra (Rutten, 2000) is
  a general mathematical theory for evolving systems
- It provides a natural framework for notions like
  - behavior
  - bisimulation/behavioral equivalence

Coalgebra

- Universal Coalgebra (Rutten, 2000) is
  a general mathematical theory for evolving systems

- It provides a natural framework for notions like
  - behavior
  - bisimulation/behavioral equivalence
  - invariants

Coalgebra

- Universal Coalgebra (Rutten, 2000) is
  a general mathematical theory for evolving systems

- It provides a natural framework for notions like

  - behavior
  - bisimulation/behavioral equivalence
  - invariants

- A coalgebra is a structure $\mathbb{S} = \langle S, \sigma : S \to \mathsf{F}S \rangle$,
  where F is the type of the coalgebra.

Coalgebra

- Universal Coalgebra (Rutten, 2000) is
  a general mathematical theory for evolving systems
- It provides a natural framework for notions like
  - behavior
  - bisimulation/behavioral equivalence
  - invariants
- A coalgebra is a structure $\mathbb{S} = \langle S, \sigma : S \to \mathsf{F}S \rangle$,
  where F is the type of the coalgebra.
- Sufficiently general to model notions like:
  input, output, non-determinism, interaction, probability, . . .

Coalgebra

- Universal Coalgebra (Rutten, 2000) is
  a general mathematical theory for evolving systems
- It provides a natural framework for notions like
  - behavior
  - bisimulation/behavioral equivalence
  - invariants
- A coalgebra is a structure $\mathbb{S} = \langle S, \sigma : S \to \mathsf{F}S \rangle$,
  where F is the type of the coalgebra.
- Sufficiently general to model notions like:
  input, output, non-determinism, interaction, probability, . . .
- A pointed F-coalgebra is a pair $(\mathbb{S}, s_0)$ where $\mathbb{S}$ is a coalgebra, and s_0 is a designated point in $\mathbb{S}$.

Examples

- streams: $\mathsf{F}S = C \times S$
- bi-streams: $\mathsf{F}S = C \times S \times S$
- Kripke frames: $\mathsf{F}S = \wp(S)$
- Kripke models: $\mathsf{F}S = \wp(\mathsf{Prop}) \times \wp(S)$

Examples

- streams: $\mathsf{F}S = C \times S$
- bi-streams: $\mathsf{F}S = C \times S \times S$
- Kripke frames: $\mathsf{F}S = \wp(S)$
- Kripke models: $\mathsf{F}S = \wp(\mathsf{Prop}) \times \wp(S)$
- finite words: $\mathsf{F}S = C \times (S \uplus \{\downarrow\})$

Examples

- streams: $\mathsf{F}S = C \times S$
- bi-streams: $\mathsf{F}S = C \times S \times S$
- Kripke frames: $\mathsf{F}S = \wp(S)$
- Kripke models: $\mathsf{F}S = \wp(\mathsf{Prop}) \times \wp(S)$
- finite words: $\mathsf{F}S = C \times (S \uplus \{\downarrow\})$
- finite trees: $\mathsf{F}S = C \times ((S \times S) \uplus \{\downarrow\})$

More examples

- deterministic automata: $\mathsf{F}S = \{0,1\} \times S^C$

More examples

- deterministic automata: $\mathsf{F}S = \{0,1\} \times S^C$
- labeled transition systems: $\mathsf{F}S = (\wp S)^A$

More examples

- deterministic automata: $\mathsf{F}S = \{0,1\} \times S^C$
- labeled transition systems: $\mathsf{F}S = (\wp S)^A$
- (non-wellfounded) sets: $\mathsf{F}S = \wp S$

More examples

- deterministic automata: $\mathsf{F}S = \{0, 1\} \times S^C$
- labeled transition systems: $\mathsf{F}S = (\wp S)^A$
- (non-wellfounded) sets: $\mathsf{F}S = \wp S$
- topologies: $\mathsf{F}S = \wp\wp(S)$

Coalgebras over other base categories than **Set**:

More examples

- deterministic automata: $\mathsf{F}S = \{0,1\} \times S^C$
- labeled transition systems: $\mathsf{F}S = (\wp S)^A$
- (non-wellfounded) sets: $\mathsf{F}S = \wp S$
- topologies: $\mathsf{F}S = \wp\wp(S)$

Coalgebras over other base categories than **Set**:

- Harsanyi type space (game theory),

More examples

- deterministic automata: $\mathsf{F}S = \{0,1\} \times S^C$
- labeled transition systems: $\mathsf{F}S = (\wp S)^A$
- (non-wellfounded) sets: $\mathsf{F}S = \wp S$
- topologies: $\mathsf{F}S = \wp\wp(S)$

Coalgebras over other base categories than **Set**:

- Harsanyi type space (game theory),
- descriptive general frames (modal logic)

More examples

- deterministic automata: $\mathsf{F}S = \{0,1\} \times S^C$
- labeled transition systems: $\mathsf{F}S = (\wp S)^A$
- (non-wellfounded) sets: $\mathsf{F}S = \wp S$
- topologies: $\mathsf{F}S = \wp\wp(S)$

Coalgebras over other base categories than **Set**:

- Harsanyi type space (game theory),
- descriptive general frames (modal logic)
- . . .

Algebra and Coalgebra

Relation between algebra & coalgebra characterized by both similarities and dualities

-

Algebra and Coalgebra

Relation between algebra & coalgebra characterized by both similarities and dualities

- construction vs observation
  $\mathbb{A} = \langle A, \alpha : A \leftarrow \mathsf{F}A \rangle$
  $\mathbb{C} = \langle C, \gamma : C \rightarrow \mathsf{F}C \rangle$

Algebra and Coalgebra

Relation between algebra & coalgebra characterized by both similarities and dualities

- construction vs observation
  $\mathbb{A} = \langle A, \alpha : A \leftarrow \mathsf{F}A \rangle$
  $\mathbb{C} = \langle C, \gamma : C \rightarrow \mathsf{F}C \rangle$
- congruence vs bisimulation

Algebra and Coalgebra

Relation between algebra & coalgebra characterized by both similarities and dualities

- construction vs observation
  $\mathbb{A} = \langle A, \alpha : A \leftarrow \mathsf{F}A \rangle$
  $\mathbb{C} = \langle C, \gamma : C \rightarrow \mathsf{F}C \rangle$
- congruence vs bisimulation
- induction vs coinduction

Algebra and Coalgebra

Relation between algebra & coalgebra characterized by both similarities and dualities

- construction vs observation
  $\mathbb{A} = \langle A, \alpha : A \leftarrow \mathsf{F}A \rangle$
  $\mathbb{C} = \langle C, \gamma : C \rightarrow \mathsf{F}C \rangle$
- congruence vs bisimulation
- induction vs coinduction
- varieties vs covarieties

Algebra and Coalgebra

Relation between algebra & coalgebra characterized by both similarities and dualities

- construction vs observation
  $\mathbb{A} = \langle A, \alpha : A \leftarrow \mathsf{F} A \rangle$
  $\mathbb{C} = \langle C, \gamma : C \rightarrow \mathsf{F} C \rangle$
- congruence vs bisimulation
- induction vs coinduction
- varieties vs covarieties
- . . .

Coalgebras and their morphisms

Definition: Let F be a set functor.

Coalgebras and their morphisms

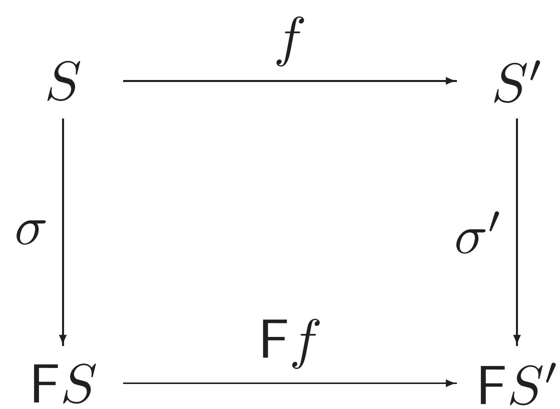
Definition: Let F be a set functor.

- An F-coalgebra is a pair $\mathbb{S} = \langle S, \sigma : S \to \mathsf{F}S \rangle$.

Coalgebras and their morphisms

Definition: Let F be a set functor.

- An F-coalgebra is a pair $\mathbb{S} = \langle S, \sigma : S \to \mathsf{F}S \rangle$.
- A coalgebra homomorphism between two coalgebras $\mathbb{S}$ and $\mathbb{S}'$ is a map $f : S \to S'$ such that $\sigma' \circ f = \mathsf{F}f \circ \sigma$:



Overview of talk

- Examples
- Coalgebra
- Automata for coalgebras
- Finally, . . .

Automata Theory

Automata Theory

- automata: finite devices classifying potentially infinite objects

Automata Theory

- automata: finite devices classifying potentially infinite objects
- strong connections with (fixpoint/second order) logic
  Slogan: formulas are automata

Automata Theory

- automata: finite devices classifying potentially infinite objects
- strong connections with (fixpoint/second order) logic
  Slogan: formulas are automata
- rich history: Büchi, Rabin, Janin & Walukiewicz, . . .

Automata Theory

- automata: finite devices classifying potentially infinite objects
- strong connections with (fixpoint/second order) logic
  Slogan: formulas are automata
- rich history: Büchi, Rabin, Janin & Walukiewicz, . . .
- applications in model checking

Automata Theory

- automata: finite devices classifying potentially infinite objects
- strong connections with (fixpoint/second order) logic
  Slogan: formulas are automata
- rich history: Büchi, Rabin, Janin & Walukiewicz, . . .
- applications in model checking
- here: coalgebraic perspective

Automata Theory

- automata: finite devices classifying potentially infinite objects
- strong connections with (fixpoint/second order) logic
  Slogan: formulas are automata
- rich history: Büchi, Rabin, Janin & Walukiewicz, . . .
- applications in model checking
- here: coalgebraic perspective

Claim: Coalgebra is a natural level of generality for studying automata

(Local) Bisimulation revisited

Definition: Let $\mathbb{A}$ and $\mathbb{S}$ be two coalgebras of type F, let $\alpha \in \mathsf{F}A$ and $\sigma \in \mathsf{F}S$.

(Local) Bisimulation revisited

Definition: Let $\mathbb{A}$ and $\mathbb{S}$ be two coalgebras of type F, let $\alpha \in \mathsf{F}A$ and $\sigma \in \mathsf{F}S$.

Then $Z \subseteq A \times S$ is a local bisimulation for α and σ, if . . .

Relation lifting

Fix a coalgebra type F.

Fact: Any relation $Z \subseteq S \times S'$ can be lifted to a relation $\overline{\mathsf{F}}(Z) \subseteq \mathsf{F}S \times \mathsf{F}S'$.

Relation lifting

Fix a coalgebra type F.

Fact: Any relation $Z \subseteq S \times S'$ can be lifted to a relation $\overline{\mathsf{F}}(Z) \subseteq \mathsf{F}S \times \mathsf{F}S'$.

Examples

streams $(\mathsf{F}S = C \times S)$
$((c,t),(c',t')) \in \overline{\mathsf{F}}(Z)$ iff $c = c'$ & $(t,t') \in Z$.

Relation lifting

Fix a coalgebra type F.

Fact: Any relation $Z \subseteq S \times S'$ can be lifted to a relation $\overline{\mathsf{F}}(Z) \subseteq \mathsf{F}S \times \mathsf{F}S'$.

Examples

streams ($\mathsf{F}S = C \times S$)
$((c,t),(c',t')) \in \overline{\mathsf{F}}(Z)$ iff $c = c'$ & $(t,t') \in Z$.

bistreams ($\mathsf{F}S = C \times S \times S$)
$((c,t_L,t_R),(c',t'_L,t'_R)) \in \overline{\mathsf{F}}(Z)$ iff $c = c'$ & $((t_L,t'_L),(t_R,t'_R)) \in Z$.

Relation lifting

Fix a coalgebra type F.

Fact: Any relation $Z \subseteq S \times S'$ can be lifted to a relation $\overline{\mathsf{F}}(Z) \subseteq \mathsf{F}S \times \mathsf{F}S'$.

Examples

streams $(\mathsf{F}S = C \times S)$
$((c,t),(c',t')) \in \overline{\mathsf{F}}(Z)$ iff $c = c'$ & $(t,t') \in Z$.

bistreams $(\mathsf{F}S = C \times S \times S)$
$((c,t_L,t_R),(c',t'_L,t'_R)) \in \overline{\mathsf{F}}(Z)$ iff $c = c'$ & $((t_L,t'_L),(t_R,t'_R)) \in Z$.

Kripke models $(\mathsf{F}S = \wp(\mathsf{Prop}) \times \wp(S))$
$((\pi,T),(\pi',T')) \in \overline{\mathsf{F}}(Z)$ iff $\pi = \pi'$ & $(T,T') \in \overline{\wp}(Z)$.

Bisimulation via relation lifting

Definition: Let $\mathbb{A}$ and $\mathbb{S}$ be two coalgebras of type F, let $\alpha \in \mathsf{F}A$ and $\sigma \in \mathsf{F}S$.

Bisimulation via relation lifting

Definition: Let $\mathbb{A}$ and $\mathbb{S}$ be two coalgebras of type F, let $\alpha \in \mathsf{F}A$ and $\sigma \in \mathsf{F}S$.

Then $Z \subseteq A \times S$ is a local bisimulation for α and σ, if $(\alpha, \sigma) \in \overline{\mathsf{F}}(Z)$.

Bisimulation via relation lifting

Definition: Let $\mathbb{A}$ and $\mathbb{S}$ be two coalgebras of type F, let $\alpha \in \mathsf{F}A$ and $\sigma \in \mathsf{F}S$.

Then $Z \subseteq A \times S$ is a local bisimulation for α and σ, if $(\alpha, \sigma) \in \overline{\mathsf{F}}(Z)$.

$Z \subseteq A \times S$ is a bisimulation iff $(\alpha(a), \sigma(s)) \in \overline{\mathsf{F}}(Z)$ whenever $(a, s) \in Z$.

Bisimilarity game

With F-coalgebras $\mathbb{A} = \langle A, \alpha \rangle$ and $\mathbb{S} = \langle S, \sigma \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{A}, \mathbb{S})$ is played by $\exists$ and $\forall$:

Bisimilarity game

With F-coalgebras $\mathbb{A} = \langle A, \alpha \rangle$ and $\mathbb{S} = \langle S, \sigma \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{A}, \mathbb{S})$ is played by $\exists$ and $\forall$:

- at position (a, s), $\exists$ chooses a local bisimulation Z for $\alpha(a)$ and $\sigma(s)$;

Bisimilarity game

With F-coalgebras $\mathbb{A} = \langle A, \alpha \rangle$ and $\mathbb{S} = \langle S, \sigma \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{A}, \mathbb{S})$ is played by $\exists$ and $\forall$:

- at position (a, s), $\exists$ chooses a local bisimulation Z for $\alpha(a)$ and $\sigma(s)$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(b, t) \in Z$

Bisimilarity game

With F-coalgebras $\mathbb{A} = \langle A, \alpha \rangle$ and $\mathbb{S} = \langle S, \sigma \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{A}, \mathbb{S})$ is played by $\exists$ and $\forall$:

- at position (a, s), $\exists$ chooses a local bisimulation Z for $\alpha(a)$ and $\sigma(s)$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(b, t) \in Z$

Winning conditions:

Bisimilarity game

With F-coalgebras $\mathbb{A} = \langle A, \alpha \rangle$ and $\mathbb{S} = \langle S, \sigma \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{A}, \mathbb{S})$ is played by $\exists$ and $\forall$:

- at position (a, s), $\exists$ chooses a local bisimulation Z for $\alpha(a)$ and $\sigma(s)$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(b, t) \in Z$

Winning conditions:

- finite matches are lost by the player who gets stuck,

Bisimilarity game

With F-coalgebras $\mathbb{A} = \langle A, \alpha \rangle$ and $\mathbb{S} = \langle S, \sigma \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{A}, \mathbb{S})$ is played by $\exists$ and $\forall$:

- at position (a, s), $\exists$ chooses a local bisimulation Z for $\alpha(a)$ and $\sigma(s)$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(b, t) \in Z$

Winning conditions:

- finite matches are lost by the player who gets stuck,
- infinite matches are won by $\exists$.

Bisimilarity game

With F-coalgebras $\mathbb{A} = \langle A, \alpha \rangle$ and $\mathbb{S} = \langle S, \sigma \rangle$,

the bisimilarity game $\mathcal{B}(\mathbb{A}, \mathbb{S})$ is played by $\exists$ and $\forall$:

- at position (a, s), $\exists$ chooses a local bisimulation Z for $\alpha(a)$ and $\sigma(s)$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(b, t) \in Z$

Winning conditions:

- finite matches are lost by the player who gets stuck,
- infinite matches are won by $\exists$.

Theorem: For all a, s: $(a, s) \in Win_{\exists}(\mathcal{B})$ iff $\mathbb{A}, a \leftrightarroweq \mathbb{S}, s$.

Change of perspective

Change of perspective

Think of one structure ($\mathbb{A}$) as classifying the other ($\mathbb{S}$):

Change of perspective

Think of one structure ($\mathbb{A}$) as classifying the other ($\mathbb{S}$):

Turn $\mathbb{A}$ into an automaton by the following three modifications:

Change of perspective

Think of one structure ($\mathbb{A}$) as classifying the other ($\mathbb{S}$):

Turn $\mathbb{A}$ into an automaton by the following three modifications:

- A must be finite

Change of perspective

Think of one structure ($\mathbb{A}$) as classifying the other ($\mathbb{S}$):

Turn $\mathbb{A}$ into an automaton by the following three modifications:

- A must be finite
- Some infinite matches may be won by $\forall$

Change of perspective

Think of one structure ($\mathbb{A}$) as classifying the other ($\mathbb{S}$):

Turn $\mathbb{A}$ into an automaton by the following three modifications:

- A must be finite
- Some infinite matches may be won by $\forall$

  Implement by (Büchi/Muller/. . .) acceptance condition $Acc \subseteq A^\omega$.

Change of perspective

Think of one structure ($\mathbb{A}$) as classifying the other ($\mathbb{S}$):

Turn $\mathbb{A}$ into an automaton by the following three modifications:

- A must be finite
- Some infinite matches may be won by $\forall$

  Implement by (Büchi/Muller/. . .) acceptance condition $Acc \subseteq A^{\omega}$.
- Introduce nondeterminism on $\mathbb{A} = \langle A, \Delta, Acc \rangle$ by putting $\Delta : A \to \wp \mathsf{F} A$.

Change of perspective

Think of one structure ($\mathbb{A}$) as classifying the other ($\mathbb{S}$):

Turn $\mathbb{A}$ into an automaton by the following three modifications:

- A must be finite
- Some infinite matches may be won by $\forall$

  Implement by (Büchi/Muller/. . .) acceptance condition $Acc \subseteq A^{\omega}$.
- Introduce nondeterminism on $\mathbb{A} = \langle A, \Delta, Acc \rangle$ by putting $\Delta : A \to \wp \mathsf{F} A$.

Definition A coalgebra automaton of type F is a triple $\mathbb{A} = \langle A, \Delta, Acc \rangle$.

Acceptance game

With $\mathbb{A}$ an F-automaton and $\mathbb{S}$ an F-coalgebra, $\exists$ and $\forall$ may play the following acceptance game $\mathcal{B}(\mathbb{A}, \mathbb{S})$:

Acceptance game

With $\mathbb{A}$ an F-automaton and $\mathbb{S}$ an F-coalgebra, $\exists$ and $\forall$ may play the following acceptance game $\mathcal{B}(\mathbb{A}, \mathbb{S})$:

- at position $(a, s) \in A \times S$, $\exists$ chooses an element α of $\Delta(a)$; the new position is (α, s);

Acceptance game

With $\mathbb{A}$ an F-automaton and $\mathbb{S}$ an F-coalgebra, $\exists$ and $\forall$ may play the following acceptance game $\mathcal{B}(\mathbb{A}, \mathbb{S})$:

- at position $(a, s) \in A \times S$, $\exists$ chooses an element α of $\Delta(a)$; the new position is (α, s);
- at position $(\alpha, s) \in \mathsf{F}A \times S$, $\exists$ chooses a local bisimulation Z for α and $\sigma(s)$;

Acceptance game

With $\mathbb{A}$ an F-automaton and $\mathbb{S}$ an F-coalgebra, $\exists$ and $\forall$ may play the following acceptance game $\mathcal{B}(\mathbb{A}, \mathbb{S})$:

- at position $(a, s) \in A \times S$, $\exists$ chooses an element α of $\Delta(a)$; the new position is (α, s);
- at position $(\alpha, s) \in \mathsf{F}A \times S$, $\exists$ chooses a local bisimulation Z for α and $\sigma(s)$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(b, t) \in Z$

Acceptance game

With $\mathbb{A}$ an F-automaton and $\mathbb{S}$ an F-coalgebra, $\exists$ and $\forall$ may play the following acceptance game $\mathcal{B}(\mathbb{A}, \mathbb{S})$:

- at position $(a, s) \in A \times S$, $\exists$ chooses an element α of $\Delta(a)$; the new position is (α, s);
- at position $(\alpha, s) \in \mathsf{F}A \times S$, $\exists$ chooses a local bisimulation Z for α and $\sigma(s)$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(b, t) \in Z$

Winning conditions:

Acceptance game

With $\mathbb{A}$ an F-automaton and $\mathbb{S}$ an F-coalgebra, $\exists$ and $\forall$ may play the following acceptance game $\mathcal{B}(\mathbb{A}, \mathbb{S})$:

- at position $(a, s) \in A \times S$, $\exists$ chooses an element α of $\Delta(a)$; the new position is (α, s);
- at position $(\alpha, s) \in \mathsf{F}A \times S$, $\exists$ chooses a local bisimulation Z for α and $\sigma(s)$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(b, t) \in Z$

Winning conditions:

- finite matches are lost by the player who gets stuck,

Acceptance game

With $\mathbb{A}$ an F-automaton and $\mathbb{S}$ an F-coalgebra, $\exists$ and $\forall$ may play the following acceptance game $\mathcal{B}(\mathbb{A}, \mathbb{S})$:

- at position $(a, s) \in A \times S$, $\exists$ chooses an element α of $\Delta(a)$; the new position is (α, s);
- at position $(\alpha, s) \in \mathsf{F}A \times S$, $\exists$ chooses a local bisimulation Z for α and $\sigma(s)$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(b, t) \in Z$

Winning conditions:

- finite matches are lost by the player who gets stuck,
- infinite matches are won as specified by Acc.

Acceptance game

With $\mathbb{A}$ an F-automaton and $\mathbb{S}$ an F-coalgebra, $\exists$ and $\forall$ may play the following acceptance game $\mathcal{B}(\mathbb{A}, \mathbb{S})$:

- at position $(a, s) \in A \times S$, $\exists$ chooses an element α of $\Delta(a)$; the new position is (α, s);
- at position $(\alpha, s) \in \mathsf{F}A \times S$, $\exists$ chooses a local bisimulation Z for α and $\sigma(s)$;
- at position $Z \subseteq A \times S$, $\forall$ chooses $(b, t) \in Z$

Winning conditions:

- finite matches are lost by the player who gets stuck,
- infinite matches are won as specified by Acc.

Definition: A pointed F-automaton $(\mathbb{A}, a)$ accepts a pointed F-coalgebra $(\mathbb{S}, s)$ if $(a, s) \in Win_{\exists}(\mathcal{B}(\mathbb{A}, \mathbb{S}))$.

Coalgebra automata

Existing automata on words, trees, graphs, etc, are all special instances of coalgebra automata.

Coalgebra automata

Existing automata on words, trees, graphs, etc, are all special instances of coalgebra automata.

Nondeterministic tree automata are usually presented as
$\mathbb{A} = \langle A, \Delta : A \times C \to \wp(A \times A), Acc \rangle$.

Coalgebra automata

Existing automata on words, trees, graphs, etc, are all special instances of coalgebra automata.

Nondeterministic tree automata are usually presented as
$\mathbb{A} = \langle A, \Delta : A \times C \to \wp(A \times A), Acc\rangle$.

This is equivalent to our presentation where $\Delta : A \to \wp(C \times A \times A)$:

Coalgebra automata

Existing automata on words, trees, graphs, etc, are all special instances of coalgebra automata.

Nondeterministic tree automata are usually presented as
$\mathbb{A} = \langle A, \Delta : A \times C \to \wp(A \times A), Acc \rangle$.

This is equivalent to our presentation where $\Delta : A \to \wp(C \times A \times A)$:

$$
\begin{aligned}
(A \times C) \to \wp(A \times A) &\cong (A \times C) \to ((A \times A) \to 2) \\
&\cong (A \times C \times A \times A) \to 2 \\
&\cong A \to (C \times A \times A) \to 2) \\
&\cong A \to \wp(C \times A \times A)
\end{aligned}
$$

General picture

Key observations

General picture

Key observations

- Coalgebra automata are similar to the coalgebras themselves:

$$
\begin{array}{lcll}
\text{F-coalgebra} & \mathbb{S} & = & \langle S, \sigma : S \to \mathsf{F}S \rangle \\
\text{F-automaton} & \mathbb{A} & = & \langle A, \Delta : A \to \wp(\mathsf{F}A), Acc \rangle
\end{array}
$$

General picture

Key observations

- Coalgebra automata are similar to the coalgebras themselves:

$$\begin{array}{llll} \text{F-coalgebra} & \mathbb{S} & = & \langle S, \sigma : S \to \mathsf{F}S \rangle \\ \text{F-automaton} & \mathbb{A} & = & \langle A, \Delta : A \to \wp(\mathsf{F}A), Acc \rangle \end{array}$$

- Acceptance generalizes bisimilarity.

General picture

Key observations

- Coalgebra automata are similar to the coalgebras themselves:

F-coalgebra	$\mathbb{S}$	$=$	$\langle S, \sigma : S \to \mathsf{F}S \rangle$
F-automaton	$\mathbb{A}$	$=$	$\langle A, \Delta : A \to \wp(\mathsf{F}A), Acc \rangle$

- Acceptance generalizes bisimilarity.
- Separate the combinatorics (Acc) from the dynamics (Δ).

Results in Universal Coalgebra

(The following are joint results with Clemens Kupke)

Theorem: Let F be an arbitrary set functor preserving weak pullbacks.

Results in Universal Coalgebra

(The following are joint results with Clemens Kupke)

Theorem: Let F be an arbitrary set functor preserving weak pullbacks.

- Finite Model Property If an nondeterministic F-automaton $\mathbb{A}$ accepts some coalgebra then it accepts a finite one.

Results in Universal Coalgebra

(The following are joint results with Clemens Kupke)

Theorem: Let F be an arbitrary set functor preserving weak pullbacks.

- Finite Model Property If an nondeterministic F-automaton $\mathbb{A}$ accepts some coalgebra then it accepts a finite one.
- Closure Properties The recognizable languages (classes of F-coalgebras that are accepted by some F-automaton) are closed under
  - union
  - intersection
  - projection modulo bisimulation

Results in Universal Coalgebra

(The following are joint results with Clemens Kupke)

Theorem: Let F be an arbitrary set functor preserving weak pullbacks.

- Finite Model Property If an nondeterministic F-automaton $\mathbb{A}$ accepts some coalgebra then it accepts a finite one.
- Closure Properties The recognizable languages (classes of F-coalgebras that are accepted by some F-automaton) are closed under
  - union
  - intersection
  - projection modulo bisimulation
- Determinization There is a construction transforming an alternating F-automaton into an equivalent nondeterministic one.

Results in Universal Coalgebra

(The following are joint results with Clemens Kupke)

Theorem: Let F be an arbitrary set functor preserving weak pullbacks.

- Finite Model Property If an nondeterministic F-automaton $\mathbb{A}$ accepts some coalgebra then it accepts a finite one.
- Closure Properties The recognizable languages (classes of F-coalgebras that are accepted by some F-automaton) are closed under
  - union
  - intersection
  - projection modulo bisimulation
- Determinization There is a construction transforming an alternating F-automaton into an equivalent nondeterministic one.
- Logic The above results have various corollaries in fixpoint logics.

Finally, . . .

What is this good for?

Finally, . . .

What is this good for?

- conceptual clarification

Finally, . . .

What is this good for?

- conceptual clarification
  - separate combinatorics from dynamics

Finally, . . .

What is this good for?

- conceptual clarification
  - separate combinatorics from dynamics
  - right level of abstraction

Finally, . . .

What is this good for?

- conceptual clarification
  - separate combinatorics from dynamics
  - right level of abstraction
  - coalgebraic automata constructions

Finally, . . .

What is this good for?

- conceptual clarification
  - separate combinatorics from dynamics
  - right level of abstraction
  - coalgebraic automata constructions
- uniform proofs

Finally, . . .

What is this good for?

- conceptual clarification
  - separate combinatorics from dynamics
  - right level of abstraction
  - coalgebraic automata constructions
- uniform proofs
- apply theory in new settings

Some reading

Some reading

- B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction *Bull. of the European Association for Theoretical Computer Science*, 62:222–259, 1997.

Some reading

- B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction
  Bull. of the European Association for Theoretical Computer Science, 62:222–259, 1997.
- J. Rutten. Universal coalgebra: A theory of systems
  Theoretical Computer Science, 249:3–80, 2000.

Some reading

- B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction
  Bull. of the European Association for Theoretical Computer Science, 62:222–259, 1997.
- J. Rutten. Universal coalgebra: A theory of systems
  Theoretical Computer Science, 249:3–80, 2000.
- A. Kurz. Coalgebras and modal logic
  Lecture Notes for ESSLLI'01; available from
  `http://www.cs.le.ac.uk/people/akurz`.
- Y. Venema. Algebras and Coalgebras
  In van Benthem, Blackburn & Wolter (eds.), *Handbook of Modal Logic*, 2006.

Some reading

- B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction
  Bull. of the European Association for Theoretical Computer Science, 62:222–259, 1997.
- J. Rutten. Universal coalgebra: A theory of systems
  Theoretical Computer Science, 249:3–80, 2000.
- A. Kurz. Coalgebras and modal logic
  Lecture Notes for ESSLLI'01; available from
  `http://www.cs.le.ac.uk/people/akurz`.
- Y. Venema. Algebras and Coalgebras
  In van Benthem, Blackburn & Wolter (eds.), *Handbook of Modal Logic*, 2006.
- Y. Venema. Automata and fixed point logic: a coalgebraic perspective
  Information and Computation, **204** (2006) 637–678.
- C. Kupke and Y. Venema. Closure properties of coalgebra automata
  LICS 2005, 199–208.