Universal Coalgebra - An Introductory Tutorial

H. Peter Gumm

Philipps-Universität Marburg

August 8, 2013

Outline

Systems

Can we get a general theory for all of the following Systems?

- Deterministic
  - Black Boxes
    - ★ w/o errors
  - Automata
    - ★ Moore, Mealy
  - Objects
    - ★ Java, C++, ...
- Nondeterministic systems
  - Automata
  - Kripke Structures
- Fuzzy Systems
  - Probabilistics Systems
  - Fuzzy Systems
  - Topologies
  - Neighbourhood systems
- and more ...

Systems

Can we get a general theory for all of the following Systems?

- Deterministic
  - Black Boxes
    - ★ w/o errors
  - Automata
    - ★ Moore, Mealy
  - Objects
    - ★ Java, C++, ...
- Nondeterministic systems
  - Automata
  - Kripke Structures
- Fuzzy Systems
  - Probabilistics Systems
  - Fuzzy Systems
  - Topologies
  - Neighbourhood systems
- and more ...

Systems

Can we get a general theory for all of the following Systems?

- Deterministic
  - Black Boxes
    - ★ w/o errors
  - Automata
    - ★ Moore, Mealy
  - Objects
    - ★ Java, C++, ...
- Nondeterministic systems
  - Automata
  - Kripke Structures
- Fuzzy Systems
  - Probabilistics Systems
  - Fuzzy Systems
  - Topologies
  - Neighbourhood systems
- and more ...

State Based

- Behaviour
  - depends on
    - ★ input
    - ★ internal state
  - visible only
    - ★ outputs
- Implementation concerns
  - Observational equivalence
  - Minimization
- Logics
  - global logics
  - local (modal) logic

State Based

- Behaviour
  - depends on
    - ★ input
    - ★ internal state
  - visible only
    - ★ outputs
- Implementation concerns
  - Observational equivalence
  - Minimization
- Logics
  - global logics
  - local (modal) logic

State Based

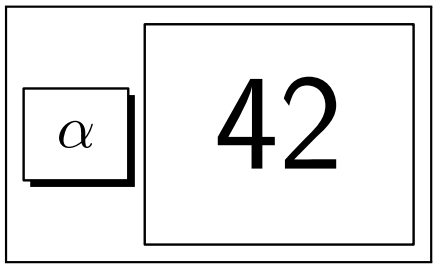
- Behaviour
  - depends on
    - ★ input
    - ★ internal state
  - visible only
    - ★ outputs
- Implementation concerns
  - Observational equivalence
  - Minimization
- Logics
  - global logics
  - local (modal) logic

State Based

- Behaviour
  - depends on
    - ★ input
    - ★ internal state
  - visible only
    - ★ outputs
- Implementation concerns
  - Observational equivalence
  - Minimization
- Logics
  - global logics
  - local (modal) logic

Black Boxes

- *Black box* with two buttons and a display...



- Signature:

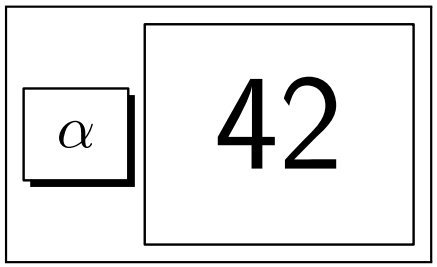
- Peeking inside

- Indistinguishability relation: largest $\sim$ with

$$\frac{s \sim s'}{h(s)=h(s'),\ t(s) \sim t(s')}$$

Black Boxes

- *Black box* with two buttons and a display...



- Signature:

$$\alpha \quad : \quad S \to D \times S$$

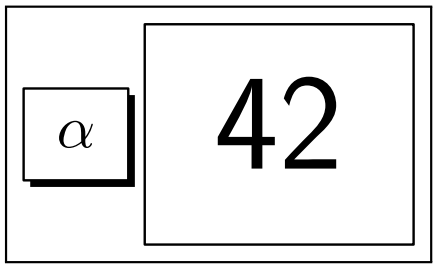
- Peeking inside

- Indistinguishability relation: largest $\sim$ with

$$\frac{s \sim s'}{h(s)=h(s'),\ t(s) \sim t(s')}$$

Black Boxes

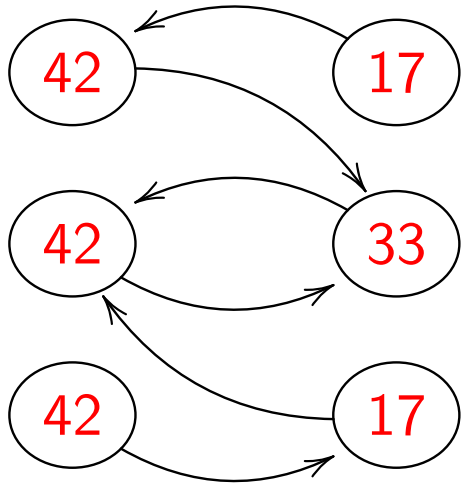
- *Black box* with two buttons and a display...



- Signature:

$$\alpha \quad : \quad S \to D \times S$$

- Peeking inside

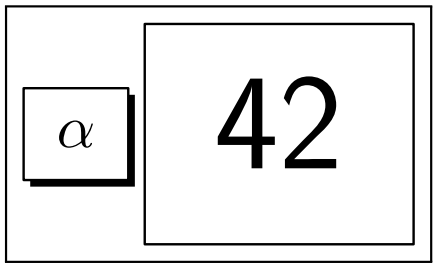


- Indistinguishability relation: largest $\sim$ with

$$\frac{s \sim s'}{h(s)=h(s'),\ t(s) \sim t(s')}$$

Black Boxes

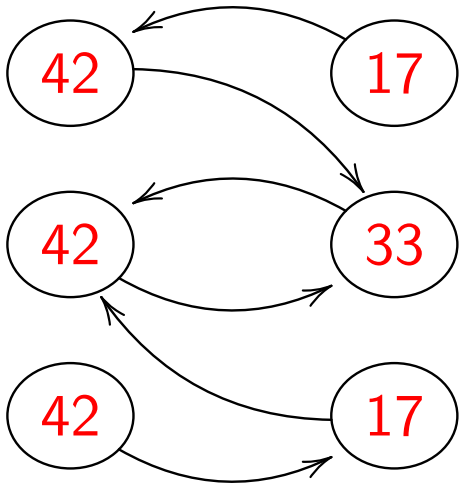
- *Black box* with two buttons and a display...



- Signature:

$$\begin{array}{rcl} h & : & S \to D \\ t & : & S \to S. \end{array}$$

- Peeking inside

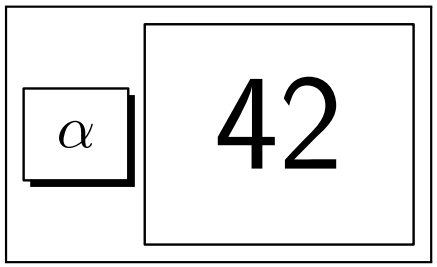


- Indistinguishability relation: largest $\sim$ with

$$\frac{s \sim s'}{h(s)=h(s'),\ t(s) \sim t(s')}$$

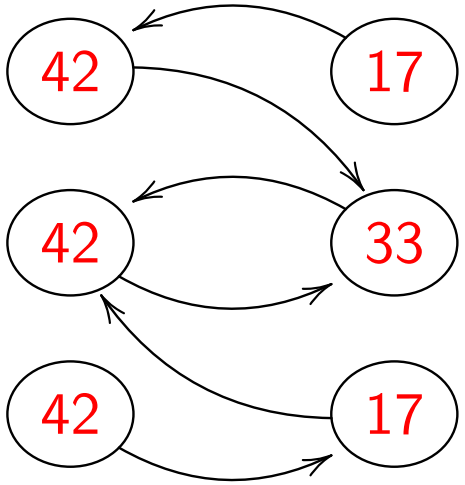
Black Boxes

- *Black box* with two buttons and a display...



- Signature:

- Peeking inside

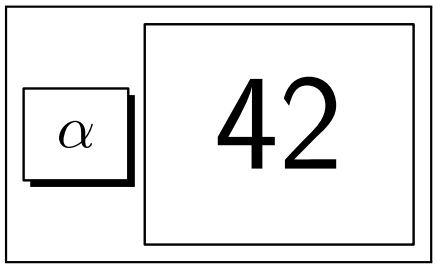


- Indistinguishability relation: largest $\sim$ with

$$\frac{s \sim s'}{h(s)=h(s'),\ t(s) \sim t(s')}$$

Black Boxes

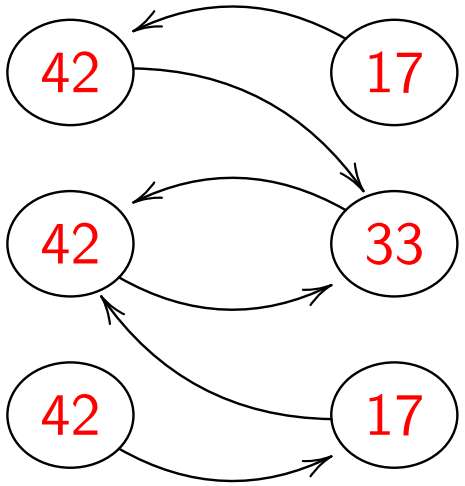
- *Black box* with two buttons and a display...



- Signature:

$$\begin{array}{c} S \\ \alpha \downarrow \\ D \times S \end{array}$$

- Peeking inside

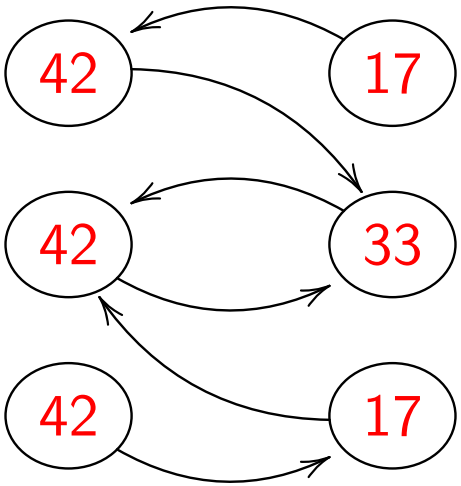


- Indistinguishability relation: largest $\sim$ with

$$\frac{s \sim s'}{h(s)=h(s'),\ t(s) \sim t(s')}$$

Observations are streams

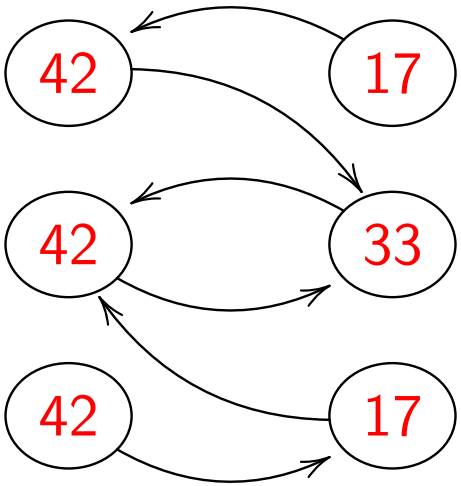
- The inside



- Observations are infinite streams

Observations are streams

- The inside

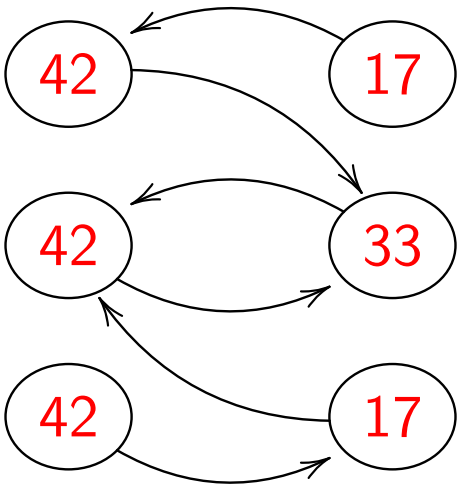


- Observations are infinite streams

$$[42, 33, 42, 33, \ldots]$$

Observations are streams

- The inside

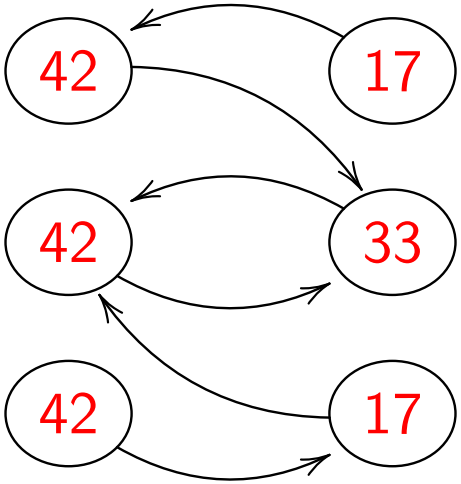


- Observations are infinite streams

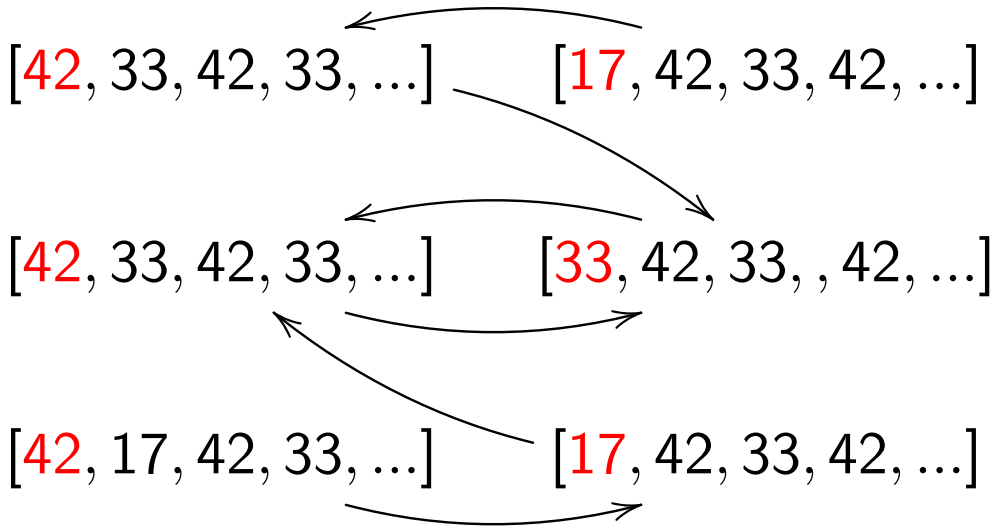
$[42, 33, 42, 33, \ldots]$ $[17, 42, 33, 42, \ldots]$

Observations are streams

- The inside

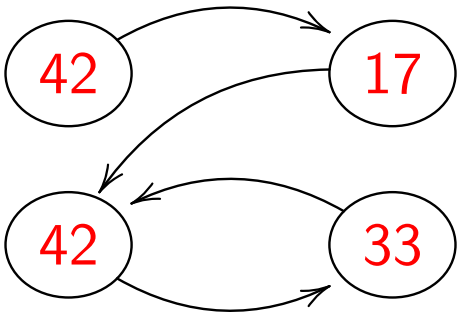


- Observations are infinite streams

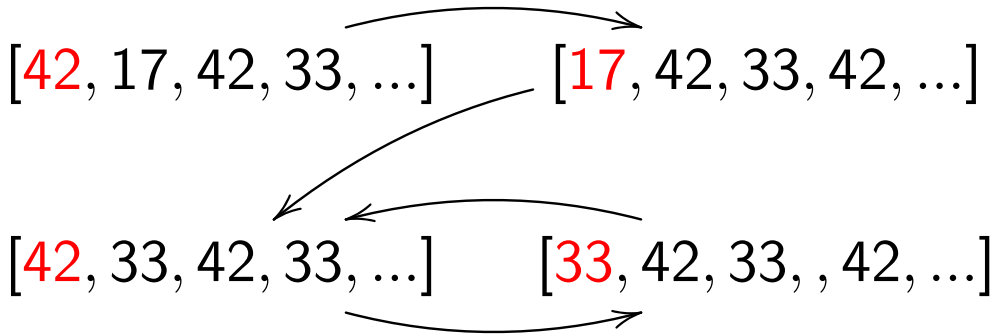


Observations are streams

- The inside



- Observations are infinite streams



Streams are universal

D-streams $= D^\omega = D \times D^\omega$

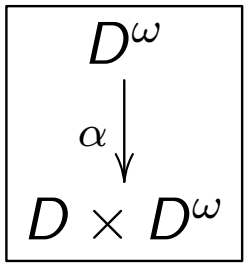
- $hd : D^\omega \to D$
- $tl : D^\omega \to D^\omega$

For each Black Box S

- *Unique* map $\varphi : S \to D^\omega$
  - φ is *homomorphism*

$$hd(\varphi(s)) = h(s)$$
$$tl(\varphi(s)) = \varphi(t(s))$$

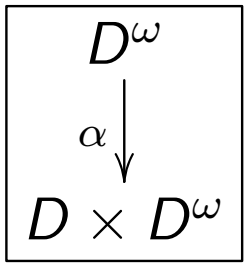
$$\varphi(s) = [h(s), h(t(s)), h(t(t(s))), \ldots]$$



Streams are universal

D-streams $= D^\omega = D \times D^\omega$

- $hd : D^\omega \to D$
- $tl : D^\omega \to D^\omega$



For each Black Box S

- *Unique* map $\varphi : S \to D^\omega$
  - φ is *homomorphism*

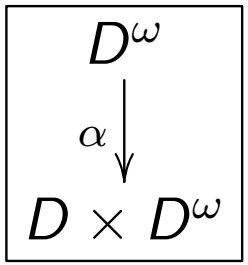
$$hd(\varphi(s)) = h(s)$$
$$tl(\varphi(s)) = \varphi(t(s))$$

$$\varphi(s) = [h(s), h(t(s)), h(t(t(s))), ...]$$

Streams are universal

D-streams $= D^\omega = D \times D^\omega$

- $hd : D^\omega \to D$
- $tl : D^\omega \to D^\omega$



For each Black Box S

- *Unique* map $\varphi : S \to D^\omega$
  - φ is *homomorphism*

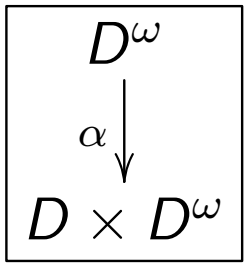
$$hd(\varphi(s)) = h(s)$$
$$tl(\varphi(s)) = \varphi(t(s))$$

$$\varphi(s) = [h(s), h(t(s)), h(t(t(s))), ...]$$

Streams are universal

D-streams $= D^\omega = D \times D^\omega$

- $hd : D^\omega \to D$
- $tl : D^\omega \to D^\omega$



For each Black Box S

- *Unique* map $\varphi : S \to D^\omega$
  - φ is *homomorphism*

$$hd(\varphi(s)) = h(s)$$
$$tl(\varphi(s)) = \varphi(t(s))$$

$$\varphi(s) = [h(s), h(t(s)), h(t(t(s))), ...]$$

Streams are universal

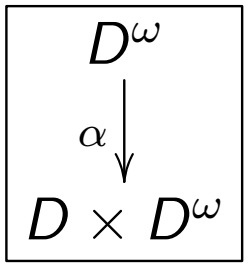
D-streams $= D^\omega = D \times D^\omega$

- $hd : D^\omega \to D$
- $tl : D^\omega \to D^\omega$

For each Black Box S

- *Unique* map $\varphi : S \to D^\omega$
  - φ is *homomorphism*

$$hd(\varphi(s)) = h(s)$$
$$tl(\varphi(s)) = \varphi(t(s))$$



$$\varphi(s) = [h(s), h(t(s)), h(t(t(s))), ...]$$

Streams are universal

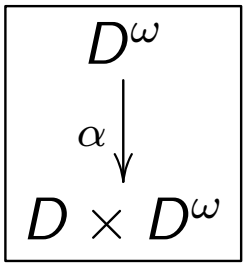
D-streams $= D^\omega = D \times D^\omega$

- $hd : D^\omega \to D$
- $tl : D^\omega \to D^\omega$

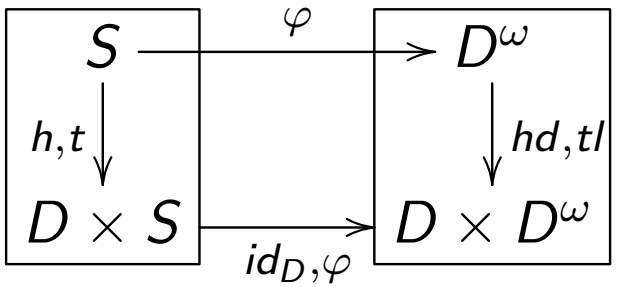
For each Black Box S

- *Unique* map $\varphi : S \to D^\omega$
  - φ is *homomorphism*

$$hd(\varphi(s)) = h(s)$$
$$tl(\varphi(s)) = \varphi(t(s))$$



$$\varphi(s) = [h(s), h(t(s)), h(t(t(s))), ...]$$



Streams are coinductive

- Indistinguishability relation

$$\frac{s \sim s'}{hd(s) = hd(s'),\ tl(s) \sim tl(s')}$$

- Streams satisfy the rule of coinduction

$$\frac{s \sim s'}{s = s'}$$

Streams are coinductive

- Indistinguishability relation

$$\frac{s \sim s'}{hd(s) = hd(s'),\ tl(s) \sim tl(s')}$$

- Streams satisfy the rule of coinduction

$$\frac{s \sim s'}{s = s'}$$

Programming with streams

```
> ones = 1 : ones;
[1,1,1,1,...]
```

```
> from n = n : from(n+1);
> ints = from 0;
[0,1,2,3,...]
```

```
> add u v = h(u)+h(v) : add t(u) t(v);
> add ones ints;
[1,2,3,4,...]
```

- Claim: *add ones ints* $=$ *from* 1

Programming with streams

```
> ones = 1 : ones;
[1,1,1,1,...]

> from n = n : from(n+1);
> ints = from 0;
[0,1,2,3,...]
```

```
> add u v = h(u)+h(v) : add t(u) t(v);
> add ones ints;
[1,2,3,4,...]
```

- Claim: $add\ ones\ ints = from\ 1$

Programming with streams

```
> ones = 1 : ones;
[1,1,1,1,...]

> from n = n : from(n+1);
> ints = from 0;
[0,1,2,3,...]


> add u v = h(u)+h(v) : add t(u) t(v);
> add ones ints;
[1,2,3,4,...]
```

- Claim: $add\ ones\ ints = from\ 1$

Programming with streams

```
> ones = 1 : ones;
[1,1,1,1,...]

> from n = n : from(n+1);
> ints = from 0;
[0,1,2,3,...]


> add u v = h(u)+h(v) : add t(u) t(v);
> add ones ints;
[1,2,3,4,...]
```

- Claim: *add ones ints* $=$ *from* 1

A Coinductive Proof

Generalize: $\mathbf{add\ ones\ (from\ n)} \sim \mathbf{from(n+1)}$

1.
$$\begin{aligned} hd(\mathbf{add\ ones\ (from\ n)}) &= hd(ones) + hd(from\ n) \\ &= 1 + hd(from\ n) \\ &= 1 + n \\ &= hd(\mathbf{from(n+1)}) \end{aligned}$$

2.
$$\begin{aligned} tl(\mathbf{add\ ones\ (from\ n)}) &= add\ tl(ones)\ tl(from\ n) \\ &= add\ ones\ from(n+1) \\ &\sim from(n+2) \\ &= tl(\mathbf{from(n+1)}) \end{aligned}$$

A Coinductive Proof

Generalize: $\mathbf{add\ ones\ (from\ n)} \sim \mathbf{from(n+1)}$

1. $$\begin{aligned} hd(\mathbf{add\ ones\ (from\ n)}) &= hd(ones) + hd(from\ n) \\ &= 1 + hd(from\ n) \\ &= 1 + n \\ &= hd(\mathbf{from(n+1)}) \end{aligned}$$

2. $$\begin{aligned} tl(\mathbf{add\ ones\ (from\ n)}) &= add\ tl(ones)\ tl(from\ n) \\ &= add\ ones\ from(n+1) \\ &\sim from(n+2) \\ &= tl(\mathbf{from(n+1)}) \end{aligned}$$

A Coinductive Proof

Generalize: $\mathbf{add\ ones\ (from\ n)} \sim \mathbf{from(n+1)}$

1.

$$\begin{aligned} hd(\mathbf{add\ ones\ (from\ n)}) &= hd(ones) + hd(from\ n) \\ &= 1 + hd(from\ n) \\ &= 1 + n \\ &= hd(\mathbf{from(n+1)}) \end{aligned}$$

2.

$$\begin{aligned} tl(\mathbf{add\ ones\ (from\ n)}) &= add\ tl(ones)\ tl(from\ n) \\ &= add\ ones\ from(n+1) \\ &\sim from(n+2) \\ &= tl(\mathbf{from(n+1)}) \end{aligned}$$

Local logic

What can be said about a state

- Formulas

$$\phi ::= \mathit{true} \mid < d > \phi, \text{ for each } d \in D$$

- semantics s

$$s \models \mathit{true}$$

$$s \models < d > \phi \quad :\Longleftrightarrow \quad h(s) = d \wedge t(s) \models \phi$$

- logical equivalence

$$s \approx s' :\Longleftrightarrow (s \models \phi \iff s' \models \phi)$$

- completeness:

$$s \sim s' \iff s \approx s'$$

Local logic

What can be said about a state

- Formulas

$$\phi ::= true \mid < d > \phi, \text{ for each } d \in D$$

- semantics s

$$s \models true$$

$$s \models < d > \phi \quad :\Longleftrightarrow \quad h(s) = d \wedge t(s) \models \phi$$

- logical equivalence

$$s \approx s' :\Longleftrightarrow (s \models \phi \iff s' \models \phi)$$

- completeness:

$$s \sim s' \iff s \approx s'$$

Local logic

What can be said about a state

- Formulas

$$\phi ::= true \mid < d > \phi, \text{ for each } d \in D$$

- semantics s

$$s \models true$$
$$s \models < d > \phi \quad :\Longleftrightarrow \quad h(s) = d \wedge t(s) \models \phi$$

- logical equivalence

$$s \approx s' :\Longleftrightarrow (s \models \phi \iff s' \models \phi)$$

- completeness:

$$s \sim s' \iff s \approx s'$$

Local logic

What can be said about a state

- Formulas

$$\phi ::= true \mid < d > \phi, \text{ for each } d \in D$$

- semantics s

$$s \models true$$
$$s \models < d > \phi \quad :\Longleftrightarrow \quad h(s) = d \wedge t(s) \models \phi$$

- logical equivalence

$$s \approx s' :\Longleftrightarrow (s \models \phi \iff s' \models \phi)$$

- completeness:

$$s \sim s' \iff s \approx s'$$

Black Box with error

- Type

$$\boxed{\begin{array}{c} S \\ \downarrow \alpha \\ D \times S + 1 \end{array}}$$

- observational equivalence

$$\frac{s \sim s'}{(\alpha(s)=1 \iff \alpha(s')=1) \vee (hd(s)=hd(s'),\ tl(s) \sim tl(s'))}$$

- Observations

$$D^\omega + D^\star$$

- ... form a black box with error:

$$\alpha(s) = \begin{cases} 1 & s = [\,] \\ (h(s), t(s)) & s \neq [\,] \end{cases}$$

- Logic - correct and complete:

$$\phi ::= true \mid < d > \phi \mid error$$

Black Box with error

- Type

$$\boxed{\begin{array}{c} S \\ \downarrow \alpha \\ D \times S + 1 \end{array}}$$

- observational equivalence

$$\boxed{\frac{s \sim s'}{(\alpha(s)=1 \iff \alpha(s')=1) \vee (hd(s)=hd(s'),\ tl(s) \sim tl(s'))}}$$

- Observations

$$D^{\omega} + D^{\star}$$

- ... form a black box with error:

$$\alpha(s) = \begin{cases} 1 & s = [\,] \\ (h(s), t(s)) & s \neq [\,] \end{cases}$$

- Logic - correct and complete:

$$\phi ::= true \mid < d > \phi \mid error$$

Black Box with error

- Type

$$\boxed{\begin{array}{c} S \\ \downarrow \alpha \\ D \times S + 1 \end{array}}$$

- observational equivalence

$$\boxed{\frac{s \sim s'}{(\alpha(s)=1 \iff \alpha(s')=1) \vee (hd(s)=hd(s'),\ tl(s) \sim tl(s'))}}$$

- Observations

$$D^{\omega} + D^{\star}$$

- ... form a black box with error:

$$\alpha(s) = \begin{cases} 1 & s = [\,] \\ (h(s), t(s)) & s \neq [\,] \end{cases}$$

- Logic - correct and complete:

$$\phi ::= true \mid < d > \phi \mid error$$

Black Box with error

- Type

$$\boxed{\begin{array}{c} S \\ \downarrow \alpha \\ D \times S + 1 \end{array}}$$

- observational equivalence

$$\boxed{\frac{s \sim s'}{(\alpha(s)=1 \iff \alpha(s')=1) \vee (hd(s)=hd(s'),\ tl(s) \sim tl(s'))}}$$

- Observations

$$D^{\omega} + D^{\star}$$

- ... form a black box with error:

$$\alpha(s) = \begin{cases} 1 & s = [\,] \\ (h(s), t(s)) & s \neq [\,] \end{cases}$$

- Logic - correct and complete:

$$\phi ::= true \mid < d > \phi \mid error$$

Black Box with error

- Type

$$\boxed{\begin{array}{c} S \\ \downarrow \alpha \\ D \times S + 1 \end{array}}$$

- observational equivalence

$$\boxed{\frac{s \sim s'}{(\alpha(s)=1 \iff \alpha(s')=1) \vee (hd(s)=hd(s'),\ tl(s) \sim tl(s'))}}$$

- Observations

$$D^{\omega} + D^{\star}$$

- ... form a black box with error:

$$\alpha(s) = \begin{cases} 1 & s = [\,] \\ (h(s), t(s)) & s \neq [\,] \end{cases}$$

- Logic - correct and complete:

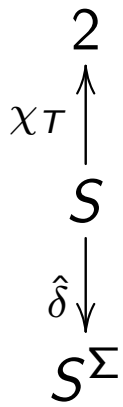
$$\phi ::= true \mid < d > \phi \mid error$$

Automata (Acceptor)

- S set of states, Σ alphabet
  - $T \subseteq S$
    - $\chi_T : S \to 2$
  - $\delta : S \times \Sigma \to S$
    - $\hat{\delta} : S \to S^\Sigma$
- Type
- Homomorphisms $\varphi : \mathcal{A} \to \mathcal{B}$
  - $a \in T_\mathcal{A} \iff \varphi(a) \in T_\mathcal{B}$
  - $\varphi(\delta^\mathcal{A}(a, e)) = \delta^\mathcal{B}(\varphi(a), e)$

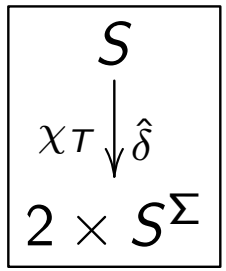
Automata (Acceptor)

- S set of states, Σ alphabet
  - $T \subseteq S$
    - ★ $\chi_T : S \to 2$
  - $\delta : S \times \Sigma \to S$
    - ★ $\hat{\delta} : S \to S^\Sigma$
- Type
- Homomorphisms $\varphi : \mathcal{A} \to \mathcal{B}$
  - $a \in T_\mathcal{A} \iff \varphi(a) \in T_\mathcal{B}$
  - $\varphi(\delta^\mathcal{A}(a, e)) = \delta^\mathcal{B}(\varphi(a), e)$



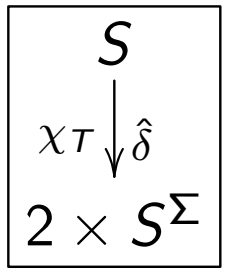
Automata (Acceptor)

- S set of states, Σ alphabet
  - $T \subseteq S$
    - ★ $\chi_T : S \to 2$
  - $\delta : S \times \Sigma \to S$
    - ★ $\hat{\delta} : S \to S^\Sigma$
- Type
- Homomorphisms $\varphi : \mathcal{A} \to \mathcal{B}$
  - $a \in T_\mathcal{A} \iff \varphi(a) \in T_\mathcal{B}$
  - $\varphi(\delta^\mathcal{A}(a, e)) = \delta^\mathcal{B}(\varphi(a), e)$



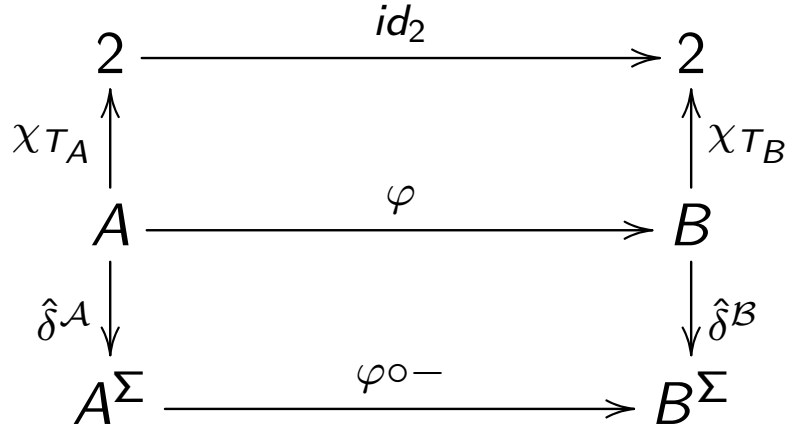
Automata (Acceptor)

- S set of states, Σ alphabet
  - $T \subseteq S$
    - ★ $\chi_T : S \to 2$
  - $\delta : S \times \Sigma \to S$
    - ★ $\hat{\delta} : S \to S^\Sigma$
- Type
- Homomorphisms $\varphi : \mathcal{A} \to \mathcal{B}$
  - $a \in T_\mathcal{A} \iff \varphi(a) \in T_\mathcal{B}$
  - $\varphi(\delta^\mathcal{A}(a, e)) = \delta^\mathcal{B}(\varphi(a), e)$



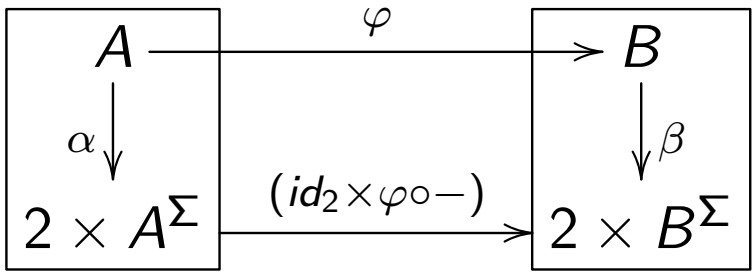
Automata (Acceptor)

- S set of states, Σ alphabet
  - $T \subseteq S$
    - ★ $\chi_T : S \to 2$
  - $\delta : S \times \Sigma \to S$
    - ★ $\hat{\delta} : S \to S^\Sigma$
- Type
- Homomorphisms $\varphi : \mathcal{A} \to \mathcal{B}$
  - $a \in T_\mathcal{A} \iff \varphi(a) \in T_\mathcal{B}$
  - $\varphi(\delta^\mathcal{A}(a, e)) = \delta^\mathcal{B}(\varphi(a), e)$



Automata (Acceptor)

- S set of states, Σ alphabet
  - $T \subseteq S$
    - ★ $\chi_T : S \to 2$
  - $\delta : S \times \Sigma \to S$
    - ★ $\hat{\delta} : S \to S^\Sigma$
- Type
- Homomorphisms $\varphi : \mathcal{A} \to \mathcal{B}$
  - $a \in T_\mathcal{A} \iff \varphi(a) \in T_\mathcal{B}$
  - $\varphi(\delta^\mathcal{A}(a, e)) = \delta^\mathcal{B}(\varphi(a), e)$



Observations are languages

- Observation at s:

$$L(s) := \{w \in \Sigma^\star \mid \delta^\star(s, w) \in T\}$$

- $s \sim s' \iff L(s) = L(s')$
- Indistinguishability relation $\sim$

$$\frac{s \sim s'}{(s \in T \iff s' \in T) \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

Observations are languages

- Observation at s:

$$L(s) := \{w \in \Sigma^\star \mid \delta^\star(s, w) \in T\}$$

- $s \sim s' \iff L(s) = L(s')$
- Indistinguishability relation $\sim$

$$\frac{s \sim s'}{(s \in T \iff s' \in T) \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

Observations are languages

- Observation at s:

$$L(s) := \{w \in \Sigma^\star \mid \delta^\star(s, w) \in T\}$$

- $s \sim s' \iff L(s) = L(s')$
- Indistinguishability relation $\sim$

$$\frac{s \sim s'}{(s \in T \iff s' \in T) \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

Languages are universal

- $\mathbb{P}(\Sigma^\star)$ = Set of all *languages* with alphabeth Σ
  - $\delta : \mathbb{P}(\Sigma^\star) \times \Sigma \to \mathbb{P}(\Sigma^\star)$
    - $\delta(L, e) := L_e = \{u \in \Sigma^\star \mid e \cdot u \in L\}$ *derivative*
    - $T = \{L \in \mathbb{P}(\Sigma^\star) \mid \varepsilon \in L\}$
- For every Σ-automaton $\mathcal{A}$ there is a unique automata homomorphism $\varphi : \mathcal{A} \to \mathbb{P}(\Sigma^\star)$
  - $\varphi(s) := L(s)$
- $s \sim s' \iff \varphi(s) = \varphi(s')$

Languages are universal

- $\mathbb{P}(\Sigma^\star)$ = Set of all *languages* with alphabeth Σ
  - $\delta : \mathbb{P}(\Sigma^\star) \times \Sigma \to \mathbb{P}(\Sigma^\star)$
    - ★ $\delta(L, e) := L_e = \{u \in \Sigma^\star \mid e \cdot u \in L\}$ *derivative*
    - ★ $T = \{L \in \mathbb{P}(\Sigma^\star) \mid \varepsilon \in L\}$
- For every Σ-automaton $\mathcal{A}$ there is a unique automata homomorphism $\varphi : \mathcal{A} \to \mathbb{P}(\Sigma^\star)$
  - $\varphi(s) := L(s)$
- $s \sim s' \iff \varphi(s) = \varphi(s')$

Languages are universal

- $\mathbb{P}(\Sigma^\star)$ = Set of all *languages* with alphabeth Σ
  - $\delta : \mathbb{P}(\Sigma^\star) \times \Sigma \to \mathbb{P}(\Sigma^\star)$
    - ★ $\delta(L, e) := L_e = \{u \in \Sigma^\star \mid e \cdot u \in L\}$ *derivative*
    - ★ $T = \{L \in \mathbb{P}(\Sigma^\star) \mid \varepsilon \in L\}$
- For every Σ-automaton $\mathcal{A}$ there is a unique automata homomorphism $\varphi : \mathcal{A} \to \mathbb{P}(\Sigma^\star)$
  - $\varphi(s) := L(s)$
- $s \sim s' \iff \varphi(s) = \varphi(s')$

Languages are universal

- $\mathbb{P}(\Sigma^\star)$ = Set of all *languages* with alphabeth Σ
  - $\delta : \mathbb{P}(\Sigma^\star) \times \Sigma \to \mathbb{P}(\Sigma^\star)$
    - ★ $\delta(L, e) := L_e = \{u \in \Sigma^\star \mid e \cdot u \in L\}$ *derivative*
    - ★ $T = \{L \in \mathbb{P}(\Sigma^\star) \mid \varepsilon \in L\}$
- For every Σ-automaton $\mathcal{A}$ there is a unique automata homomorphism $\varphi : \mathcal{A} \to \mathbb{P}(\Sigma^\star)$
  - $\varphi(s) := L(s)$
- $s \sim s' \iff \varphi(s) = \varphi(s')$

Languages are universal

- $\mathbb{P}(\Sigma^\star)$ = Set of all *languages* with alphabeth Σ
  - $\delta : \mathbb{P}(\Sigma^\star) \times \Sigma \to \mathbb{P}(\Sigma^\star)$
    - ★ $\delta(L, e) := L_e = \{u \in \Sigma^\star \mid e \cdot u \in L\}$ *derivative*
    - ★ $T = \{L \in \mathbb{P}(\Sigma^\star) \mid \varepsilon \in L\}$
- For every Σ-automaton $\mathcal{A}$ there is a unique automata homomorphism $\varphi : \mathcal{A} \to \mathbb{P}(\Sigma^\star)$
  - $\varphi(s) := L(s)$
- $s \sim s' \iff \varphi(s) = \varphi(s')$

Languages are coinductive

- Indistinguishability relation

$$\frac{L \sim L'}{(\varepsilon \in L \iff \varepsilon \in L') \wedge \forall e \in \Sigma . L_e \sim L'_e}$$

- Languages satisfy

Rule of coinduction

$$\frac{L \sim L'}{L = L'}$$

- Can be used to prove language equalities, e.g.

$$L^* \cdot M = L \cdot (L^* \cdot M) + M$$

Languages are coinductive

- Indistinguishability relation

$$\frac{L \sim L'}{(\varepsilon \in L \iff \varepsilon \in L') \wedge \forall e \in \Sigma . L_e \sim L'_e}$$

- Languages satisfy

Rule of coinduction

$$\frac{L \sim L'}{L = L'}$$

- Can be used to prove language equalities, e.g.

$$L^* \cdot M = L \cdot (L^* \cdot M) + M$$

Languages are coinductive

- Indistinguishability relation

$$\frac{L \sim L'}{(\varepsilon \in L \iff \varepsilon \in L') \wedge \forall e \in \Sigma . L_e \sim L'_e}$$

- Languages satisfy

Rule of coinduction

$$\frac{L \sim L'}{L = L'}$$

- Can be used to prove language equalities, e.g.

$$L^\star \cdot M = L \cdot (L^\star \cdot M) + M$$

Object oriented programs

- State encapsulated in objects
  - Mutators, Observers

```
class Account{
   private int balance = 0;
   public void trans(int z){balance+=z;}
   public int show(){ return balance; }
}
```

- Do'nt care whether
  - $acc.trans(z_1).trans(z_2) = acc.trans(z_1 + z_2)$
  - not observable
- Insist that
  - $acc.trans(z_1).trans(z_2).show() = acc.trans(z_1 + z_2).show()$

Object oriented programs

- State encapsulated in objects
  - Mutators, Observers

```
class Account{
   private int balance = 0;
   public void trans(int z){balance+=z;}
   public int show(){ return balance; }
}
```

- Do'nt care whether
  - $acc.trans(z_1).trans(z_2) = acc.trans(z_1 + z_2)$
  - not observable
- Insist that
  - $acc.trans(z_1).trans(z_2).show() = acc.trans(z_1 + z_2).show()$

Object oriented programs

- State encapsulated in objects
  - Mutators, Observers

```
class Account{
   private int balance = 0;
   public void trans(int z){balance+=z;}
   public int show(){ return balance; }
}
```

- Do'nt care whether
  - $acc.trans(z_1).trans(z_2) = acc.trans(z_1 + z_2)$
  - not observable
- Insist that
  - $acc.trans(z_1).trans(z_2).show() = acc.trans(z_1 + z_2).show()$

Object oriented programs

- State encapsulated in objects
  - Mutators, Observers

```
class Account{
   private int balance = 0;
   public void trans(int z){balance+=z;}
   public int show(){ return balance; }
}
```

- Do'nt care whether
  - $acc.trans(z_1).trans(z_2) = acc.trans(z_1 + z_2)$
  - not observable
- Insist that
  - $acc.trans(z_1).trans(z_2).show() = acc.trans(z_1 + z_2).show()$

Object oriented programs

- State encapsulated in objects
  - Mutators, Observers

```
class Account{
   private int balance = 0;
   public void trans(int z){balance+=z;}
   public int show(){ return balance; }
}
```

- Do'nt care whether
  - $acc.trans(z_1).trans(z_2) = acc.trans(z_1 + z_2)$
  - not observable
- Insist that
  - $acc.trans(z_1).trans(z_2).show() = acc.trans(z_1 + z_2).show()$

Moore-Automata

- S set of states, Σ alphabet (input), D set of outputs
  - $\delta : S \times \Sigma \to S$,
  - $\gamma : S \to D$
- in state s word $w = e_1 \cdot e_2 \cdot \ldots \cdot e_n$ gives output $\gamma(\delta^\star(s, w))$
  - observations at s: $D - \mathit{labelled}, \Sigma - \mathit{branching}$ tree:

Moore-Automata

- S set of states, Σ alphabet (input), D set of outputs
  - $\delta : S \times \Sigma \to S$,
  - $\gamma : S \to D$
- in state s word $w = e_1 \cdot e_2 \cdot \ldots \cdot e_n$ gives output $\gamma(\delta^\star(s, w))$
  - observations at s: $D - \mathit{labelled}, \Sigma - \mathit{branching}$ tree:

Moore-Automata

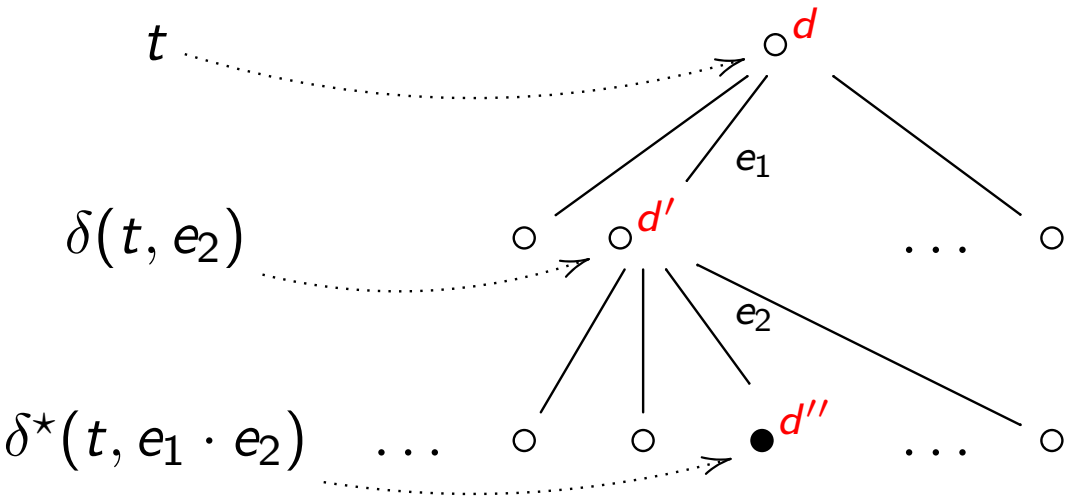
- S set of states, Σ alphabet (input), D set of outputs
  - $\delta : S \times \Sigma \to S$,
  - $\gamma : S \to D$
- in state s word $w = e_1 \cdot e_2 \cdot \ldots \cdot e_n$ gives output $\gamma(\delta^\star(s, w))$
  - observations at s: $D - \mathit{labelled}, \Sigma - \mathit{branching}$ tree:

Moore-Automata

- S set of states, Σ alphabet (input), D set of outputs
  - $\delta : S \times \Sigma \to S$,
  - $\gamma : S \to D$
- in state s word $w = e_1 \cdot e_2 \cdot \ldots \cdot e_n$ gives output $\gamma(\delta^\star(s, w))$
  - observations at s: $D - \mathit{labelled}, \Sigma - \mathit{branching}$ tree:

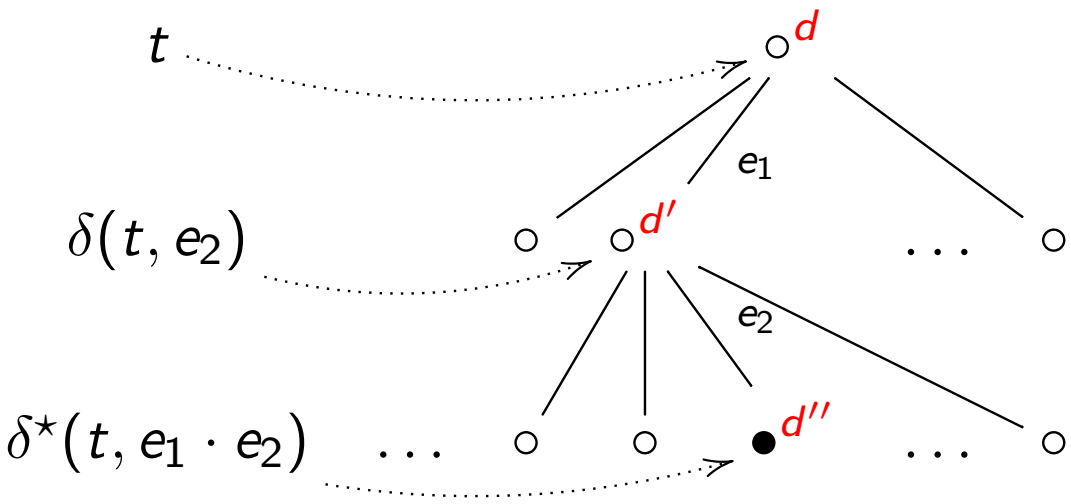
Moore-Automata

- S set of states, Σ alphabet (input), D set of outputs
  - $\delta : S \times \Sigma \to S$,
  - $\gamma : S \to D$
- in state s word $w = e_1 \cdot e_2 \cdot \ldots \cdot e_n$ gives output $\gamma(\delta^\star(s, w))$
  - *observations* at s: $D - labelled, \Sigma - branching$ tree:



D-labelled Σ-branching trees

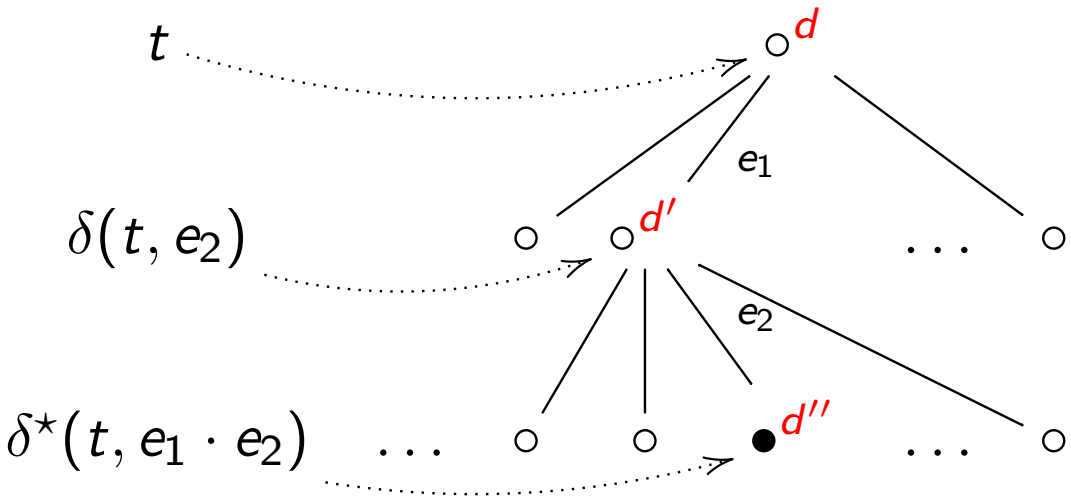
- Automaton of all $a\Sigma - branching\ D - labeled$ trees: $D^{\Sigma^\star}$



- Each node uniquely determined by its "*address*" $w \in \Sigma^\star$
- Tree represents a map $t : \Sigma^\star \to D$
- $D^{\Sigma^\star}$ = set of all such trees has automaton structure
  - $\gamma(t) = t(\varepsilon)$
  - $\delta(t, e)(w) = t(e \cdot w)$ for each $w \in \Sigma^\star$

D-labelled Σ-branching trees

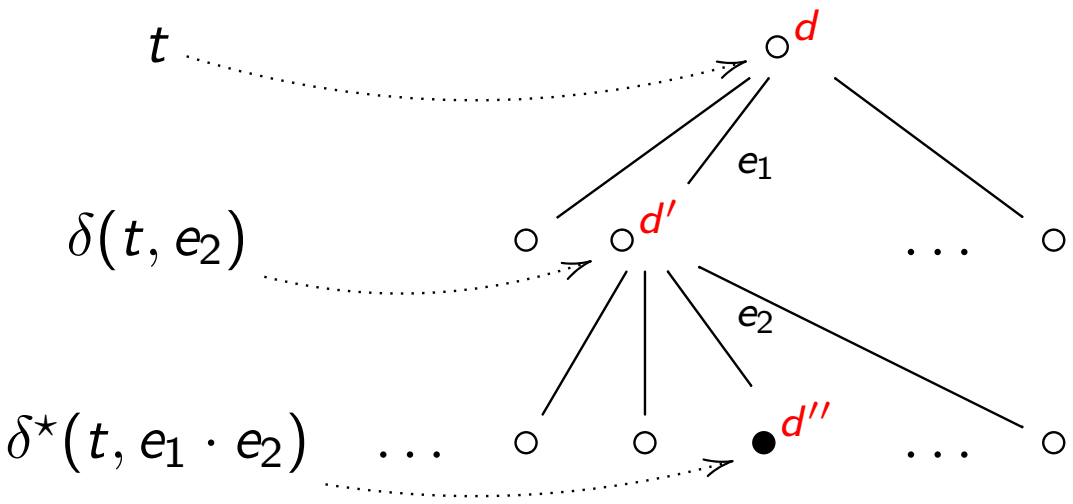
- Automaton of all $a\Sigma - \textit{branching}\ D - \textit{labeled}$ trees: $D^{\Sigma^\star}$



- Each node uniquely determined by its "*address*" $w \in \Sigma^\star$
- Tree represents a map $t : \Sigma^\star \to D$
- $D^{\Sigma^\star}$ = set of all such trees has automaton structure
  - $\gamma(t) = t(\varepsilon)$
  - $\delta(t, e)(w) = t(e \cdot w)$ for each $w \in \Sigma^\star$

D-labelled Σ-branching trees

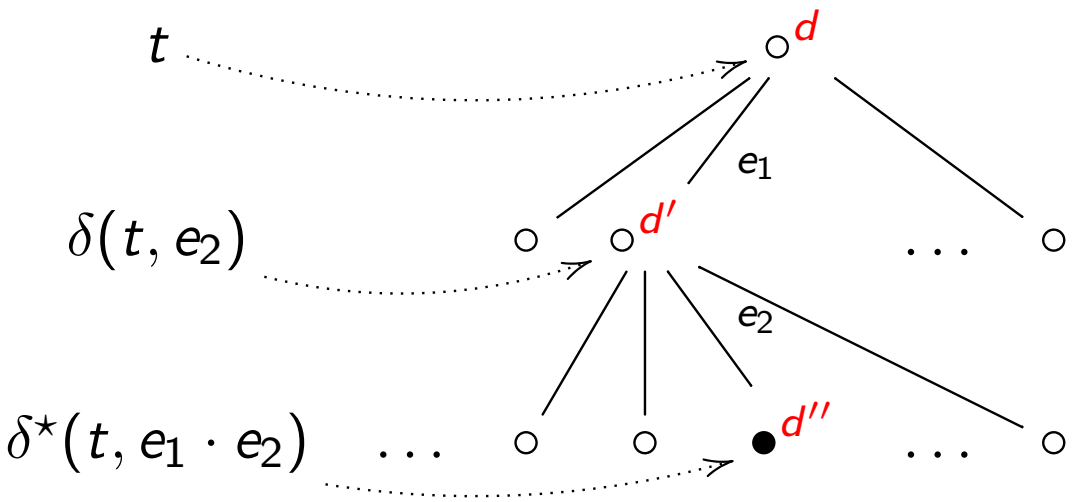
- Automaton of all $a\Sigma - branching\ D - labeled$ trees: $D^{\Sigma^\star}$



- Each node uniquely determined by its "*address*" $w \in \Sigma^\star$
- Tree represents a map $t : \Sigma^\star \to D$
- $D^{\Sigma^\star}$ = set of all such trees has automaton structure
  - $\gamma(t) = t(\varepsilon)$
  - $\delta(t, e)(w) = t(e \cdot w)$ for each $w \in \Sigma^\star$

D-labelled Σ-branching trees

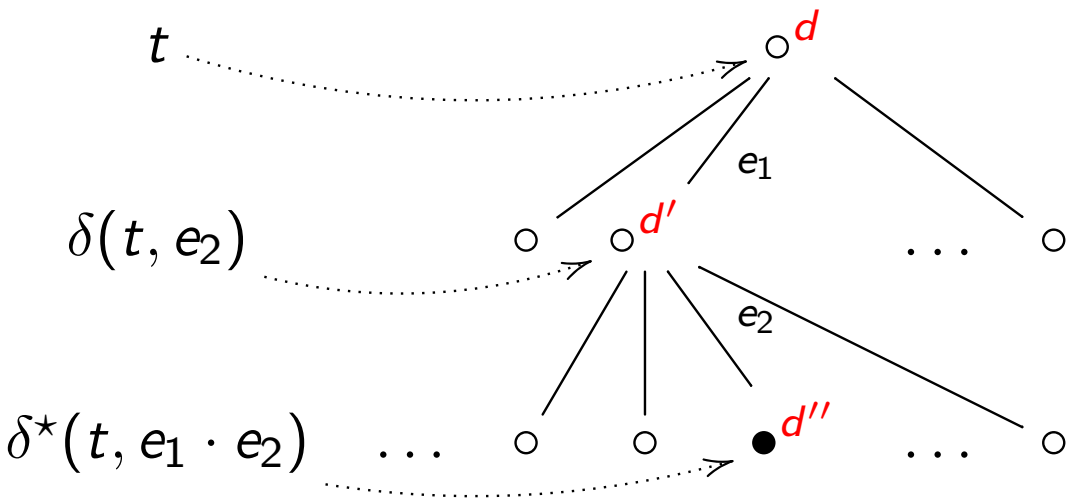
- Automaton of all $a\Sigma - branching\ D - labeled$ trees: $D^{\Sigma^\star}$



- Each node uniquely determined by its "*address*" $w \in \Sigma^\star$
- Tree represents a map $t : \Sigma^\star \to D$
- $D^{\Sigma^\star}$ = set of all such trees has automaton structure
  - $\gamma(t) = t(\varepsilon)$
  - $\delta(t, e)(w) = t(e \cdot w)$ for each $w \in \Sigma^\star$

D-labelled Σ-branching trees

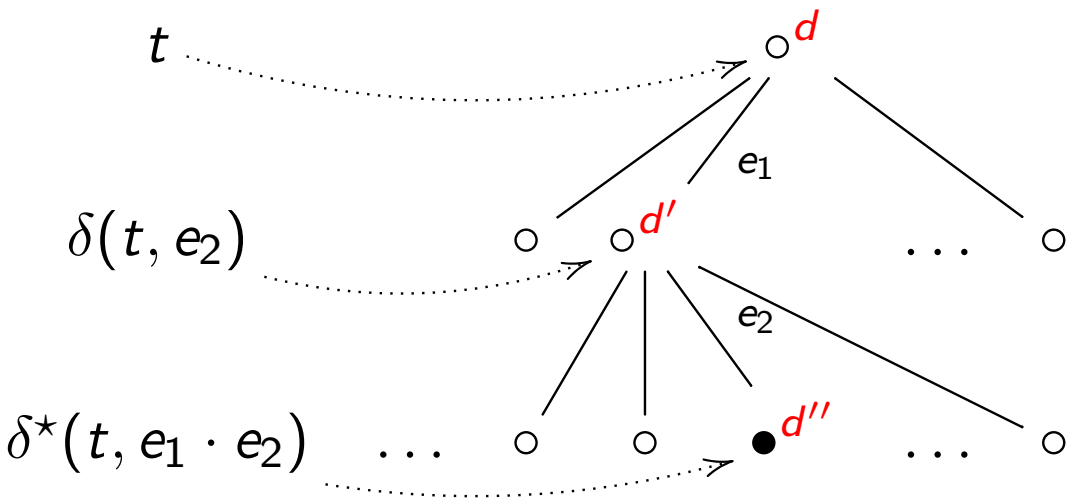
- Automaton of all $a\Sigma - branching\ D - labeled$ trees: $D^{\Sigma^\star}$



- Each node uniquely determined by its "*address*" $w \in \Sigma^\star$
- Tree represents a map $t : \Sigma^\star \to D$
- $D^{\Sigma^\star}$ = set of all such trees has automaton structure
  - $\gamma(t) = t(\varepsilon)$
  - $\delta(t, e)(w) = t(e \cdot w)$ for each $w \in \Sigma^\star$

D-labelled Σ-branching trees

- Automaton of all $a\Sigma - branching\ D - labeled$ trees: $D^{\Sigma^\star}$



- Each node uniquely determined by its "*address*" $w \in \Sigma^\star$
- Tree represents a map $t : \Sigma^\star \to D$
- $D^{\Sigma^\star}$ = set of all such trees has automaton structure
  - $\gamma(t) = t(\varepsilon)$
  - $\delta(t, e)(w) = t(e \cdot w)$ for each $w \in \Sigma^\star$

Moore-Automata

- Morphism between automata preserve γ and δ
- $\varphi : A \to B$ with
  - $\gamma^B(\varphi(a)) = \gamma^A(a)$
  - $\delta^B(\varphi(a), e) = \varphi(\delta^A(a, e))$
- Unique for automaton of trees $B = D^{\Sigma^\star}$:
  - $\varphi(a)(\varepsilon) = \gamma(\tau(a)) = \gamma(a)$
  - $\varphi(a)(e \cdot w) = \delta(\varphi(a), e)(w) = \varphi(\delta(a, e))(w)$
- Indistinguishability: largest relation $\sim$ with

$$\frac{s \sim s'}{\gamma(s)=\gamma(s') \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

- $s \sim s' :\iff \varphi(s) = \varphi(s')$

Moore-Automata

- Morphism between automata preserve γ and δ
- $\varphi : A \to B$ with
  - $\gamma^B(\varphi(a)) = \gamma^A(a)$
  - $\delta^B(\varphi(a), e) = \varphi(\delta^A(a, e))$
- Unique for automaton of trees $B = D^{\Sigma^\star}$:
  - $\varphi(a)(\varepsilon) = \gamma(\tau(a)) = \gamma(a)$
  - $\varphi(a)(e \cdot w) = \delta(\varphi(a), e)(w) = \varphi(\delta(a, e))(w)$
- Indistinguishability: largest relation $\sim$ with

$$\frac{s \sim s'}{\gamma(s)=\gamma(s') \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

- $s \sim s' :\iff \varphi(s) = \varphi(s')$

Moore-Automata

- Morphism between automata preserve γ and δ
- $\varphi : A \to B$ with
  - $\gamma^B(\varphi(a)) = \gamma^A(a)$
  - $\delta^B(\varphi(a), e) = \varphi(\delta^A(a, e))$
- Unique for automaton of trees $B = D^{\Sigma^\star}$:
  - $\varphi(a)(\varepsilon) = \gamma(\tau(a)) = \gamma(a)$
  - $\varphi(a)(e \cdot w) = \delta(\varphi(a), e)(w) = \varphi(\delta(a, e))(w)$
- Indistinguishability: largest relation $\sim$ with

$$\frac{s \sim s'}{\gamma(s)=\gamma(s') \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

- $s \sim s' :\iff \varphi(s) = \varphi(s')$

Moore-Automata

- Morphism between automata preserve γ and δ
- $\varphi : A \to B$ with
  - $\gamma^B(\varphi(a)) = \gamma^A(a)$
  - $\delta^B(\varphi(a), e) = \varphi(\delta^A(a, e))$
- Unique for automaton of trees $B = D^{\Sigma^\star}$:
  - $\varphi(a)(\varepsilon) = \gamma(\tau(a)) = \gamma(a)$
  - $\varphi(a)(e \cdot w) = \delta(\varphi(a), e)(w) = \varphi(\delta(a, e))(w)$
- Indistinguishability: largest relation $\sim$ with

$$\frac{s \sim s'}{\gamma(s)=\gamma(s') \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

- $s \sim s' :\iff \varphi(s) = \varphi(s')$

Moore-Automata

- Morphism between automata preserve γ and δ
- $\varphi : A \to B$ with
  - $\gamma^B(\varphi(a)) = \gamma^A(a)$
  - $\delta^B(\varphi(a), e) = \varphi(\delta^A(a, e))$
- Unique for automaton of trees $B = D^{\Sigma^\star}$:
  - $\varphi(a)(\varepsilon) = \gamma(\tau(a)) = \gamma(a)$
  - $\varphi(a)(e \cdot w) = \delta(\varphi(a), e)(w) = \varphi(\delta(a, e))(w)$
- Indistinguishability: largest relation $\sim$ with

$$\frac{s \sim s'}{\gamma(s)=\gamma(s') \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

- $s \sim s' :\iff \varphi(s) = \varphi(s')$

Moore-Automata

- Morphism between automata preserve γ and δ
- $\varphi : A \to B$ with
  - $\gamma^B(\varphi(a)) = \gamma^A(a)$
  - $\delta^B(\varphi(a), e) = \varphi(\delta^A(a, e))$
- Unique for automaton of trees $B = D^{\Sigma^\star}$:
  - $\varphi(a)(\varepsilon) = \gamma(\tau(a)) = \gamma(a)$
  - $\varphi(a)(e \cdot w) = \delta(\varphi(a), e)(w) = \varphi(\delta(a, e))(w)$
- Indistinguishability: largest relation $\sim$ with

$$\frac{s \sim s'}{\gamma(s)=\gamma(s') \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

- $s \sim s' :\iff \varphi(s) = \varphi(s')$

Moore-Automata

- Morphism between automata preserve γ and δ
- $\varphi : A \to B$ with
  - $\gamma^B(\varphi(a)) = \gamma^A(a)$
  - $\delta^B(\varphi(a), e) = \varphi(\delta^A(a, e))$
- Unique for automaton of trees $B = D^{\Sigma^\star}$:
  - $\varphi(a)(\varepsilon) = \gamma(\tau(a)) = \gamma(a)$
  - $\varphi(a)(e \cdot w) = \delta(\varphi(a), e)(w) = \varphi(\delta(a, e))(w)$
- Indistinguishability: largest relation $\sim$ with

$$\frac{s \sim s'}{\gamma(s)=\gamma(s') \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

- $s \sim s' :\iff \varphi(s) = \varphi(s')$

Moore-Automata

- Morphism between automata preserve γ and δ
- $\varphi : A \to B$ with
  - $\gamma^B(\varphi(a)) = \gamma^A(a)$
  - $\delta^B(\varphi(a), e) = \varphi(\delta^A(a, e))$
- Unique for automaton of trees $B = D^{\Sigma^\star}$:
  - $\varphi(a)(\varepsilon) = \gamma(\tau(a)) = \gamma(a)$
  - $\varphi(a)(e \cdot w) = \delta(\varphi(a), e)(w) = \varphi(\delta(a, e))(w)$
- Indistinguishability: largest relation $\sim$ with

$$\frac{s \sim s'}{\gamma(s)=\gamma(s') \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

- $s \sim s' :\iff \varphi(s) = \varphi(s')$

Moore-Automata

- Morphism between automata preserve γ and δ
- $\varphi : A \to B$ with
  - $\gamma^B(\varphi(a)) = \gamma^A(a)$
  - $\delta^B(\varphi(a), e) = \varphi(\delta^A(a, e))$
- Unique for automaton of trees $B = D^{\Sigma^\star}$:
  - $\varphi(a)(\varepsilon) = \gamma(\tau(a)) = \gamma(a)$
  - $\varphi(a)(e \cdot w) = \delta(\varphi(a), e)(w) = \varphi(\delta(a, e))(w)$
- Indistinguishability: largest relation $\sim$ with

$$\frac{s \sim s'}{\gamma(s)=\gamma(s') \ \wedge \ \forall a \in \Sigma.\ \delta(s,a) \sim \delta(s',a)}$$

- $s \sim s' :\iff \varphi(s) = \varphi(s')$

Nondeterminism

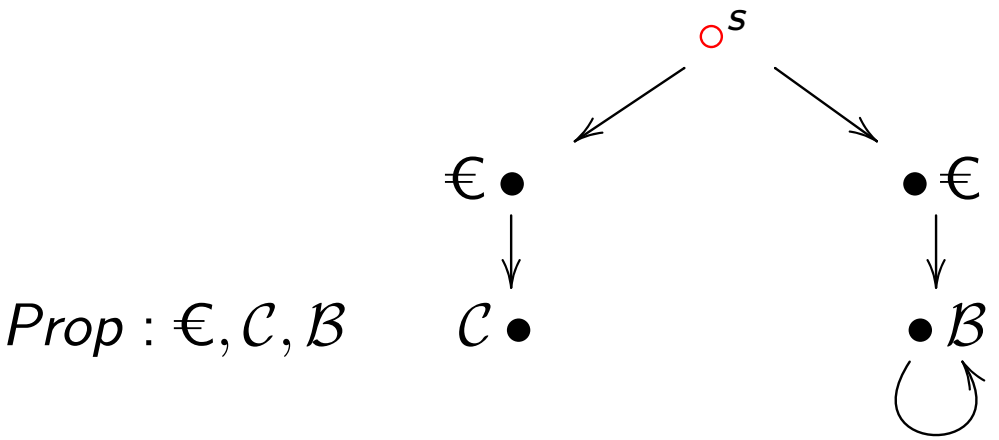
- Transition relation $R \subseteq S \times S$ with state properties,
- labeling $l : S \to \mathbb{P}(Prop)$

Nondeterminism

- Transition relation $R \subseteq S \times S$ with state properties,
- labeling $l : S \to \mathbb{P}(Prop)$

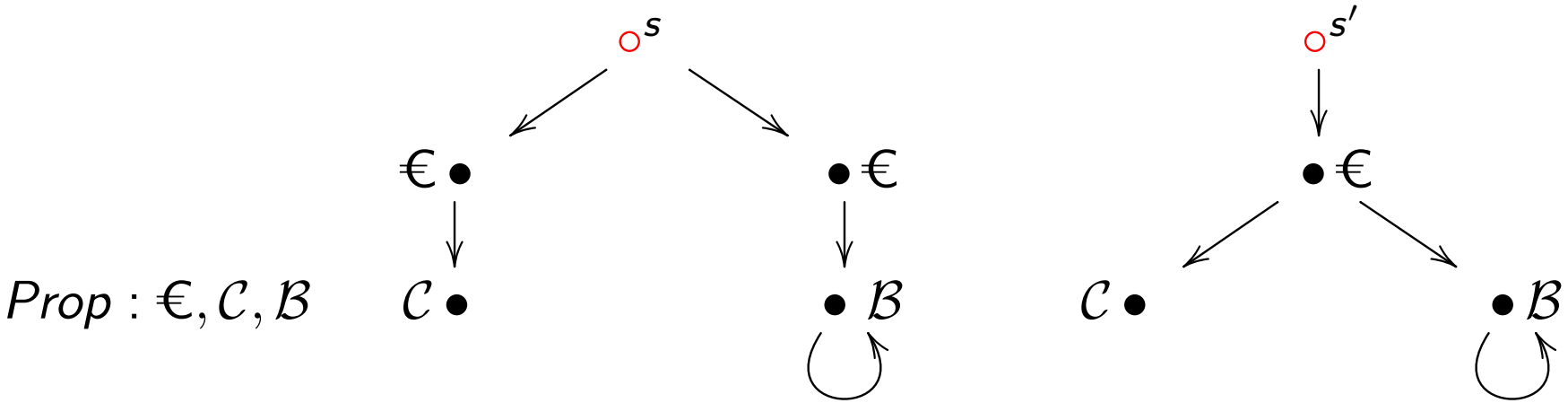
Nondeterminism

- Transition relation $R \subseteq S \times S$ with state properties,
- labeling $l : S \to \mathbb{P}(Prop)$



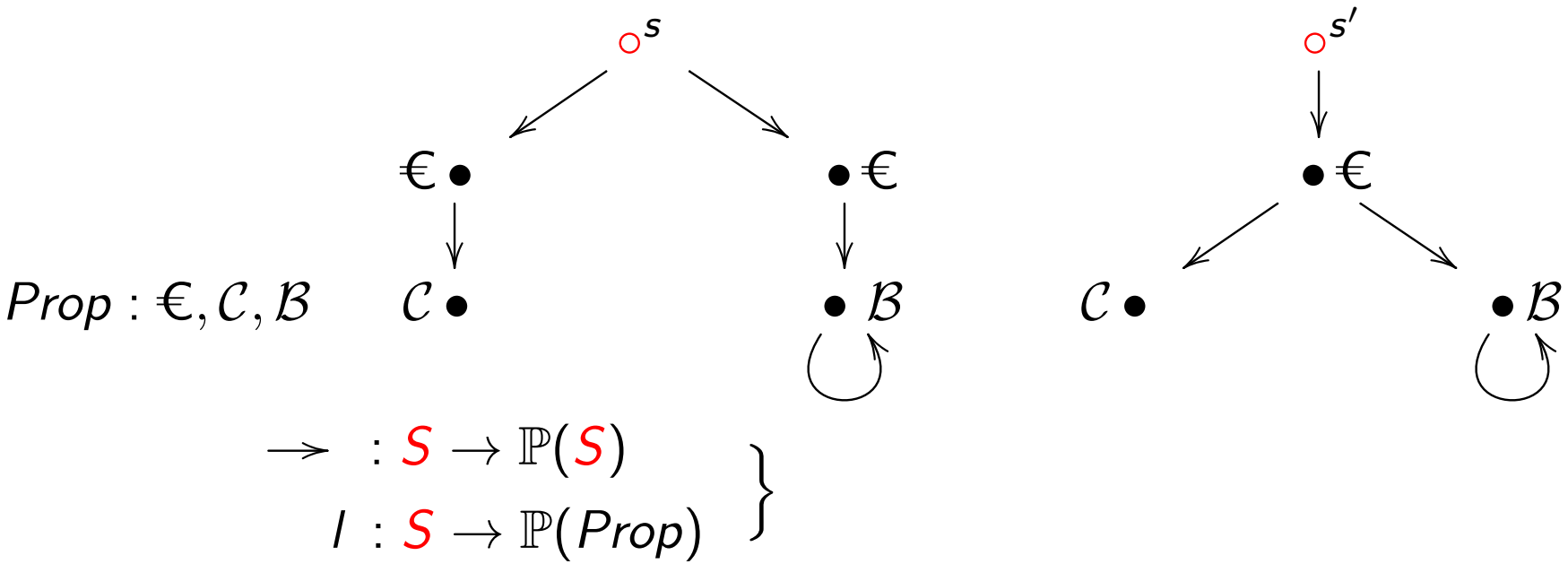
Nondeterminism

- Transition relation $R \subseteq S \times S$ with state properties,
- labeling $l : S \to \mathbb{P}(Prop)$



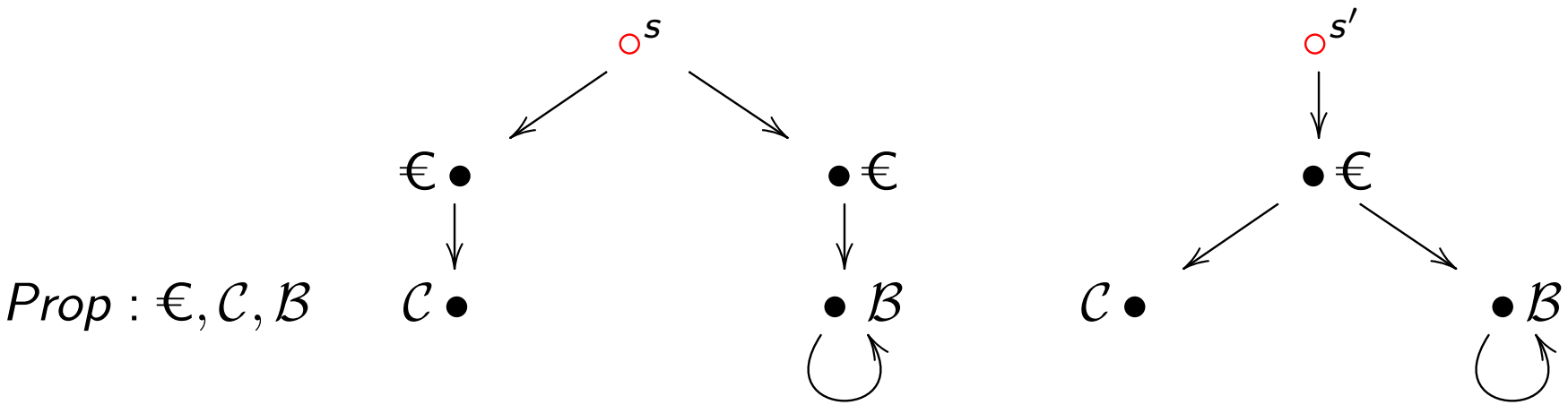
Nondeterminism

- Transition relation $R \subseteq S \times S$ with state properties,
- labeling $l : S \to \mathbb{P}(Prop)$



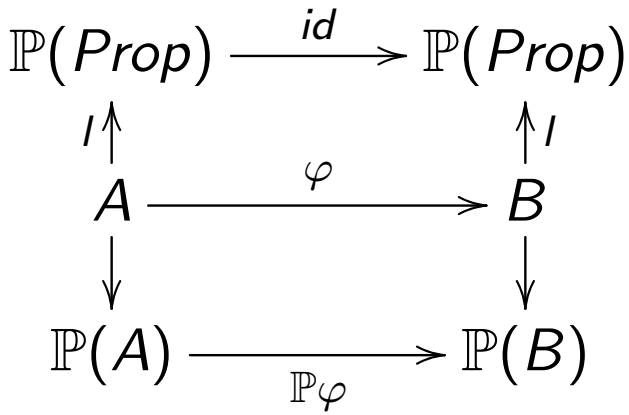
Nondeterminism

- Transition relation $R \subseteq S \times S$ with state properties,
- labeling $l : S \to \mathbb{P}(Prop)$



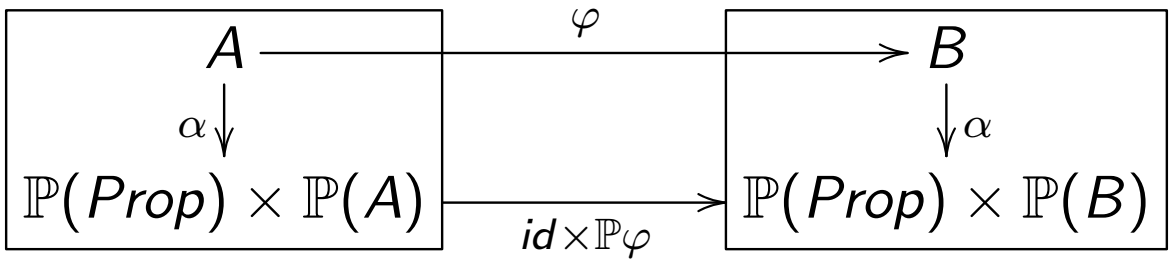
Homomorphisms

- Homomorphism



Homomorphisms

- Homomorphism



Observations

- *Set − branching*, $\mathbb{P}(AP)$-labelled trees
  - too many to form one Kripke structure
  - may restrict to finitely branching
- Observational equivalence still definable
  - largest relation $\sim$ *with*
  - $s \sim s' \implies I(s) = I(s')$ and

Observations

- *Set − branching*, $\mathbb{P}(AP)$-labelled trees
  - too many to form one Kripke structure
  - may restrict to finitely branching
- Observational equivalence still definable
  - largest relation $\sim$ *with*
  - $s \sim s' \implies l(s) = l(s')$ and

Observations

- *Set – branching*, $\mathbb{P}(AP)$-labelled trees
  - too many to form one Kripke structure
  - may restrict to finitely branching
- Observational equivalence still definable
  - largest relation $\sim$ *with*
  - $s \sim s' \implies I(s) = I(s')$ and

Observations

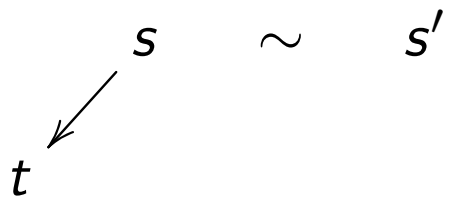
- *Set − branching*, $\mathbb{P}(AP)$-labelled trees
  - too many to form one Kripke structure
  - may restrict to finitely branching
- Observational equivalence still definable
  - largest relation $\sim$ *with*
  - $s \sim s' \implies l(s) = l(s')$ and

Observations

- *Set − branching*, $\mathbb{P}(AP)$-labelled trees
  - too many to form one Kripke structure
  - may restrict to finitely branching
- Observational equivalence still definable
  - largest relation $\sim$ *with*
  - $s \sim s' \implies l(s) = l(s')$ and

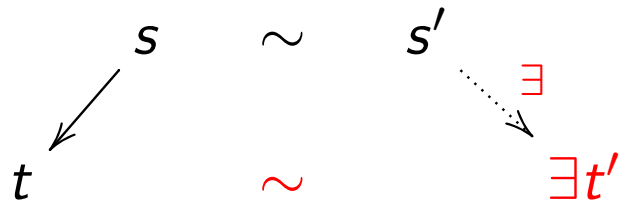
Observations

- *Set − branching*, $\mathbb{P}(AP)$-labelled trees
  - too many to form one Kripke structure
  - may restrict to finitely branching
- Observational equivalence still definable
  - largest relation $\sim$ *with*
  - $s \sim s' \implies l(s) = l(s')$ and



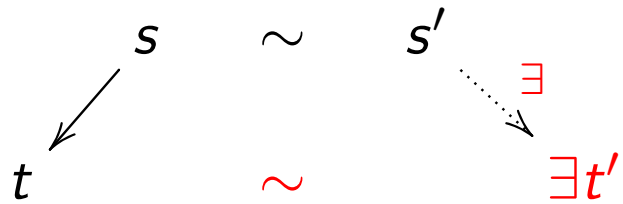
Observations

- *Set − branching*, $\mathbb{P}(AP)$-labelled trees
  - too many to form one Kripke structure
  - may restrict to finitely branching
- Observational equivalence still definable
  - largest relation $\sim$ *with*
  - $s \sim s' \implies l(s) = l(s')$ and



Observations

- *Set − branching*, $\mathbb{P}(AP)$-labelled trees
  - too many to form one Kripke structure
  - may restrict to finitely branching
- Observational equivalence still definable
  - largest relation $\sim$ *with*
  - $s \sim s' \implies l(s) = l(s')$ and



...and symmetrically

$$\frac{s \sim s'}{s' \sim s \wedge l(s) = l(s') \wedge \forall t.\, s \to t \implies \exists t'.\, s' \to t' \wedge t \sim t'}$$

Modal Kripke Logic

- Formulae

$$\begin{array}{rcl} \phi & :: & p \in AP \\ & | & \mathit{true} \mid \neg\phi \mid \phi_1 \vee \phi_2 \\ & | & \Box\phi \mid \Diamond\phi \end{array}$$

- Semantics
  - $s \models p :\Longleftrightarrow p \in I(s)$
  - $s \models \Box\phi :\Longleftrightarrow \forall t.(s \to t) \implies t \models \phi$
  - $\Diamond\phi :\Longleftrightarrow \neg\Box\neg\phi$

- Logical equivalence

$$s \approx t \;:\Longleftrightarrow\; (s \models \phi \iff t \models \phi) \text{ for all formulae } \phi$$

Modal Kripke Logic

- Formulae

$$\begin{array}{rcl} \phi & :: & p \in AP \\ & | & true \mid \neg\phi \mid \phi_1 \vee \phi_2 \\ & | & \Box\phi \mid \Diamond\phi \end{array}$$

- Semantics
  - $s \models p :\Longleftrightarrow p \in l(s)$
  - $s \models \Box\phi :\Longleftrightarrow \forall t.(s \to t) \implies t \models \phi$
  - $\Diamond\phi :\Longleftrightarrow \neg\Box\neg\phi$
- Logical equivalence

$$s \approx t \;:\Longleftrightarrow\; (s \models \phi \iff t \models \phi) \text{ for all formulae } \phi$$

Modal Kripke Logic

- Formulae

$$\begin{array}{rcl} \phi & :: & p \in AP \\ & | & true \mid \neg\phi \mid \phi_1 \vee \phi_2 \\ & | & \Box\phi \mid \Diamond\phi \end{array}$$

- Semantics
  - $s \models p :\Longleftrightarrow\ p \in l(s)$
  - $s \models \Box\phi :\Longleftrightarrow\ \forall t.(s \twoheadrightarrow t) \implies t \models \phi$
  - $\Diamond\phi :\Longleftrightarrow\ \neg\Box\neg\phi$
- Logical equivalence

$$s \approx t\ :\Longleftrightarrow\ (s \models \phi \iff t \models \phi) \text{ for all formulae } \phi$$

Modal Kripke Logic

- Formulae

$$\begin{array}{rcl} \phi & :: & p \in AP \\ & | & true \mid \neg\phi \mid \phi_1 \vee \phi_2 \\ & | & \Box\phi \mid \Diamond\phi \end{array}$$

- Semantics
  - $s \models p :\Longleftrightarrow\ p \in l(s)$
  - $s \models \Box\phi :\Longleftrightarrow\ \forall t.(s \twoheadrightarrow t) \implies t \models \phi$
  - $\Diamond\phi :\Longleftrightarrow\ \neg\Box\neg\phi$
- Logical equivalence

$$s \approx t \ :\Longleftrightarrow\ (s \models \phi \iff t \models \phi) \text{ for all formulae } \phi$$

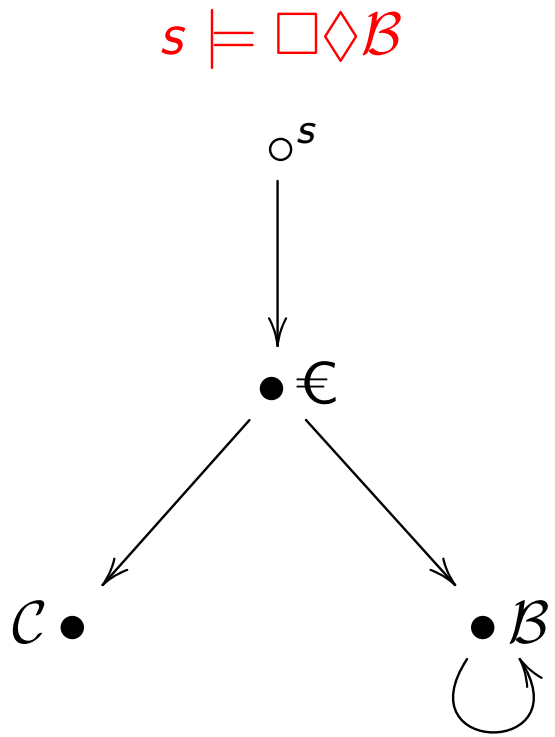
Modal Kripke Logic

- Formulae

$$\begin{array}{rcl} \phi & :: & p \in AP \\ & | & true \mid \neg\phi \mid \phi_1 \vee \phi_2 \\ & | & \Box\phi \mid \Diamond\phi \end{array}$$

- Semantics
  - $s \models p :\iff p \in I(s)$
  - $s \models \Box\phi :\iff \forall t.(s \twoheadrightarrow t) \implies t \models \phi$
  - $\Diamond\phi :\iff \neg\Box\neg\phi$
- Logical equivalence

$$s \approx t \;:\iff\; (s \models \phi \iff t \models \phi) \text{ for all formulae } \phi$$



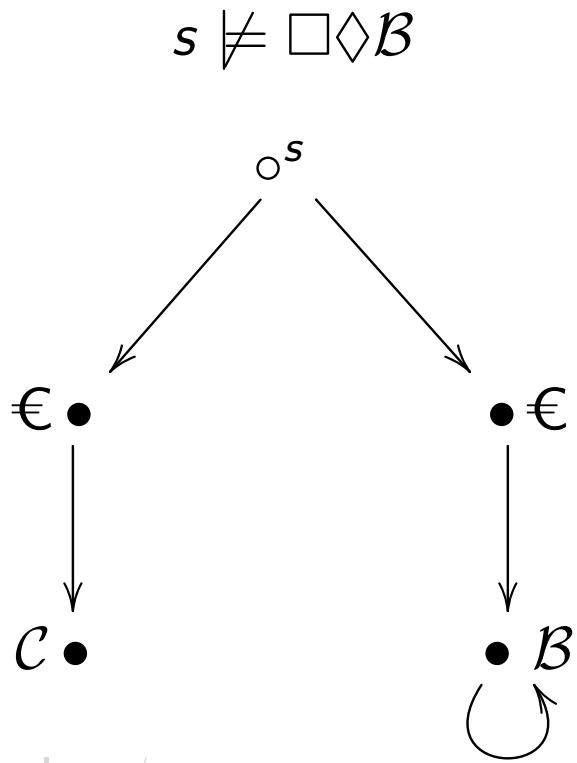
Modal Kripke Logic

- Formulae

$$\begin{array}{rcl} \phi & :: & p \in AP \\ & | & true \mid \neg\phi \mid \phi_1 \vee \phi_2 \\ & | & \Box\phi \mid \Diamond\phi \end{array}$$

- Semantics
  - $s \models p :\iff p \in I(s)$
  - $s \models \Box\phi :\iff \forall t.(s \twoheadrightarrow t) \implies t \models \phi$
  - $\Diamond\phi :\iff \neg\Box\neg\phi$
- Logical equivalence

$$s \approx t \;:\iff\; (s \models \phi \iff t \models \phi) \text{ for all formulae } \phi$$

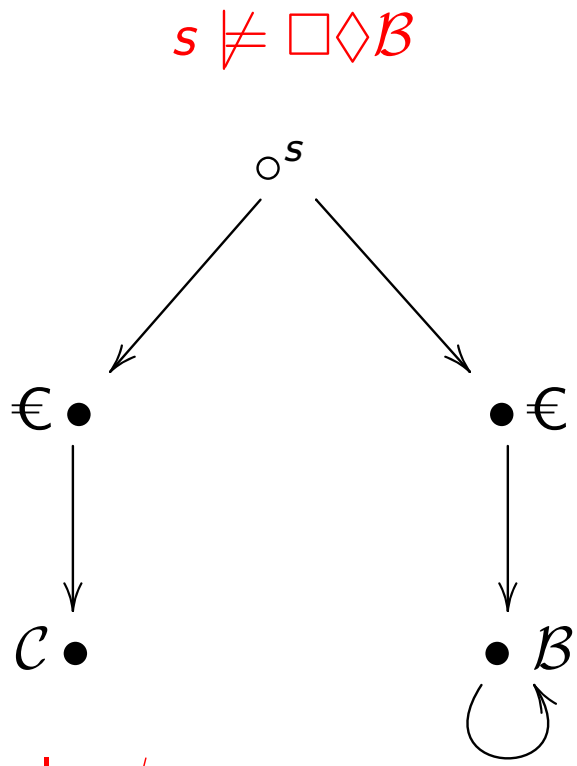


Modal Kripke Logic

- Formulae

$$\begin{array}{rcl} \phi & :: & p \in AP \\ & | & true \mid \neg\phi \mid \phi_1 \vee \phi_2 \\ & | & \Box\phi \mid \Diamond\phi \end{array}$$

- Semantics
  - $s \models p :\iff p \in I(s)$
  - $s \models \Box\phi :\iff \forall t.(s \twoheadrightarrow t) \implies t \models \phi$
  - $\Diamond\phi :\iff \neg\Box\neg\phi$



- Logical equivalence

$s \approx t :\iff (s \models \phi \iff t \models \phi)$ for all formulae ϕ

Hennessy-Milner Theorem

- Adequacy: $s \sim t \implies s \approx t$
- Expressiveness: $s \approx t \stackrel{?}{\implies} s \sim t$
  - needs bounded nondeterminism

Theorem (Hennessy,Milner)

If $\mathcal{K}$ is image − finite then $s \approx t \iff s \sim t$.

Hennessy-Milner Theorem

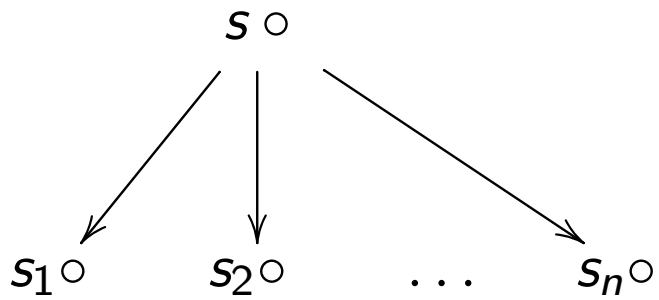
- Adequacy: $s \sim t \implies s \approx t$
- Expressiveness: $s \approx t \overset{?}{\implies} s \sim t$
  - needs bounded nondeterminism

Theorem (Hennessy,Milner)

If $\mathcal{K}$ is image − finite then $s \approx t \iff s \sim t$.

Hennessy-Milner Theorem

- Adequacy: $s \sim t \implies s \approx t$
- Expressiveness: $s \approx t \overset{?}{\implies} s \sim t$
  - needs bounded nondeterminism

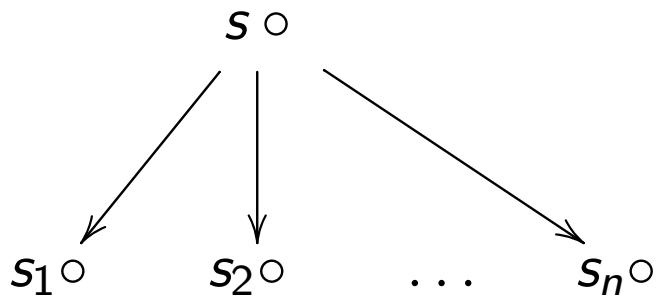


Theorem (Hennessy,Milner)

If $\mathcal{K}$ is image − finite then $s \approx t \iff s \sim t$.

Hennessy-Milner Theorem

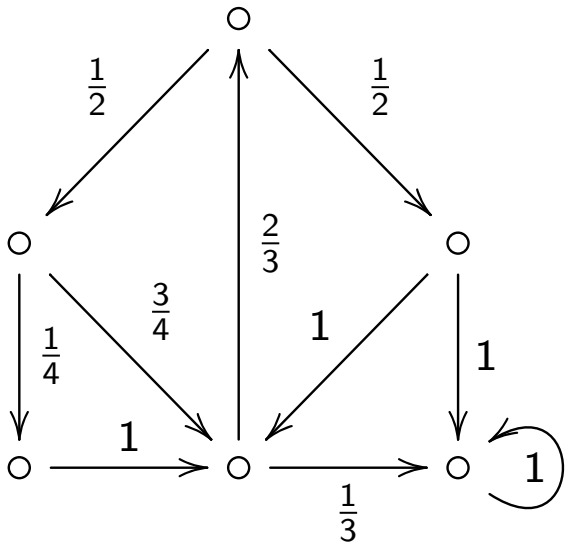
- Adequacy: $s \sim t \implies s \approx t$
- Expressiveness: $s \approx t \overset{?}{\implies} s \sim t$
  - needs bounded nondeterminism



Theorem (Hennessy,Milner)

If $\mathcal{K}$ is image − finite then $s \approx t \iff s \sim t$.

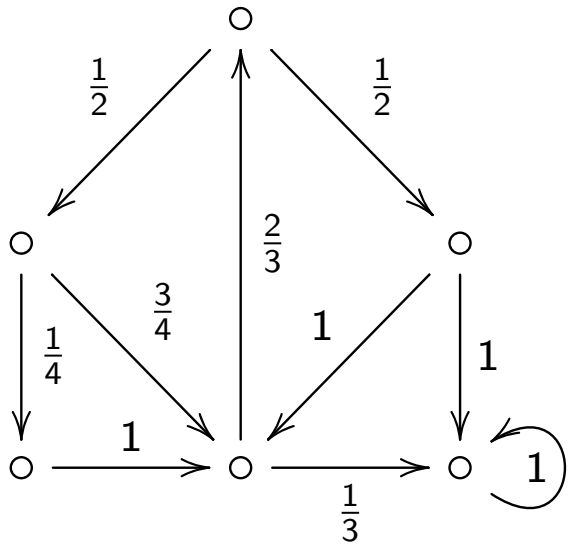
Probabilistic systems



Probabilistic System

- $f : S \times S \to [0,1]$
- $\forall s \in S.\ \sum_{s' \in S} \{w \mid s \overset{w}{\to} s'\} = 1$

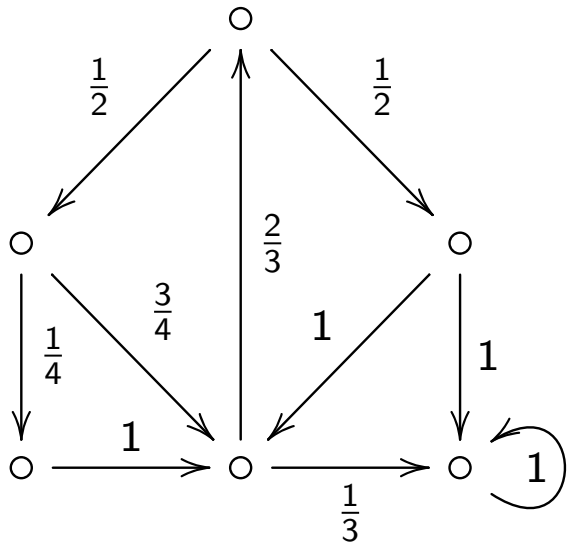
Probabilistic systems



Probabilistic System

- $f : S \times S \to [0, 1]$
- $\forall s \in S. \sum_{s' \in S} \{w \mid s \overset{w}{\to} s'\} = 1$

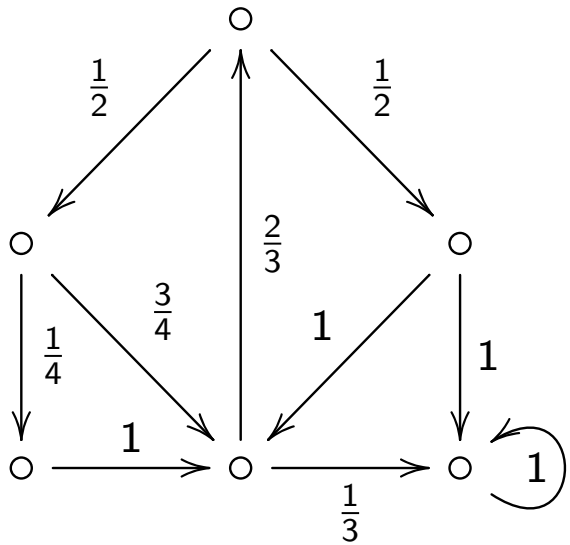
Probabilistic systems



Probabilistic System

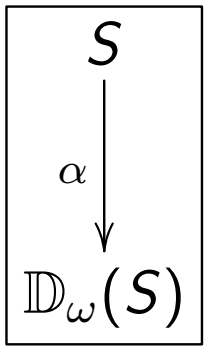
- $f : S \times S \to [0, 1]$
- $\forall s \in S.\ \sum_{s' \in S} \{w \mid s \overset{w}{\twoheadrightarrow} s'\} = 1$

Probabilistic systems



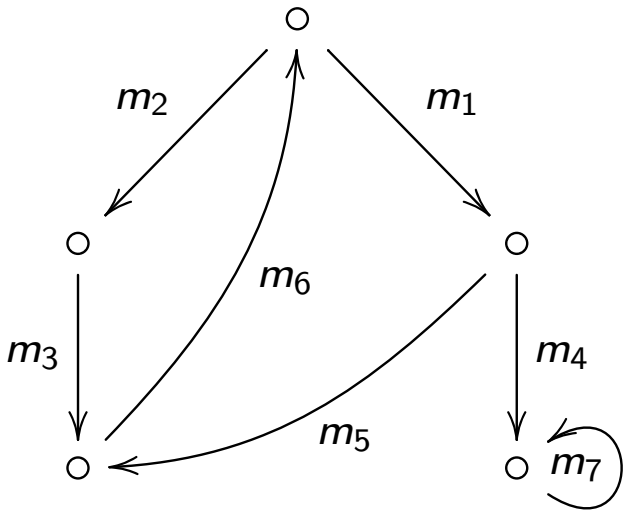
Probabilistic System

- $f : S \times S \to [0,1]$
- $\forall s \in S.\ \sum_{s' \in S} \{w \mid s \overset{w}{\twoheadrightarrow} s'\} = 1$



Monoid labeled systems

$\mathcal{M}$ any commutative monoid



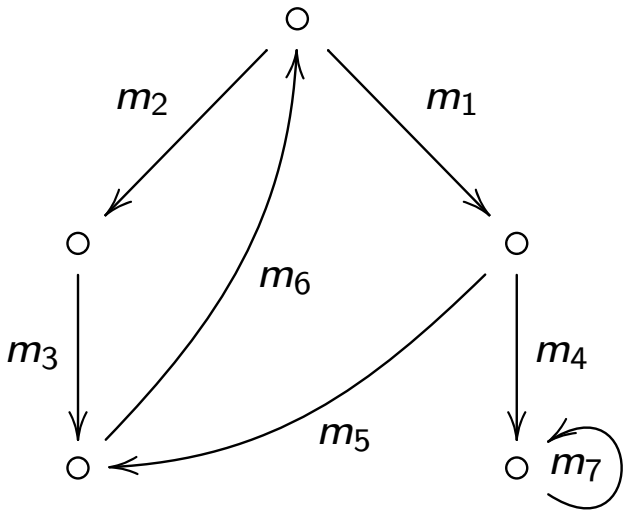
Monoid labeled system

- $f : S \times S \to \mathcal{M}$
- $\forall s \in S. \sum_{s' \in S} f(s, s')$ exists

$$S \xrightarrow{\alpha} \mathcal{M}_\omega^S$$

Monoid labeled systems

$\mathcal{M}$ any commutative monoid



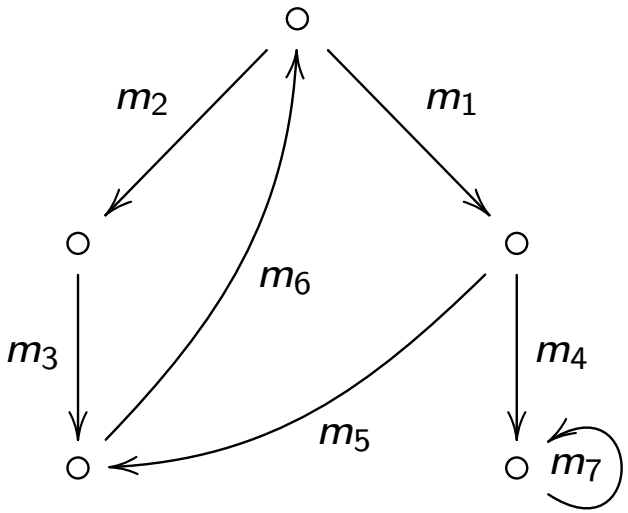
Monoid labeled system

- $f : S \times S \to \mathcal{M}$
- $\forall s \in S. \sum_{s' \in S} f(s, s')$ exists

$$S \xrightarrow{\alpha} \mathcal{M}_\omega^S$$

Monoid labeled systems

$\mathcal{M}$ any commutative monoid

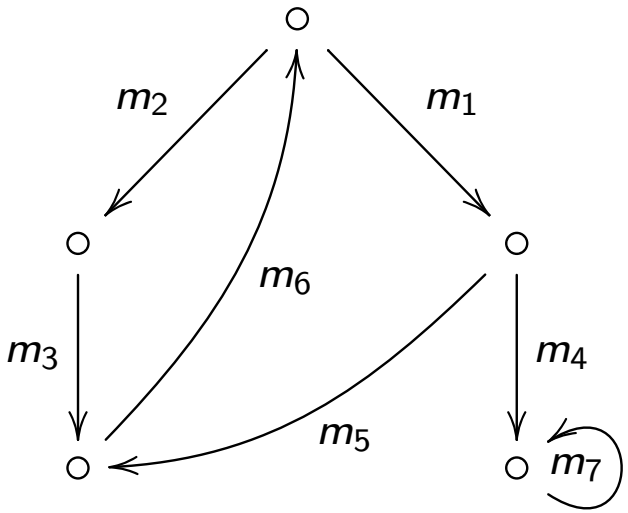


Monoid labeled system

- $f : S \times S \to \mathcal{M}$
- $\forall s \in S.\ \sum_{s' \in S} f(s, s')$ exists

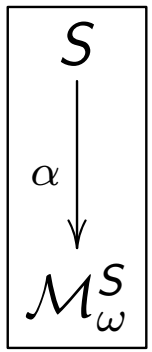
Monoid labeled systems

$\mathcal{M}$ any commutative monoid



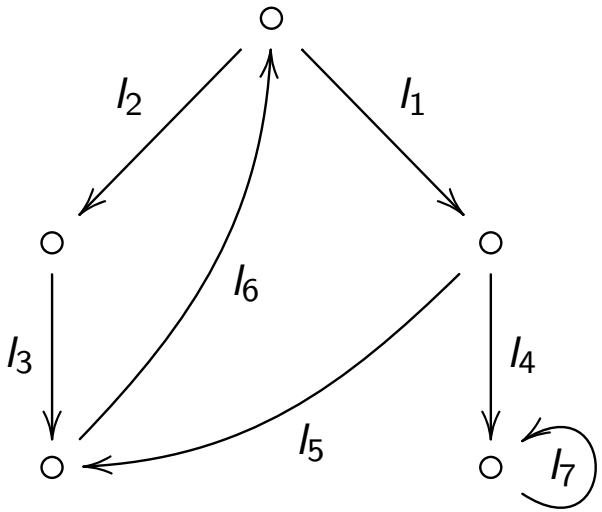
Monoid labeled system

- $f : S \times S \to \mathcal{M}$
- $\forall s \in S. \sum_{s' \in S} f(s, s')$ exists



Lattice labeled systems

$\mathcal{L}$ any complete lattice



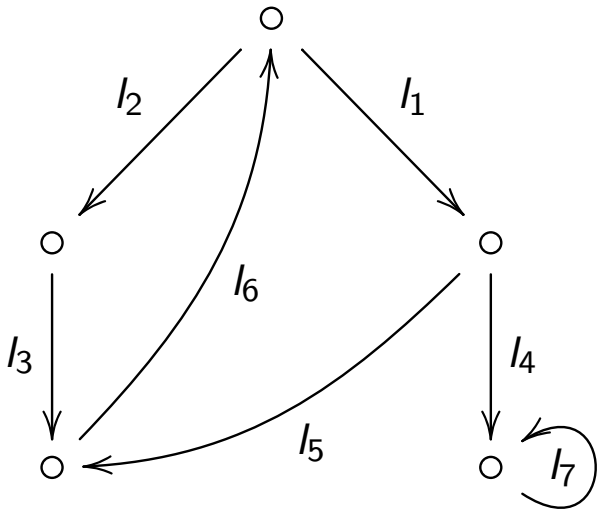
Lattice labeled system

- $f : S \times S \to \mathcal{M}$

$$\begin{array}{c} S \\ \downarrow \alpha \\ \mathcal{L}^S \end{array}$$

Lattice labeled systems

$\mathcal{L}$ any complete lattice



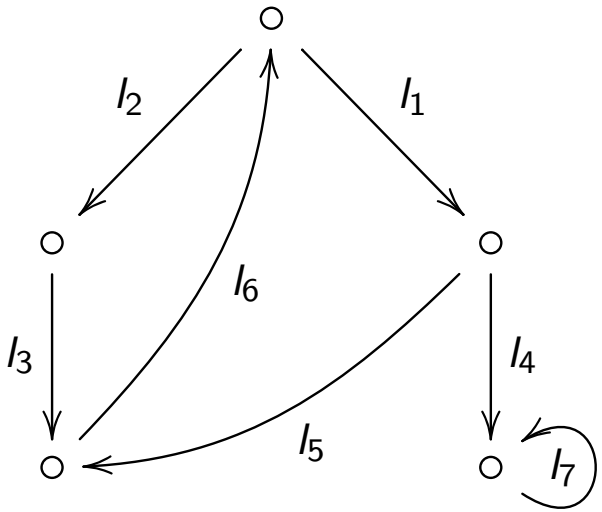
Lattice labeled system

- $f : S \times S \to \mathcal{M}$

$$S \xrightarrow{\alpha} \mathcal{L}^S$$

Lattice labeled systems

$\mathcal{L}$ any complete lattice



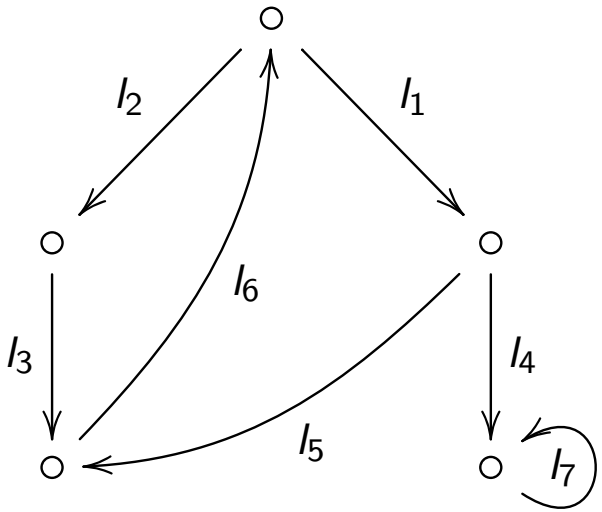
Lattice labeled system

- $f : S \times S \to \mathcal{M}$

$$S \xrightarrow{\alpha} \mathcal{L}^S$$

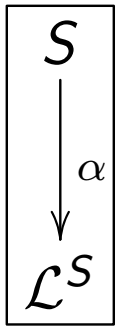
Lattice labeled systems

$\mathcal{L}$ any complete lattice



Lattice labeled system

- $f : S \times S \to \mathcal{M}$



Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	
Acceptor		
Bank Account		
Transition System		
Probabilistic system		
topological space		
F-coalgebra		

Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor		
Bank Account		
Transition System		
Probabilistic system		
topological space		
F-coalgebra		

Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor	$T \subseteq S$ $\delta : S \times \Sigma \to S$	
Bank Account		
Transition System		
Probabilistic system		
topological space		
F-coalgebra		

Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor	$T \subseteq S$ $\delta : S \times \Sigma \to S$	$S \to 2 \times S^{\Sigma}$
Bank Account		
Transition System		
Probabilistic system		
topological space		
F-coalgebra		

Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor	$T \subseteq S$ $\delta : S \times \Sigma \to S$	$S \to 2 \times S^{\Sigma}$
Bank Account	$show : S \to \mathbb{Z}$ $trans : S \times \mathbb{Z} \to S$	
Transition System		
Probabilistic system		
topological space		
F-coalgebra		

Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor	$T \subseteq S$ $\delta : S \times \Sigma \to S$	$S \to 2 \times S^{\Sigma}$
Bank Account	$show : S \to \mathbb{Z}$ $trans : S \times \mathbb{Z} \to S$	$S \to \mathbb{Z} \times S^{\mathbb{Z}}$
Transition System		
Probabilistic system		
topological space		
F-coalgebra		

Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor	$T \subseteq S$ $\delta : S \times \Sigma \to S$	$S \to 2 \times S^{\Sigma}$
Bank Account	$show : S \to \mathbb{Z}$ $trans : S \times \mathbb{Z} \to S$	$S \to \mathbb{Z} \times S^{\mathbb{Z}}$
Transition System	$l : Prop \to \mathbb{P}(S)$ $R \subseteq S \times S$	
Probabilistic system		
topological space		
F-coalgebra		

Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor	$T \subseteq S$ $\delta : S \times \Sigma \to S$	$S \to 2 \times S^{\Sigma}$
Bank Account	$show : S \to \mathbb{Z}$ $trans : S \times \mathbb{Z} \to S$	$S \to \mathbb{Z} \times S^{\mathbb{Z}}$
Transition System	$l : Prop \to \mathbb{P}(S)$ $R \subseteq S \times S$	$S \to \mathbb{P}(Prop) \times \mathbb{P}(S)$
Probabilistic system		
topological space		
F-coalgebra		

Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor	$T \subseteq S$ $\delta : S \times \Sigma \to S$	$S \to 2 \times S^{\Sigma}$
Bank Account	$show : S \to \mathbb{Z}$ $trans : S \times \mathbb{Z} \to S$	$S \to \mathbb{Z} \times S^{\mathbb{Z}}$
Transition System	$l : Prop \to \mathbb{P}(S)$ $R \subseteq S \times S$	$S \to \mathbb{P}(Prop) \times \mathbb{P}(S)$
Probabilistic system	$p : S \times S \to [0, 1]$ $\sum_{y \in X} p(x, y) = 1$	
topological space		
F-coalgebra		

Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor	$T \subseteq S$ $\delta : S \times \Sigma \to S$	$S \to 2 \times S^{\Sigma}$
Bank Account	$show : S \to \mathbb{Z}$ $trans : S \times \mathbb{Z} \to S$	$S \to \mathbb{Z} \times S^{\mathbb{Z}}$
Transition System	$l : Prop \to \mathbb{P}(S)$ $R \subseteq S \times S$	$S \to \mathbb{P}(Prop) \times \mathbb{P}(S)$
Probabilistic system	$p : S \times S \to [0,1]$ $\sum_{y \in X} p(x,y) = 1$	$S \to \mathbb{D}_{\omega}(S)$
topological space		
F-coalgebra		

Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor	$T \subseteq S$ $\delta : S \times \Sigma \to S$	$S \to 2 \times S^{\Sigma}$
Bank Account	$show : S \to \mathbb{Z}$ $trans : S \times \mathbb{Z} \to S$	$S \to \mathbb{Z} \times S^{\mathbb{Z}}$
Transition System	$l : Prop \to \mathbb{P}(S)$ $R \subseteq S \times S$	$S \to \mathbb{P}(Prop) \times \mathbb{P}(S)$
Probabilistic system	$p : S \times S \to [0,1]$ $\sum_{y \in X} p(x,y) = 1$	$S \to \mathbb{D}_{\omega}(S)$
topological space	$\tau \subseteq \mathbb{P}(S)$	
F-coalgebra		

Systems to coalgebras

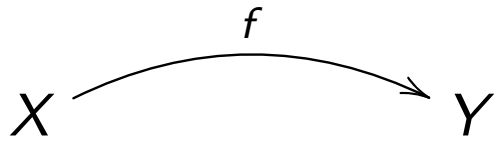
Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor	$T \subseteq S$ $\delta : S \times \Sigma \to S$	$S \to 2 \times S^{\Sigma}$
Bank Account	$show : S \to \mathbb{Z}$ $trans : S \times \mathbb{Z} \to S$	$S \to \mathbb{Z} \times S^{\mathbb{Z}}$
Transition System	$l : Prop \to \mathbb{P}(S)$ $R \subseteq S \times S$	$S \to \mathbb{P}(Prop) \times \mathbb{P}(S)$
Probabilistic system	$p : S \times S \to [0,1]$ $\sum_{y \in X} p(x,y) = 1$	$S \to \mathbb{D}_{\omega}(S)$
topological space	$\tau \subseteq \mathbb{P}(S)$	$S \to \mathbb{F}il(S)$
F-coalgebra		

Systems to coalgebras

Name	**Structure**	**Coalgebra**
Black Box	$h : S \to D$ $t : S \to S$	$S \to D \times S$
Acceptor	$T \subseteq S$ $\delta : S \times \Sigma \to S$	$S \to 2 \times S^{\Sigma}$
Bank Account	$show : S \to \mathbb{Z}$ $trans : S \times \mathbb{Z} \to S$	$S \to \mathbb{Z} \times S^{\mathbb{Z}}$
Transition System	$l : Prop \to \mathbb{P}(S)$ $R \subseteq S \times S$	$S \to \mathbb{P}(Prop) \times \mathbb{P}(S)$
Probabilistic system	$p : S \times S \to [0,1]$ $\sum_{y \in X} p(x,y) = 1$	$S \to \mathbb{D}_{\omega}(S)$
topological space	$\tau \subseteq \mathbb{P}(S)$	$S \to \mathbb{F}il(S)$
F-coalgebra		$S \to F(S)$

Relevant Properties of $\mathcal{S}et$

- epi-regular mono factorization



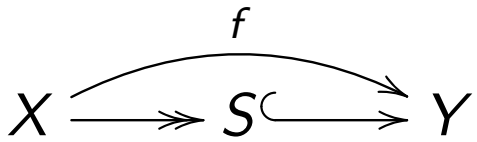
- $E - M$-property

- currying-uncurrying
- Axiom of choice

$e \circ e^{-} = id$

Relevant Properties of $\mathcal{S}et$

- epi-regular mono factorization



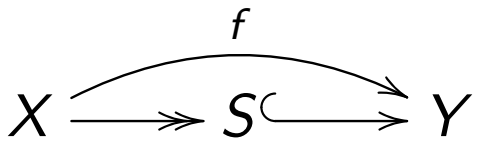
- $E - M$-property

$$e \circ e^{-} = id$$

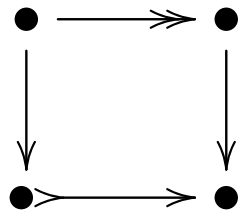
- currying-uncurrying
- Axiom of choice

Relevant Properties of $\mathcal{S}et$

- epi-regular mono factorization



- $E - M$-property

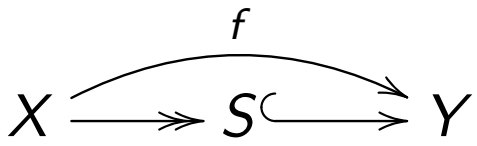


- currying-uncurrying
- Axiom of choice

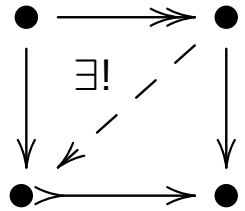
$e \circ e^- = id$

Relevant Properties of $\mathcal{S}et$

- epi-regular mono factorization



- $E - M$-property

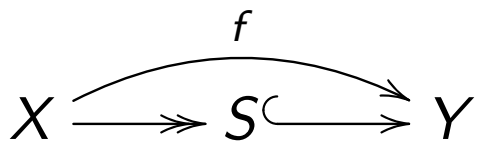


- currying-uncurrying
- Axiom of choice

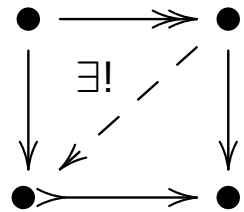
$e \circ e^{-} = id$

Relevant Properties of $\mathcal{S}et$

- epi-regular mono factorization



- $E - M$-property



- currying-uncurrying

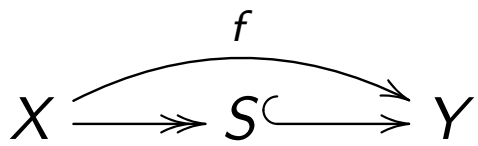
$$A \times B \to C \cong A \to C^B$$

- Axiom of choice

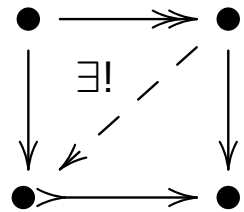
$$e \circ e^{-} = id$$

Relevant Properties of $\mathcal{S}et$

- epi-regular mono factorization



- $E - M$-property



- currying-uncurrying

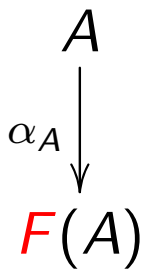
$$A \times B \to C \cong A \to C^B$$

- Axiom of choice

$$e \circ e^{-} = id$$

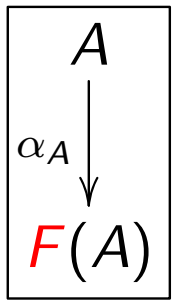
The notion of F-coalgebra

- Type (signature) F:
  - $F : \mathcal{S}et \to \mathcal{S}et$
- F-Coalgebra
  - $\alpha : A \to F(A)$
- Homomorphism
- Category: $\mathcal{S}et_F$



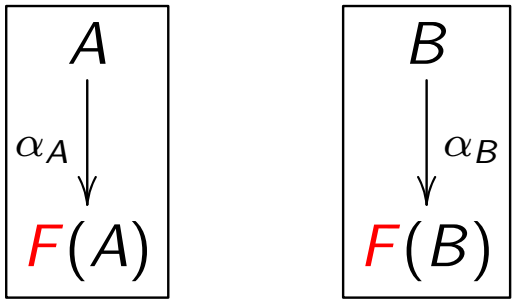
The notion of F-coalgebra

- Type (signature) F:
  - $F : \mathcal{S}et \to \mathcal{S}et$
- F-Coalgebra
  - $\alpha : A \to F(A)$
- Homomorphism
- Category: $\mathcal{S}et_F$



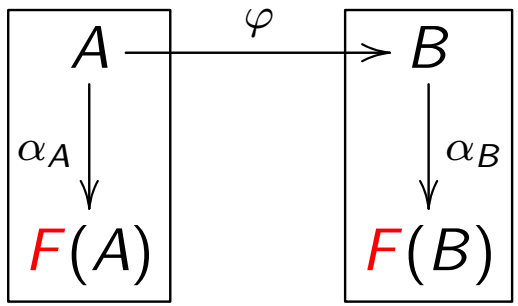
The notion of F-coalgebra

- Type (signature) F:
  - $F : \mathcal{S}et \to \mathcal{S}et$
- F-Coalgebra
  - $\alpha : A \to F(A)$
- Homomorphism
- Category: $\mathcal{S}et_F$



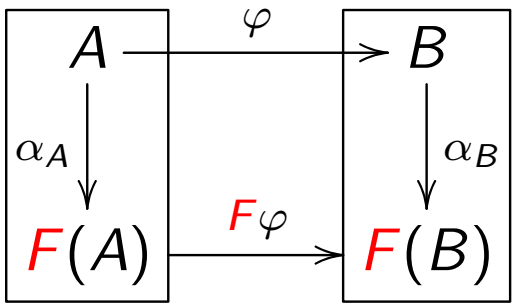
The notion of F-coalgebra

- Type (signature) F:
  - $F : \mathcal{S}et \to \mathcal{S}et$
- F-Coalgebra
  - $\alpha : A \to F(A)$
- Homomorphism
- Category: $\mathcal{S}et_F$



The notion of F-coalgebra

- Type (signature) F:
  - $F : \mathcal{S}et \to \mathcal{S}et$
- F-Coalgebra
  - $\alpha : A \to F(A)$
- Homomorphism
- Category: $\mathcal{S}et_F$



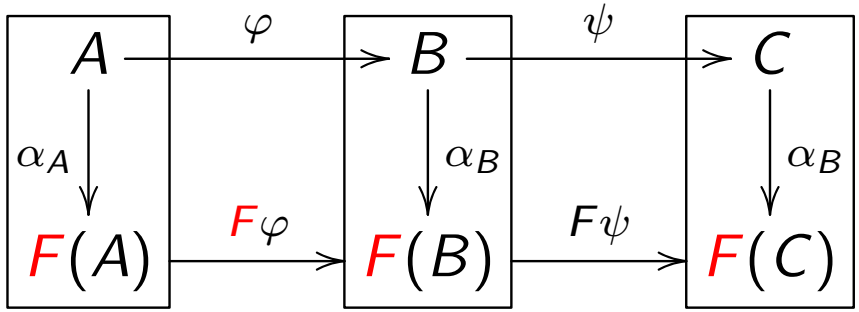
The notion of F-coalgebra

- Type (signature) F:
  - $F : \mathcal{S}et \to \mathcal{S}et$
- F-Coalgebra
  - $\alpha : A \to F(A)$
- Homomorphism
- Category: $\mathcal{S}et_F$

$$\mathcal{A} \xrightarrow{\varphi} \mathcal{B}$$

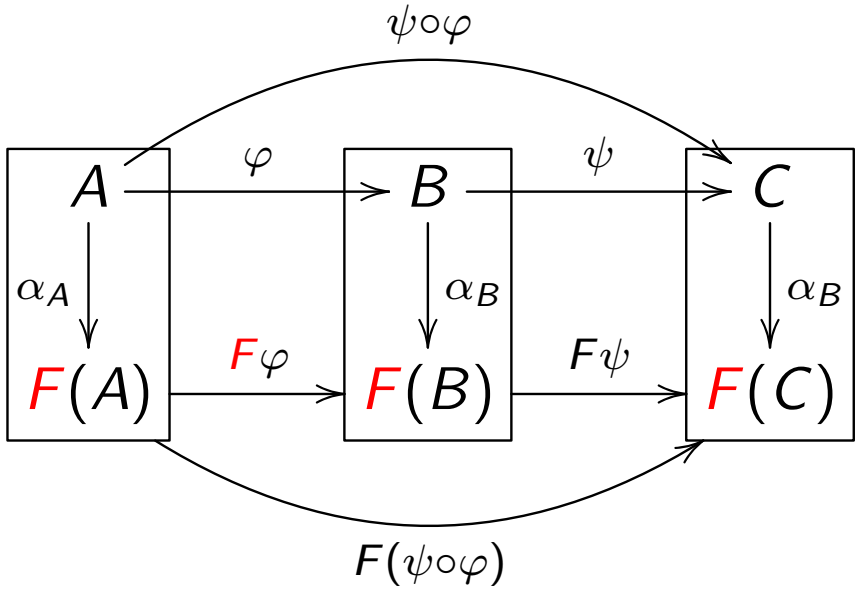
The notion of F-coalgebra

- Type (signature) F:
  - $F : \mathcal{S}et \to \mathcal{S}et$
- F-Coalgebra
  - $\alpha : A \to F(A)$
- Homomorphism
- Category: $\mathcal{S}et_F$



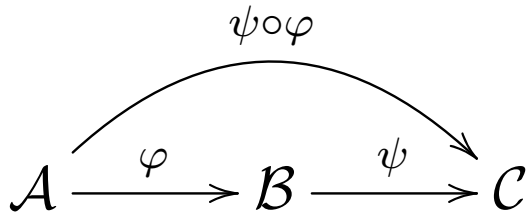
The notion of F-coalgebra

- Type (signature) F:
  - $F : \mathcal{S}et \to \mathcal{S}et$
- F-Coalgebra
  - $\alpha : A \to F(A)$
- Homomorphism
- Category: $\mathcal{S}et_F$



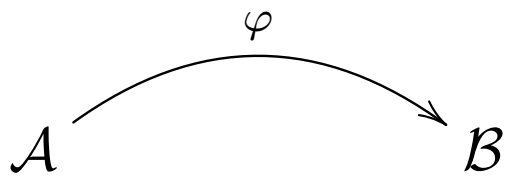
The notion of F-coalgebra

- Type (signature) F:
  - $F : \mathcal{S}et \to \mathcal{S}et$
- F-Coalgebra
  - $\alpha : A \to F(A)$
- Homomorphism
- Category: $\mathcal{S}et_F$



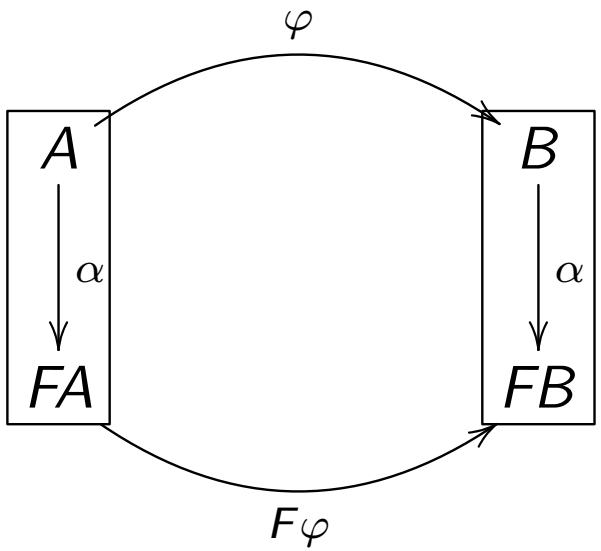
Homomorphism theorem

- $\varphi : \mathcal{A} \to \mathcal{B}$ homomorphism.
- Set has epi-mono-factorization
- F preserves epis and monos
- diagonal yields structure
- Set_F has epi-mono-factorization



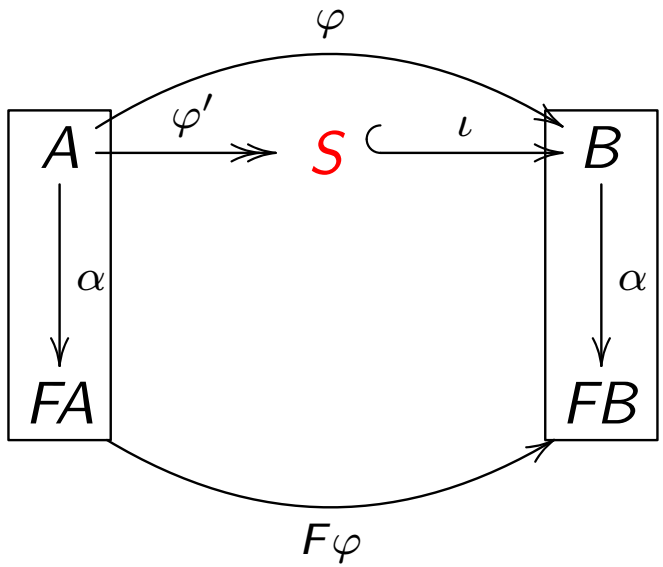
Homomorphism theorem

- $\varphi : \mathcal{A} \to \mathcal{B}$ homomorphism.
- Set has epi-mono-factorization
- F preserves epis and monos
- diagonal yields structure
- Set_F has epi-mono-factorization



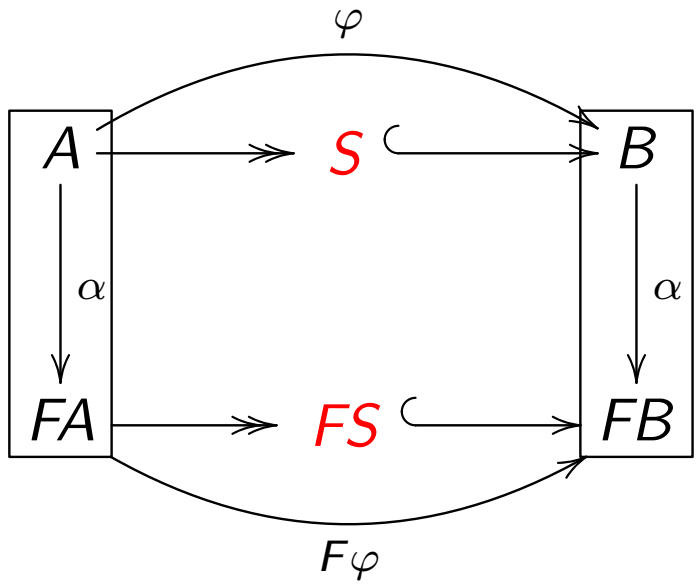
Homomorphism theorem

- $\varphi : \mathcal{A} \to \mathcal{B}$ homomorphism.
- *Set* has epi-mono-factorization
- F preserves epis and monos
- diagonal yields structure
- Set_F has epi-mono-factorization



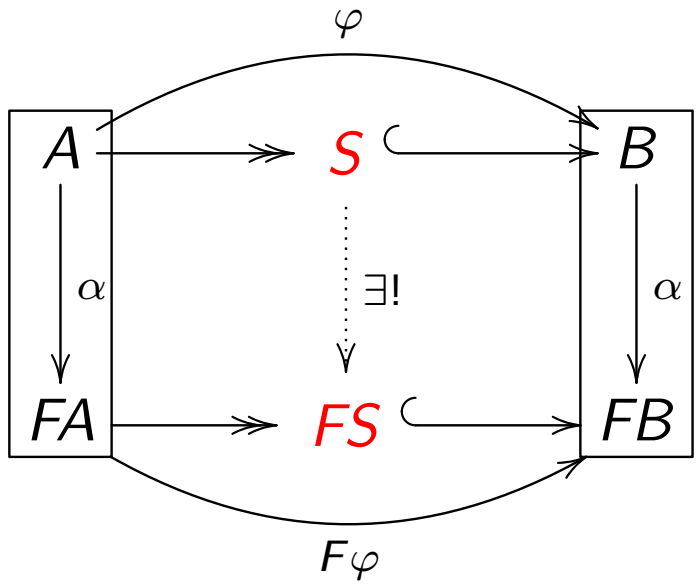
Homomorphism theorem

- $\varphi : \mathcal{A} \to \mathcal{B}$ homomorphism.
- *Set* has epi-mono-factorization
- F preserves epis and monos
- diagonal yields structure
- Set_F has epi-mono-factorization



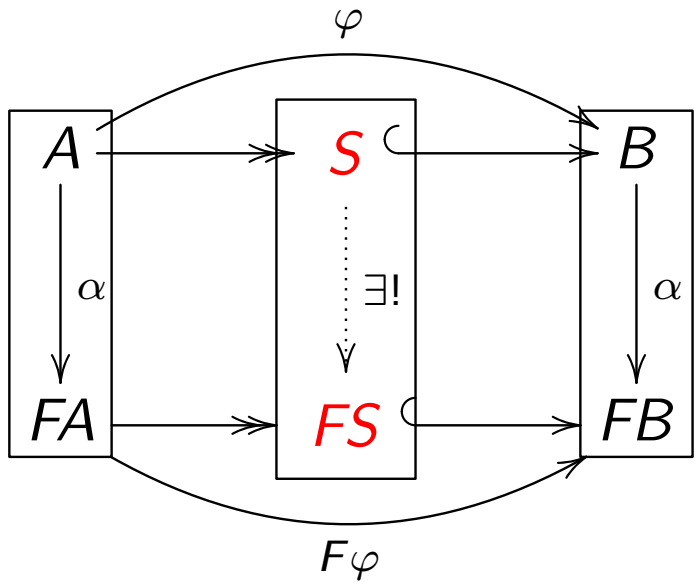
Homomorphism theorem

- $\varphi : \mathcal{A} \to \mathcal{B}$ homomorphism.
- *Set* has epi-mono-factorization
- F preserves epis and monos
- diagonal yields structure
- Set_F has epi-mono-factorization



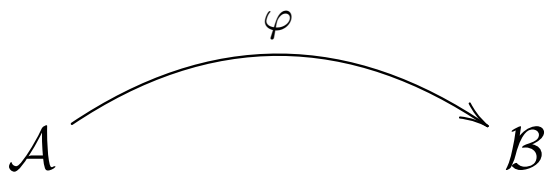
Homomorphism theorem

- $\varphi : \mathcal{A} \to \mathcal{B}$ homomorphism.
- *Set* has epi-mono-factorization
- F preserves epis and monos
- diagonal yields structure
- Set_F has epi-mono-factorization



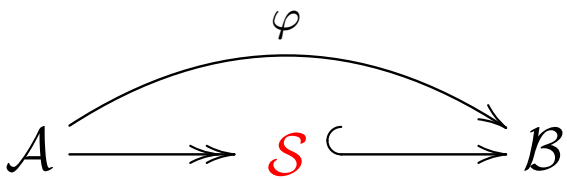
Homomorphism theorem

- $\varphi : \mathcal{A} \to \mathcal{B}$ homomorphism.
- *Set* has epi-mono-factorization
- F preserves epis and monos
- diagonal yields structure
- Set_F has epi-mono-factorization



Homomorphism theorem

- $\varphi : \mathcal{A} \to \mathcal{B}$ homomorphism.
- *Set* has epi-mono-factorization
- F preserves epis and monos
- diagonal yields structure
- Set_F has epi-mono-factorization



Colimits exist canonically

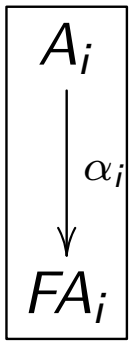
- $\mathcal{S}et_F$ has all colimits
  - Forgetful functor $U : \mathcal{S}et_F \to \mathcal{S}et$
    - ⋆ preserves and
    - ⋆ creates colimits
- Example: Sum of coalgebras $\mathcal{A}_i = (A, \alpha_i)$

Colimits exist canonically

- $\mathcal{S}et_F$ has all colimits
  - Forgetful functor $U : \mathcal{S}et_F \to \mathcal{S}et$
    - ★ preserves and
    - ★ creates colimits
- Example: Sum of coalgebras $\mathcal{A}_i = (A, \alpha_i)$

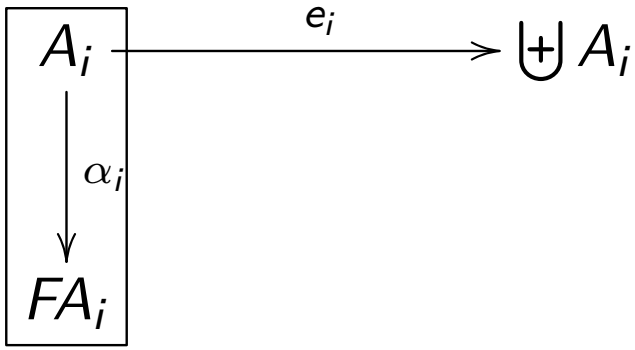
Colimits exist canonically

- $\mathcal{S}et_F$ has all colimits
  - Forgetful functor $U : \mathcal{S}et_F \to \mathcal{S}et$
    - ★ preserves and
    - ★ creates colimits
- Example: Sum of coalgebras $\mathcal{A}_i = (A, \alpha_i)$



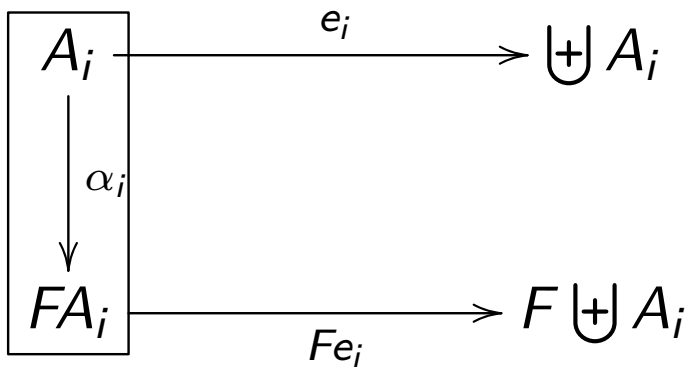
Colimits exist canonically

- $\mathcal{S}et_F$ has all colimits
  - Forgetful functor $U : \mathcal{S}et_F \to \mathcal{S}et$
    - ★ preserves and
    - ★ creates colimits
- Example: Sum of coalgebras $\mathcal{A}_i = (A, \alpha_i)$



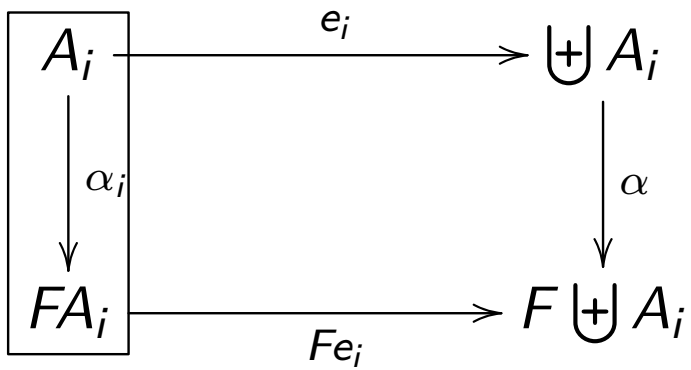
Colimits exist canonically

- $\mathcal{S}et_F$ has all colimits
  - Forgetful functor $U : \mathcal{S}et_F \to \mathcal{S}et$
    - ★ preserves and
    - ★ creates colimits
- Example: Sum of coalgebras $\mathcal{A}_i = (A, \alpha_i)$



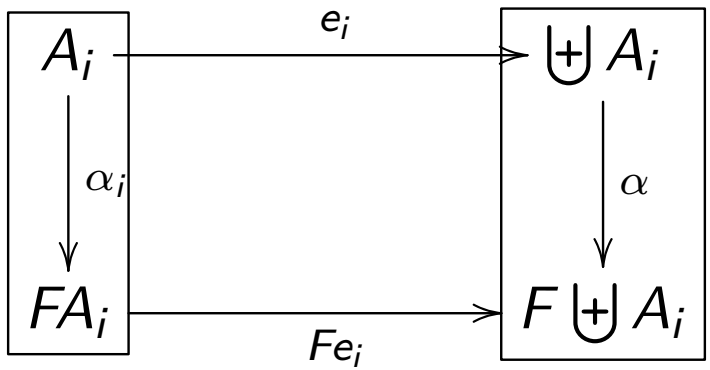
Colimits exist canonically

- $\mathcal{S}et_F$ has all colimits
  - Forgetful functor $U : \mathcal{S}et_F \to \mathcal{S}et$
    - ★ preserves and
    - ★ creates colimits
- Example: Sum of coalgebras $\mathcal{A}_i = (A, \alpha_i)$



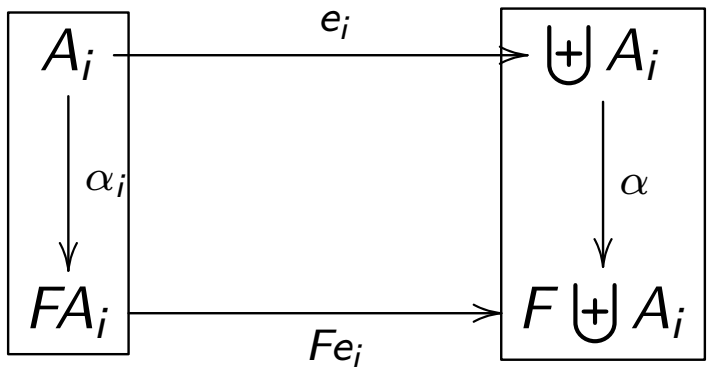
Colimits exist canonically

- $\mathcal{S}et_F$ has all colimits
  - Forgetful functor $U : \mathcal{S}et_F \to \mathcal{S}et$
    - ★ preserves and
    - ★ creates colimits
- Example: Sum of coalgebras $\mathcal{A}_i = (A, \alpha_i)$



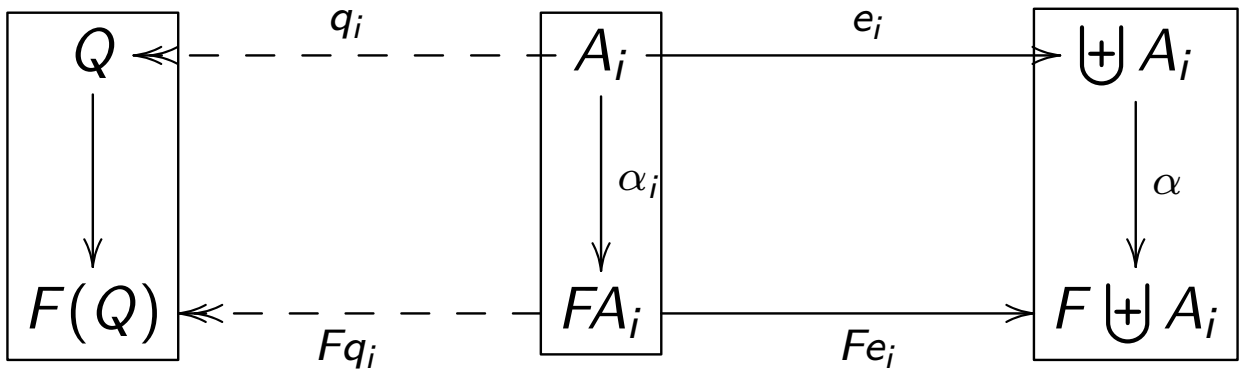
Colimits exist canonically

- $\mathcal{S}et_F$ has all colimits
  - Forgetful functor $U : \mathcal{S}et_F \to \mathcal{S}et$
    - ★ preserves and
    - ★ creates colimits
- Example: Sum of coalgebras $\mathcal{A}_i = (A, \alpha_i)$



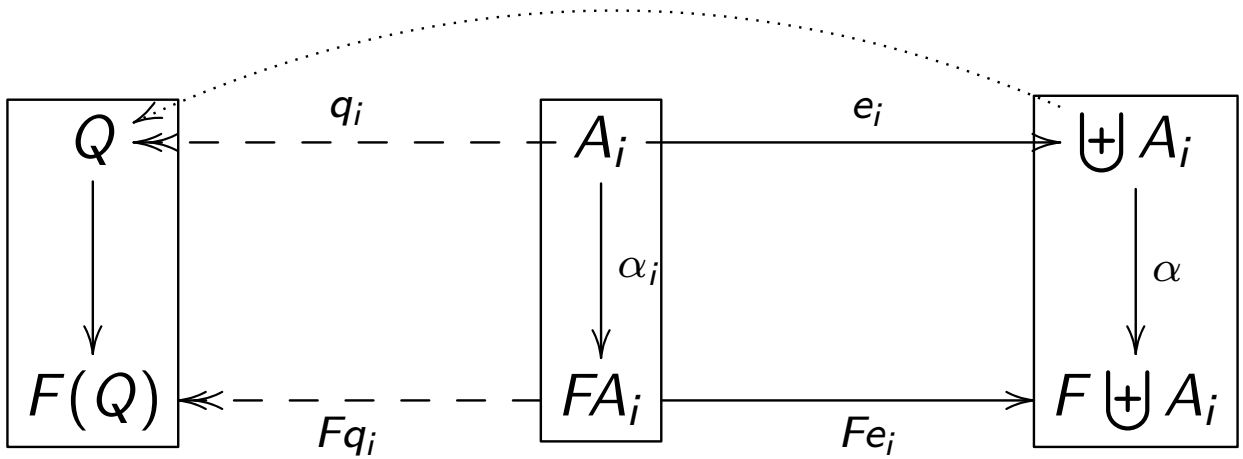
Colimits exist canonically

- $\mathcal{S}et_F$ has all colimits
  - Forgetful functor $U : \mathcal{S}et_F \to \mathcal{S}et$
    - ★ preserves and
    - ★ creates colimits
- Example: Sum of coalgebras $\mathcal{A}_i = (A, \alpha_i)$



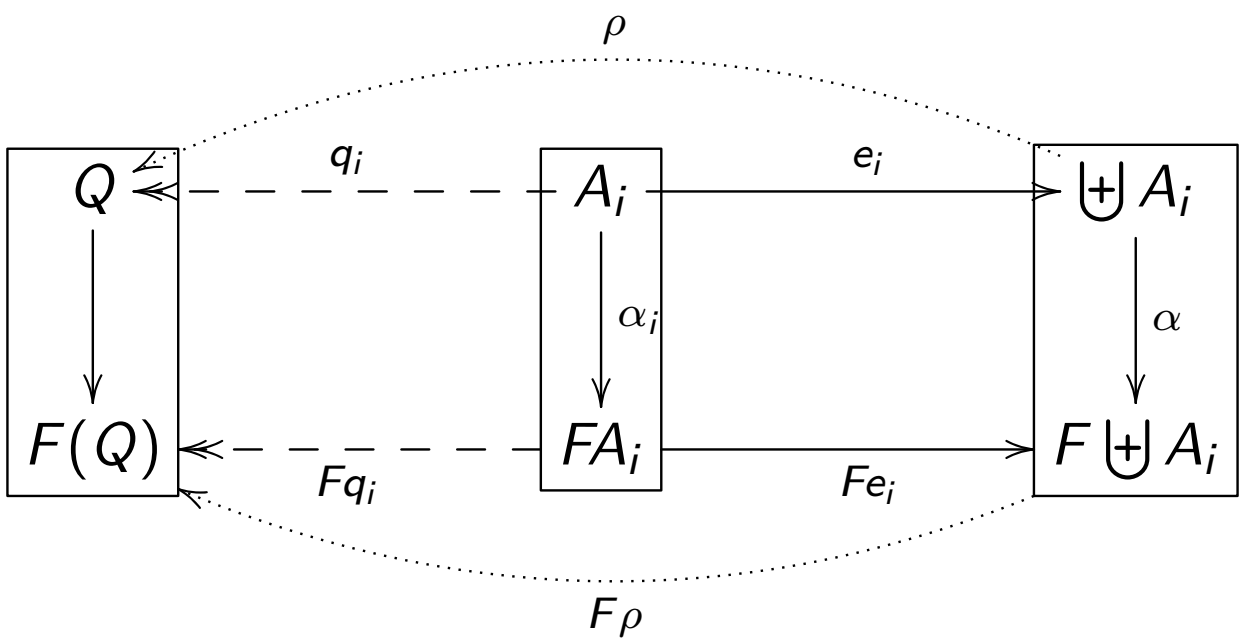
Colimits exist canonically

- $\mathcal{S}et_F$ has all colimits
  - Forgetful functor $U : \mathcal{S}et_F \to \mathcal{S}et$
    - ★ preserves and
    - ★ creates colimits
- Example: Sum of coalgebras $\mathcal{A}_i = (A, \alpha_i)$



Colimits exist canonically

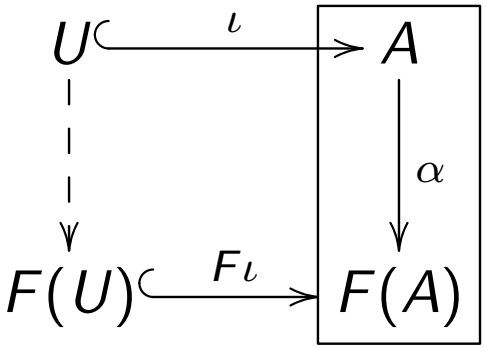
- $\mathcal{S}et_F$ has all colimits
  - Forgetful functor $U : \mathcal{S}et_F \to \mathcal{S}et$
    - ★ preserves and
    - ★ creates colimits
- Example: Sum of coalgebras $\mathcal{A}_i = (A, \alpha_i)$



Subcoalgebras

$\mathcal{A} = (A, \alpha)$ coalgebra

- U subset with coalgebra structure and inclusion homomorphism



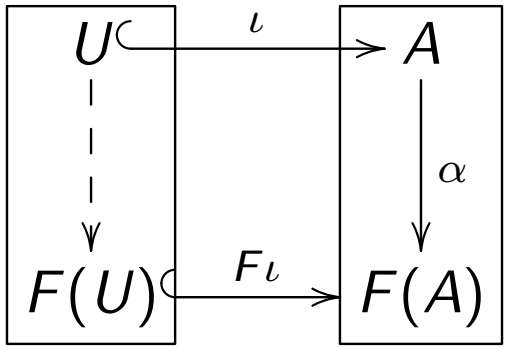
- structure is unique
- each subset $S \subseteq A$ contains largest subcoalgebra of $\mathcal{A}$

$$[S] := \bigcup \{ U \mid U \subseteq S, U \leq \mathcal{A} \}$$

Subcoalgebras

$\mathcal{A} = (A, \alpha)$ coalgebra

- U subset with coalgebra structure and inclusion homomorphism



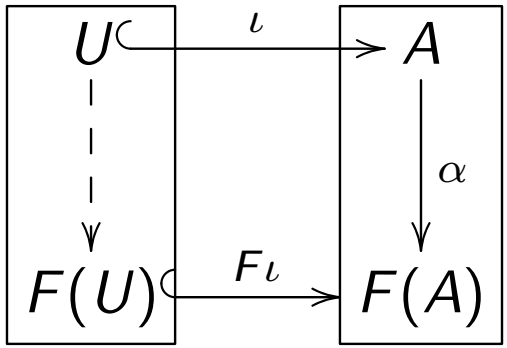
- structure is unique
- each subset $S \subseteq A$ contains largest subcoalgebra of $\mathcal{A}$

$$[S] := \bigcup \{U \mid U \subseteq S, U \leq \mathcal{A}\}$$

Subcoalgebras

$\mathcal{A} = (A, \alpha)$ coalgebra

- U subset with coalgebra structure and inclusion homomorphism



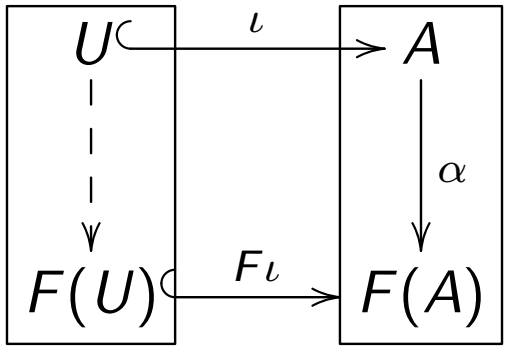
- structure is unique
- each subset $S \subseteq A$ contains largest subcoalgebra of $\mathcal{A}$

$$[S] := \bigcup \{U \mid U \subseteq S, \mathcal{U} \leq \mathcal{A}\}$$

Subcoalgebras

$\mathcal{A} = (A, \alpha)$ coalgebra

- U subset with coalgebra structure and inclusion homomorphism



- structure is unique
- each subset $S \subseteq A$ contains largest subcoalgebra of $\mathcal{A}$

$$[S] := \bigcup \{U \mid U \subseteq S, \mathcal{U} \leq \mathcal{A}\}$$

Factors and congruences

- Epimorphism $\varphi : \mathcal{A} \twoheadrightarrow \mathcal{B}$
  - same as surjective homomorphisms
- Congruences
  - *Set*-kernels of homomorphisms
- Diagram Lemma

Factors and congruences

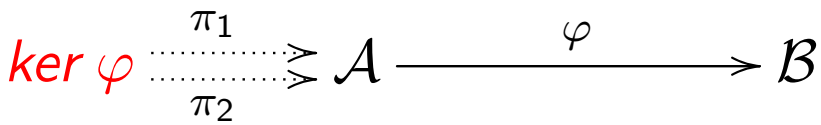
- Epimorphism $\varphi : \mathcal{A} \twoheadrightarrow \mathcal{B}$
  - same as surjective homomorphisms
- Congruences
  - *Set*-kernels of homomorphisms

$$\mathcal{A} \xrightarrow{\quad\varphi\quad} \mathcal{B}$$

- Diagram Lemma

Factors and congruences

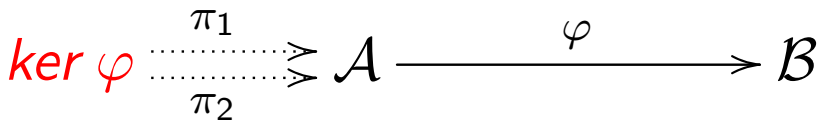
- Epimorphism $\varphi : \mathcal{A} \twoheadrightarrow \mathcal{B}$
  - same as surjective homomorphisms
- Congruences
  - *Set*-kernels of homomorphisms



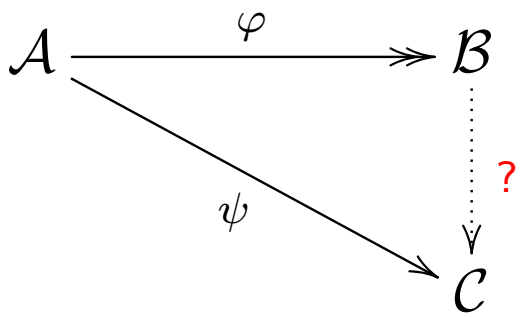
- Diagram Lemma

Factors and congruences

- Epimorphism $\varphi : \mathcal{A} \twoheadrightarrow \mathcal{B}$
  - same as surjective homomorphisms
- Congruences
  - *Set*-kernels of homomorphisms

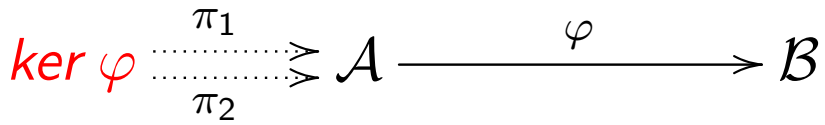


- Diagram Lemma

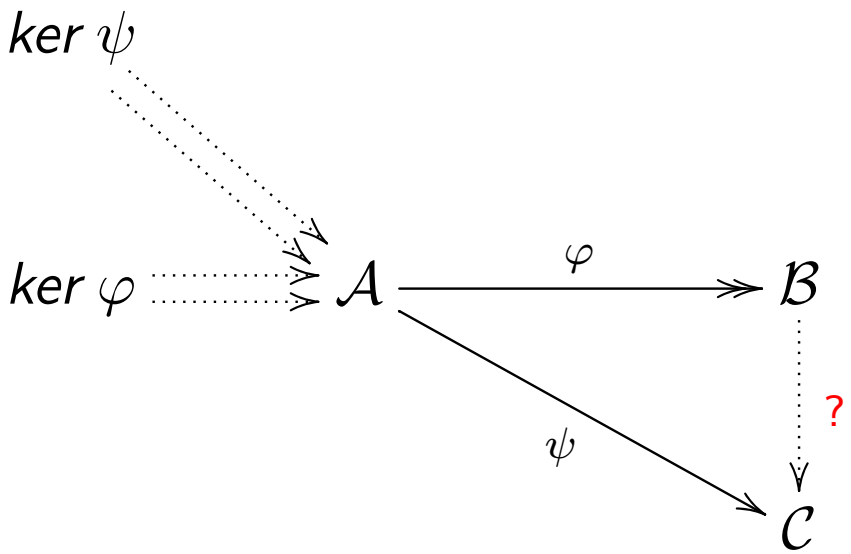


Factors and congruences

- Epimorphism $\varphi : \mathcal{A} \twoheadrightarrow \mathcal{B}$
  - same as surjective homomorphisms
- Congruences
  - *Set*-kernels of homomorphisms

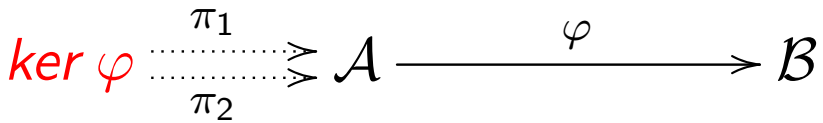


- Diagram Lemma

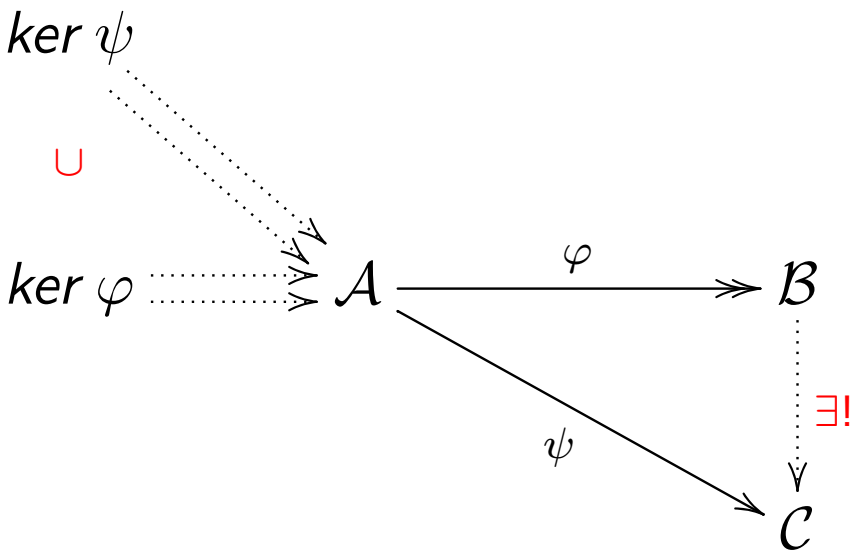


Factors and congruences

- Epimorphism $\varphi : \mathcal{A} \twoheadrightarrow \mathcal{B}$
  - same as surjective homomorphisms
- Congruences
  - *Set*-kernels of homomorphisms

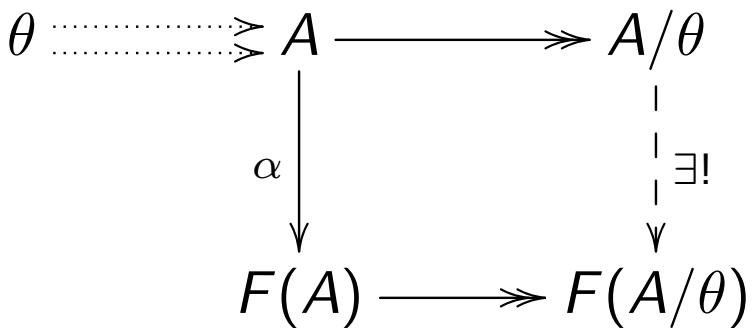


- Diagram Lemma



Factors

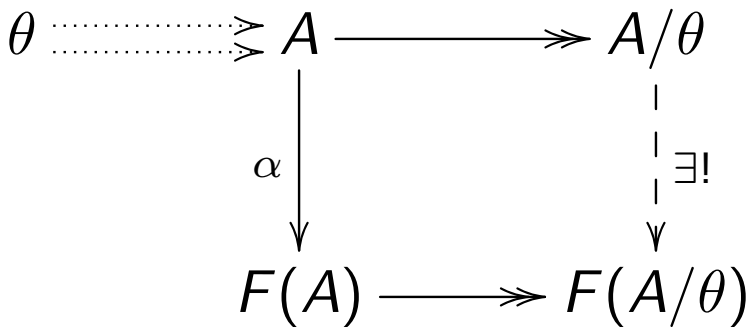
- congruences uniquely determine factors



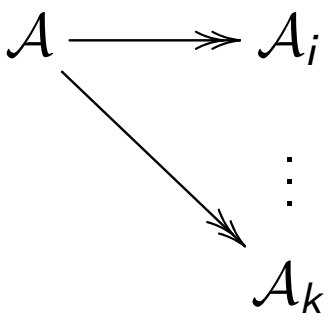
- Smallest factor = Colimit of all factors

Factors

- congruences uniquely determine factors

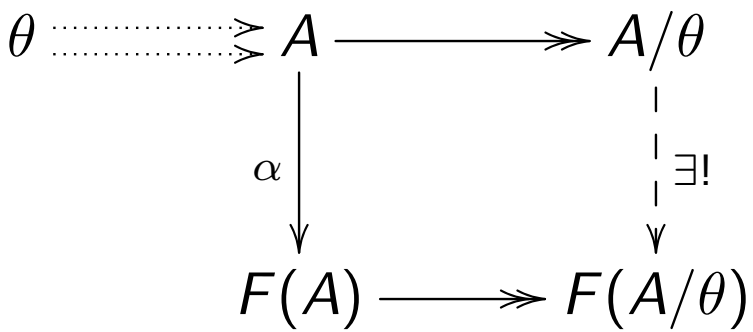


- Smallest factor = Colimit of all factors

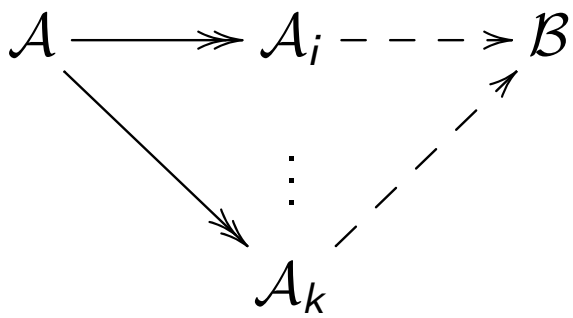


Factors

- congruences uniquely determine factors

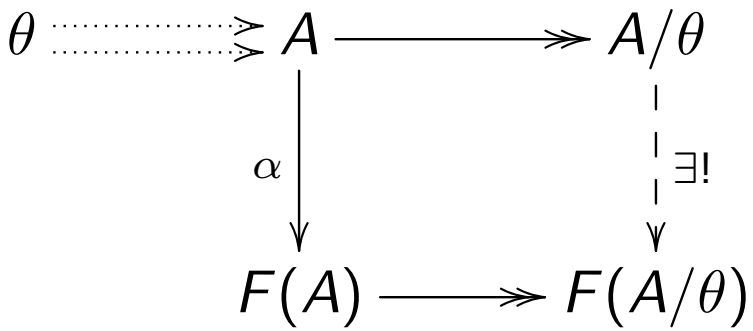


- Smallest factor = Colimit of all factors



Factors

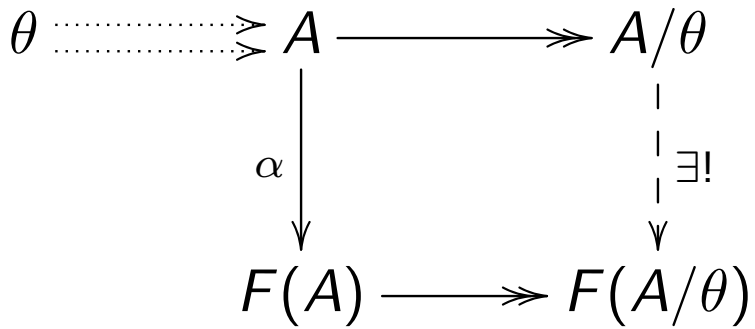
- congruences uniquely determine factors



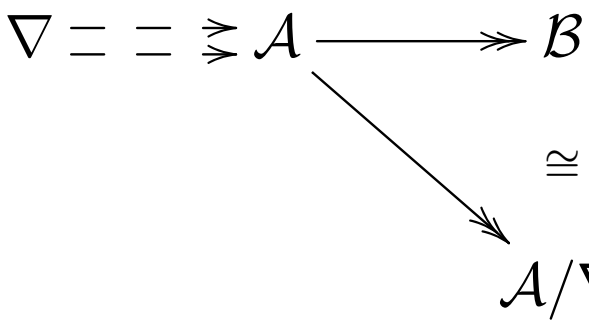
- Smallest factor = Colimit of all factors

Factors

- congruences uniquely determine factors



- Smallest factor = Colimit of all factors



minimal

- largest congruence ∇

Some Limits exist

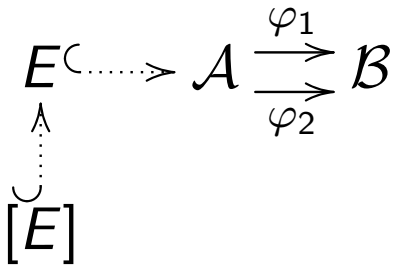
- Equalizers
  ... start with set $E := \{a \in A \mid \varphi_1(a) = \varphi_2(a)\}$ then ...

$$E \hookrightarrow \mathcal{A} \underset{\varphi_2}{\overset{\varphi_1}{\rightrightarrows}} \mathcal{B}$$

- Preimages

Some Limits exist

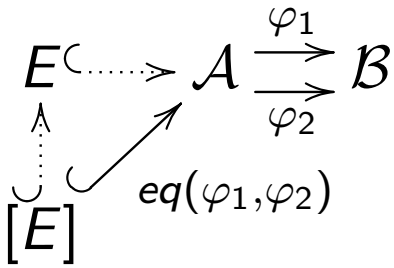
- Equalizers
  ... start with set $E := \{a \in A \mid \varphi_1(a) = \varphi_2(a)\}$ then ...



- Preimages

Some Limits exist

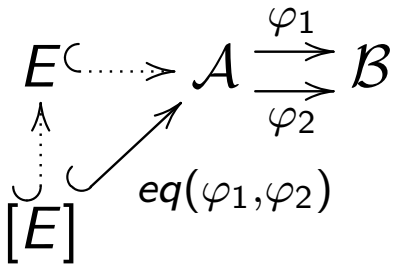
- Equalizers
  ... start with set $E := \{a \in A \mid \varphi_1(a) = \varphi_2(a)\}$ then ...



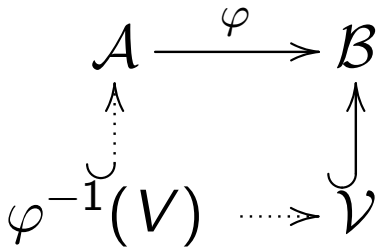
- Preimages

Some Limits exist

- Equalizers
  ... start with set $E := \{a \in A \mid \varphi_1(a) = \varphi_2(a)\}$ then ...

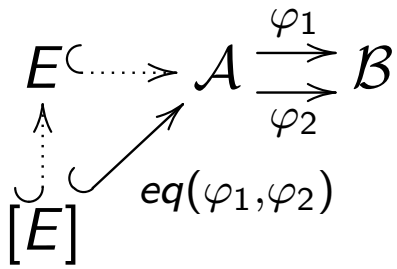


- Preimages

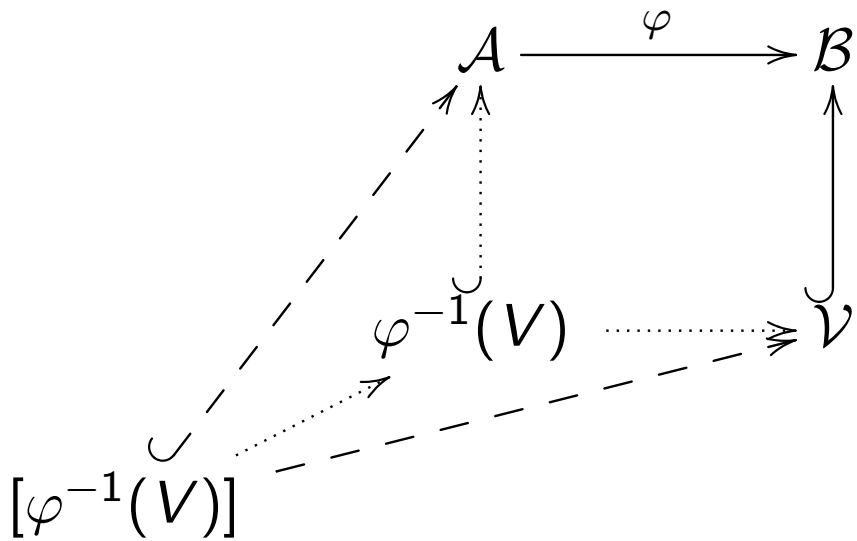


Some Limits exist

- Equalizers
  ... start with set $E := \{a \in A \mid \varphi_1(a) = \varphi_2(a)\}$ then ...



- Preimages



Products

- Products

$\mathcal{A} =$ 

$\mathcal{B} =$ 

- $\mathcal{A} \times \mathcal{B} \cong \emptyset$
- $\mathcal{A} \times \mathcal{A} \cong \mathcal{A}$
- $\mathcal{B} \times \mathcal{B}$ infinite in $Set_{\mathbb{P}_\omega}$
- $\mathcal{B} \times \mathcal{B}$ *not* in $Set_{\mathbb{P}}$.

- Terminal object
  - = product of empty family
- $D \times (-)^{\Sigma}$ has terminal $D^{\Sigma^\star}$
- no terminal object for $\mathbb{P}(-)$

Products

- Products

$\mathcal{A} =$ 

$\mathcal{B} =$ 

- $\mathcal{A} \times \mathcal{B} \cong \emptyset$
- $\mathcal{A} \times \mathcal{A} \cong \mathcal{A}$
- $\mathcal{B} \times \mathcal{B}$ infinite in $Set_{\mathbb{P}_\omega}$
- $\mathcal{B} \times \mathcal{B}$ *not* in $Set_{\mathbb{P}}$.

- Terminal object
  - = product of empty family
- $D \times (-)^\Sigma$ has terminal $D^{\Sigma^\star}$
- no terminal object for $\mathbb{P}(-)$

Products

- Products

$\mathcal{A} =$ 

$\mathcal{B} =$ 

- $\mathcal{A} \times \mathcal{B} \cong \emptyset$
- $\mathcal{A} \times \mathcal{A} \cong \mathcal{A}$
- $\mathcal{B} \times \mathcal{B}$ infinite in $Set_{\mathbb{P}_\omega}$
- $\mathcal{B} \times \mathcal{B}$ *not* in $Set_{\mathbb{P}}$.

- Terminal object
  - = product of empty family
- $D \times (-)^\Sigma$ has terminal $D^{\Sigma^\star}$
- no terminal object for $\mathbb{P}(-)$

Products

- Products

$\mathcal{A} =$ 

$\mathcal{B} =$ 

- $\mathcal{A} \times \mathcal{B} \cong \emptyset$
- $\mathcal{A} \times \mathcal{A} \cong \mathcal{A}$
- $\mathcal{B} \times \mathcal{B}$ infinite in $Set_{\mathbb{P}_\omega}$
- $\mathcal{B} \times \mathcal{B}$ *not* in $Set_{\mathbb{P}}$.

- Terminal object
  - = product of empty family
- $D \times (-)^\Sigma$ has terminal $D^{\Sigma^\star}$
- no terminal object for $\mathbb{P}(-)$

Products

- Products

$\mathcal{A} =$ 

$\mathcal{B} =$ 

- $\mathcal{A} \times \mathcal{B} \cong \emptyset$
- $\mathcal{A} \times \mathcal{A} \cong \mathcal{A}$
- $\mathcal{B} \times \mathcal{B}$ infinite in $Set_{\mathbb{P}_\omega}$
- $\mathcal{B} \times \mathcal{B}$ *not* in $Set_{\mathbb{P}}$.

- Terminal object
  - = product of empty family
- $D \times (-)^{\Sigma}$ has terminal $D^{\Sigma^\star}$
- no terminal object for $\mathbb{P}(-)$

Products

- Products

$\mathcal{A} =$ 

$\mathcal{B} =$ 

- $\mathcal{A} \times \mathcal{B} \cong \emptyset$
- $\mathcal{A} \times \mathcal{A} \cong \mathcal{A}$
- $\mathcal{B} \times \mathcal{B}$ infinite in $Set_{\mathbb{P}_\omega}$
- $\mathcal{B} \times \mathcal{B}$ *not* in $Set_{\mathbb{P}}$.

- Terminal object
  - = product of empty family
- $D \times (-)^\Sigma$ has terminal $D^{\Sigma^\star}$
- no terminal object for $\mathbb{P}(-)$

Products

- Products

$\mathcal{A} =$ 

$\mathcal{B} =$ 

- $\mathcal{A} \times \mathcal{B} \cong \emptyset$
- $\mathcal{A} \times \mathcal{A} \cong \mathcal{A}$
- $\mathcal{B} \times \mathcal{B}$ infinite in $Set_{\mathbb{P}_\omega}$
- $\mathcal{B} \times \mathcal{B}$ *not* in $Set_{\mathbb{P}}$.

- Terminal object
  - $=$ product of empty family
- $D \times (-)^{\Sigma}$ has terminal $D^{\Sigma^\star}$
- no terminal object for $\mathbb{P}(-)$

Products

- Products

$\mathcal{A} =$ 

$\mathcal{B} =$ 

- $\mathcal{A} \times \mathcal{B} \cong \emptyset$
- $\mathcal{A} \times \mathcal{A} \cong \mathcal{A}$
- $\mathcal{B} \times \mathcal{B}$ infinite in $Set_{\mathbb{P}_\omega}$
- $\mathcal{B} \times \mathcal{B}$ *not* in $Set_{\mathbb{P}}$.

- Terminal object
  - $=$ product of empty family
- $D \times (-)^{\Sigma}$ has terminal $D^{\Sigma^\star}$
- no terminal object for $\mathbb{P}(-)$

Lambek's lemma

- The structure map $\alpha : T \to F(T)$ on a terminal coalgebra is an isomorphism.

$$\begin{array}{c} T \\ {\scriptstyle\alpha}\downarrow \\ F(T) \end{array}$$

- There is *no* terminal Kripke structure
  - no set X satisfies $X \cong \mathbb{P}(X)$
- ...but there *is* a terminal *image finite* Kripke Structure
  - replace $\mathbb{P}(-)$ by $\mathbb{P}_\omega(-)$

Lambek's lemma

- The structure map $\alpha : T \to F(T)$ on a terminal coalgebra is an isomorphism.

$$\begin{array}{ccc} T & \xrightarrow{\alpha} & F(T) \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle F\alpha} \\ F(T) & \xrightarrow[F\alpha]{} & F(F(T)) \end{array}$$

- There is *no* terminal Kripke structure
  - no set X satisfies $X \cong \mathbb{P}(X)$
- ...but there *is* a terminal *image finite* Kripke Structure
  - replace $\mathbb{P}(-)$ by $\mathbb{P}_\omega(-)$

Lambek's lemma

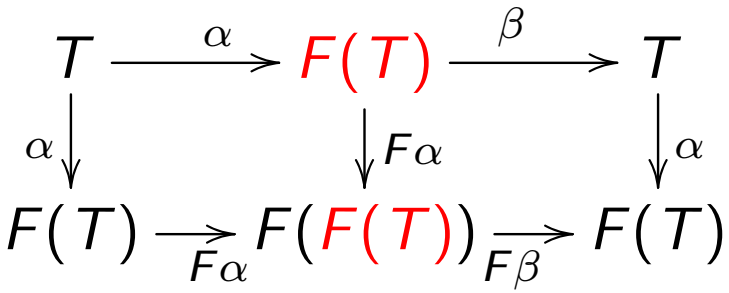
- The structure map $\alpha : T \to F(T)$ on a terminal coalgebra is an isomorphism.

$$\begin{array}{ccc} T & \xrightarrow{\alpha} & F(T) \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle F\alpha} \\ F(T) & \xrightarrow[F\alpha]{} & F(F(T)) \end{array}$$

- There is *no* terminal Kripke structure
  - no set X satisfies $X \cong \mathbb{P}(X)$
- ...but there *is* a terminal *image finite* Kripke Structure
  - replace $\mathbb{P}(-)$ by $\mathbb{P}_\omega(-)$

Lambek's lemma

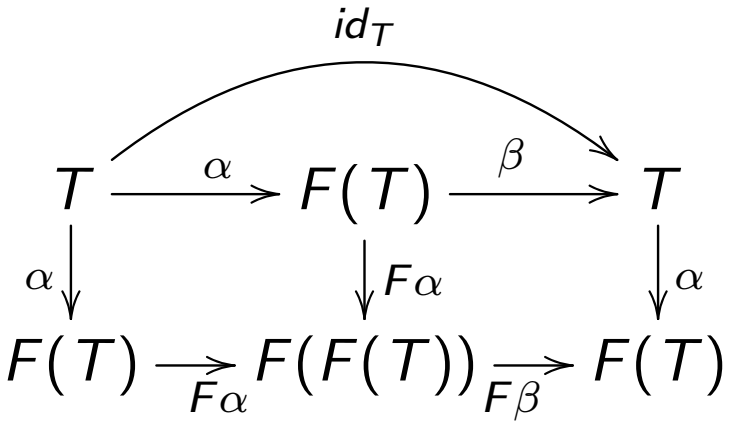
- The structure map $\alpha : T \to F(T)$ on a terminal coalgebra is an isomorphism.



- There is *no* terminal Kripke structure
  - no set X satisfies $X \cong \mathbb{P}(X)$
- ...but there *is* a terminal *image finite* Kripke Structure
  - replace $\mathbb{P}(-)$ by $\mathbb{P}_\omega(-)$

Lambek's lemma

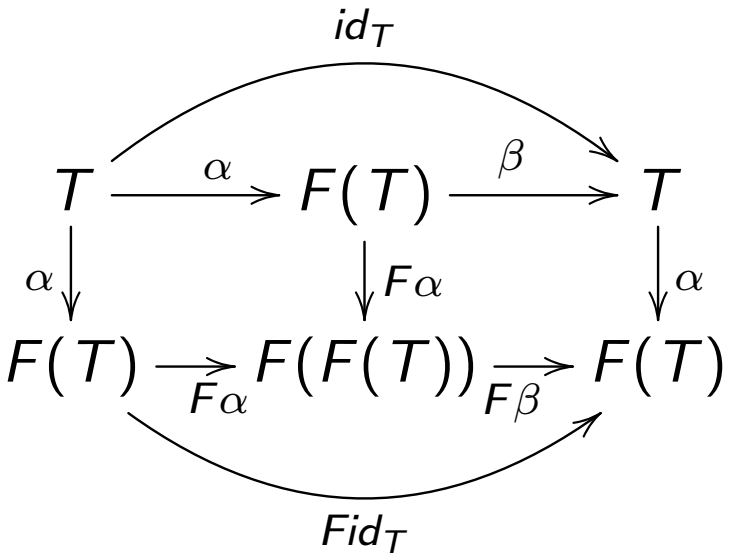
- The structure map $\alpha : T \to F(T)$ on a terminal coalgebra is an isomorphism.



- There is *no* terminal Kripke structure
  - no set X satisfies $X \cong \mathbb{P}(X)$
- ...but there *is* a terminal *image finite* Kripke Structure
  - replace $\mathbb{P}(-)$ by $\mathbb{P}_\omega(-)$

Lambek's lemma

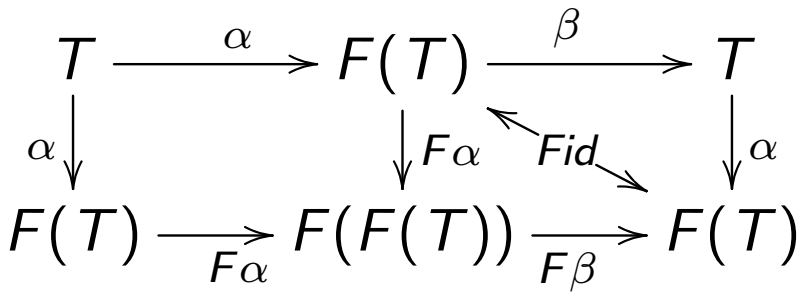
- The structure map $\alpha : T \to F(T)$ on a terminal coalgebra is an isomorphism.



- There is *no* terminal Kripke structure
  - no set X satisfies $X \cong \mathbb{P}(X)$
- ...but there *is* a terminal *image finite* Kripke Structure
  - replace $\mathbb{P}(-)$ by $\mathbb{P}_\omega(-)$

Lambek's lemma

- The structure map $\alpha : T \to F(T)$ on a terminal coalgebra is an isomorphism.

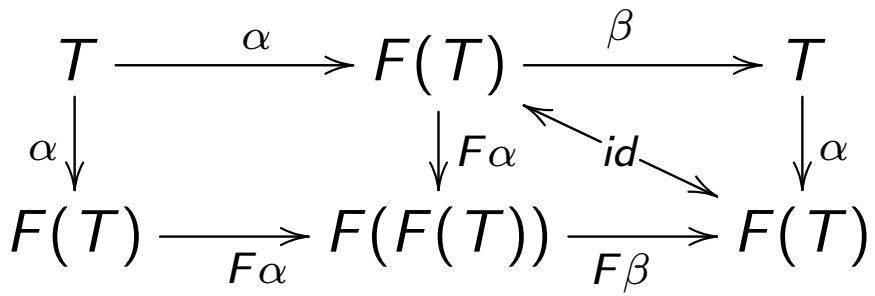


$$\beta \circ \alpha = id_T$$

- There is *no* terminal Kripke structure
  - no set X satisfies $X \cong \mathbb{P}(X)$
- ...but there *is* a terminal *image finite* Kripke Structure
  - replace $\mathbb{P}(-)$ by $\mathbb{P}_\omega(-)$

Lambek's lemma

- The structure map $\alpha : T \to F(T)$ on a terminal coalgebra is an isomorphism.

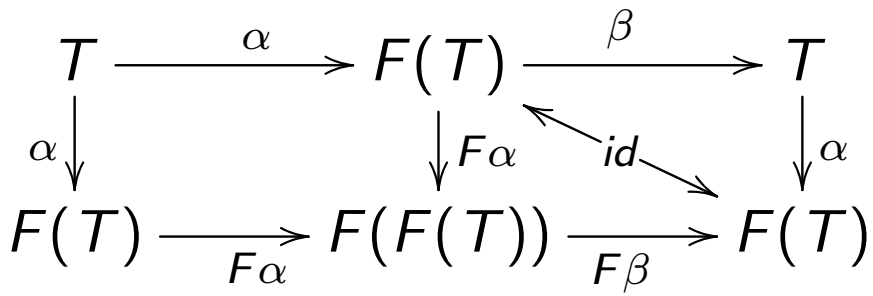


$$\alpha \circ \beta = id_{F(T)}$$

- There is *no* terminal Kripke structure
  - no set X satisfies $X \cong \mathbb{P}(X)$
- ...but there *is* a terminal *image finite* Kripke Structure
  - replace $\mathbb{P}(-)$ by $\mathbb{P}_\omega(-)$

Lambek's lemma

- The structure map $\alpha : T \to F(T)$ on a terminal coalgebra is an isomorphism.

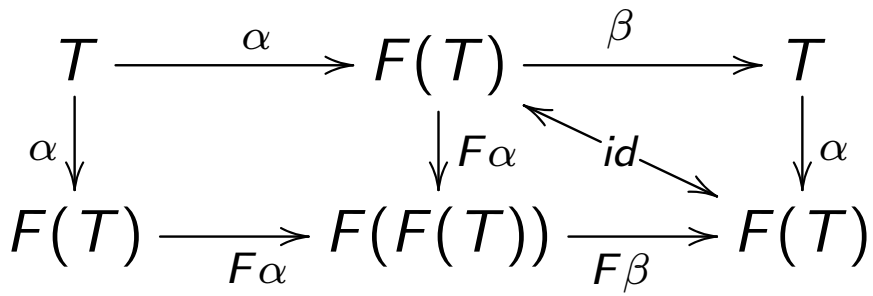


$$\alpha \circ \beta = id_{F(T)}$$

- There is *no* terminal Kripke structure
  - no set X satisfies $X \cong \mathbb{P}(X)$
- ...but there *is* a terminal *image finite* Kripke Structure
  - replace $\mathbb{P}(-)$ by $\mathbb{P}_\omega(-)$

Lambek's lemma

- The structure map $\alpha : T \to F(T)$ on a terminal coalgebra is an isomorphism.



$$\alpha \circ \beta = id_{F(T)}$$

- There is *no* terminal Kripke structure
  - no set X satisfies $X \cong \mathbb{P}(X)$
- ...but there *is* a terminal *image finite* Kripke Structure
  - replace $\mathbb{P}(-)$ by $\mathbb{P}_\omega(-)$

Bounded functors

F is $\kappa - \textit{bounded}$: $\iff$ $\forall u \in F(X).\exists_{|X_0|<\kappa}.u \in F(X_0)$

- Every functor F has a $\kappa - \textit{bounded}$ subfunctor:

Examples

- $\mathbb{P}_\omega(X)$ - all finite subsets
- $X^*_\omega = X^*$
- X^ω_ω - streams with finitely different elements
- X^3_3 - $\{(x_1, x_2, x_3) \mid |\{x_1, x_2, x_3\}| < 3\}$

Bounded functors

F is $\kappa - \mathit{bounded}$: $\iff$ $\forall u \in F(X).\exists_{|X_0|<\kappa}.u \in F(X_0)$

- Every functor F has a $\kappa - \mathit{bounded}$ subfunctor:

$$F_\kappa(X) := \bigcup_{X_0 \subseteq_\kappa X} F(X_0)$$

Examples

- $\mathbb{P}_\omega(X)$ - all finite subsets
- $X^*_\omega = X^*$
- X^ω_ω - streams with finitely different elements
- X^3_3 - $\{(x_1, x_2, x_3) \mid |\{x_1, x_2, x_3\}| < 3\}$

Bounded functors

F is $\kappa - \textit{bounded}$: $\iff \forall u \in F(X).\exists_{|X_0|<\kappa}.u \in F(X_0)$

- Every functor F has a $\kappa - \textit{bounded}$ subfunctor:

$$F_\kappa(X) := \bigcup_{\substack{\kappa_0<\kappa \\ f:\kappa_0\to X}} Ff[F(\kappa_0)]$$

Examples

- $\mathbb{P}_\omega(X)$ - all finite subsets
- $X^*_\omega = X^*$
- X^ω_ω - streams with finitely different elements
- X^3_3 - $\{(x_1, x_2, x_3) \mid |\{x_1, x_2, x_3\}| < 3\}$

Bounded functors

The following are equivalent

1. F is $\kappa-$bounded
2. Each F-coalgebra has a subcoalgebra of size $\leq \kappa$
3. $<_\kappa$-small subsets extend to $\leq_\kappa$-small sub-coalgebras
4. Each F-coalgebra is a *subdirect sum* of size κ subcoalgebras
5. There is surjective nat. transformation $\mu : D \times (-)^\kappa \twoheadrightarrow F(-)$

Bounded functors

The following are equivalent

1. F is $\kappa-$bounded
2. Each F-coalgebra has a subcoalgebra of size $\leq \kappa$
3. $<_\kappa$-small subsets extend to $\leq_\kappa$-small sub-coalgebras
4. Each F-coalgebra is a *subdirect sum* of size κ subcoalgebras
5. There is surjective nat. transformation $\mu : D \times (-)^\kappa \twoheadrightarrow F(-)$

$$X^\omega \longrightarrow F(X)^{F(\omega)}$$

Bounded functors

The following are equivalent

1. F is $\kappa-$bounded
2. Each F-coalgebra has a subcoalgebra of size $\leq \kappa$
3. $<_\kappa$-small subsets extend to $\leq_\kappa$-small sub-coalgebras
4. Each F-coalgebra is a *subdirect sum* of size κ subcoalgebras
5. There is surjective nat. transformation $\mu : D \times (-)^\kappa \twoheadrightarrow F(-)$

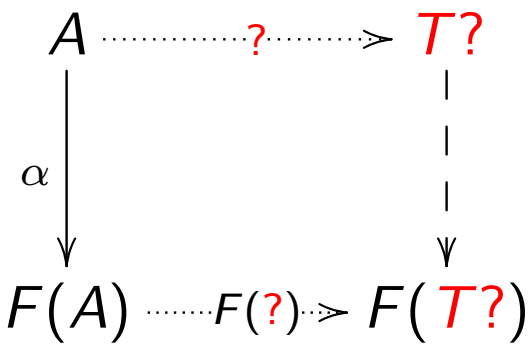
$$F(\omega) \times X^\omega \xrightarrow{\mu_X} \!\!\!\!\rightarrow F(X)$$

Existence of terminal algebras

Theorem

If F is $\kappa-$ bounded, then there is a terminal F-coalgebra

- ... choose $D = F(\kappa)$ and $\Sigma = \kappa$

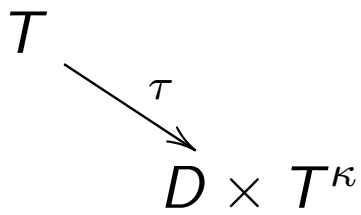


Existence of terminal algebras

Theorem

If F is $\kappa-$ bounded, then there is a terminal F-coalgebra

- ... choose $D = F(\kappa)$ and $\Sigma = \kappa$

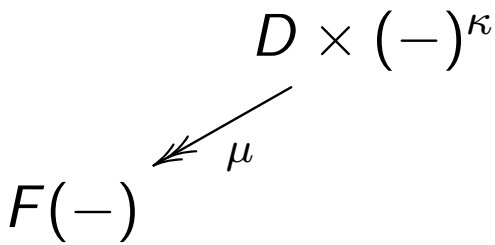


Existence of terminal algebras

Theorem

If F is $\kappa -$ bounded, then there is a terminal F-coalgebra

- ... choose $D = F(\kappa)$ and $\Sigma = \kappa$

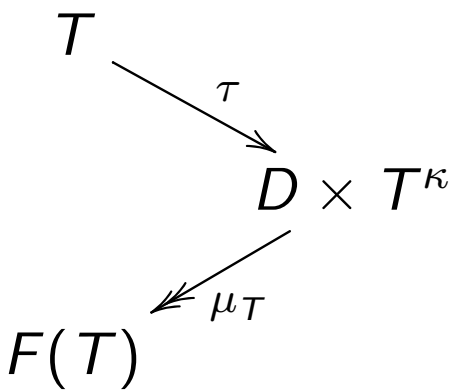


Existence of terminal algebras

Theorem

If F is $\kappa-$ bounded, then there is a terminal F-coalgebra

- ... choose $D = F(\kappa)$ and $\Sigma = \kappa$

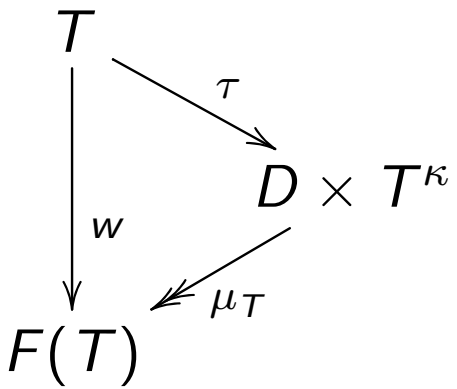


Existence of terminal algebras

Theorem

If F is $\kappa-$ bounded, then there is a terminal F-coalgebra

- ... choose $D = F(\kappa)$ and $\Sigma = \kappa$

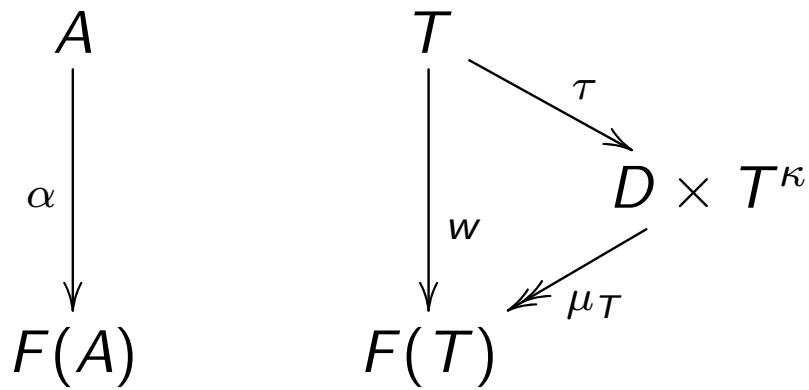


Existence of terminal algebras

Theorem

If F is $\kappa-$ bounded, then there is a terminal F-coalgebra

- ... choose $D = F(\kappa)$ and $\Sigma = \kappa$

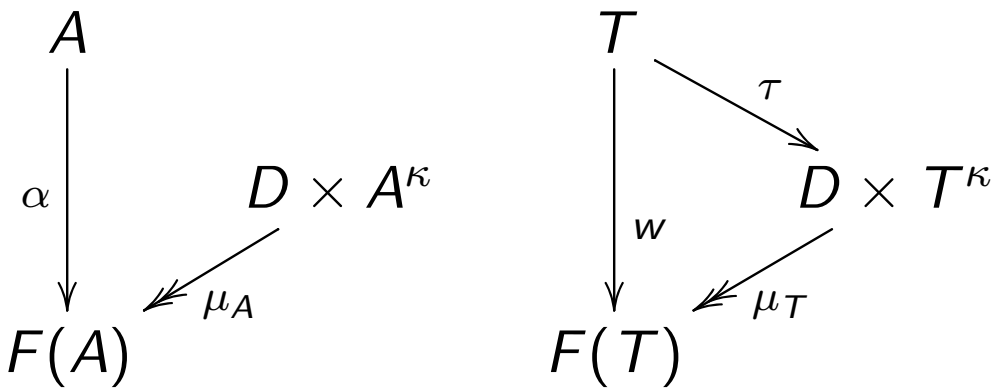


Existence of terminal algebras

Theorem

If F is $\kappa-$ bounded, then there is a terminal F-coalgebra

- ... choose $D = F(\kappa)$ and $\Sigma = \kappa$

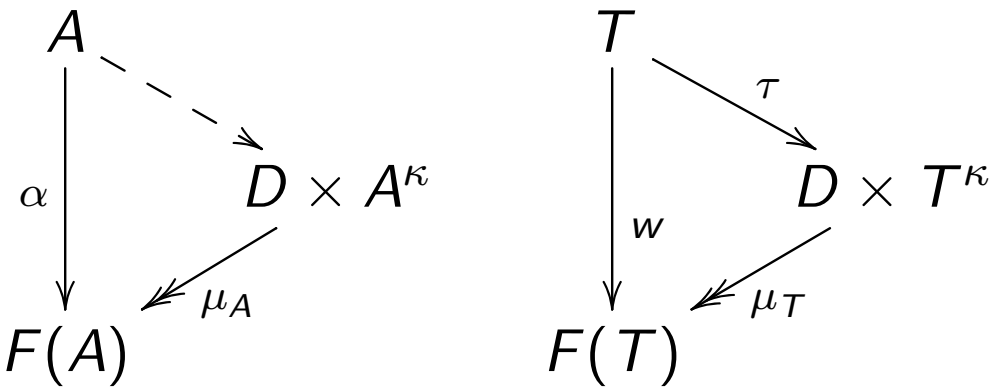


Existence of terminal algebras

Theorem

If F is $\kappa-$ bounded, then there is a terminal F-coalgebra

- ... choose $D = F(\kappa)$ and $\Sigma = \kappa$

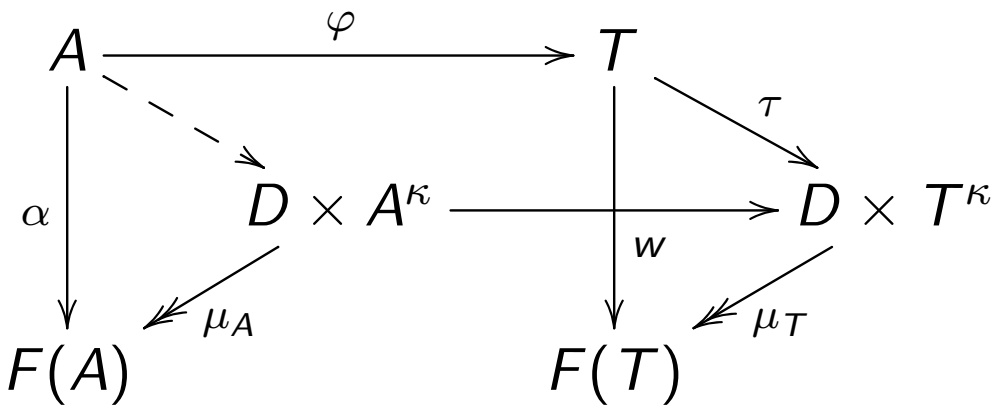


Existence of terminal algebras

Theorem

If F is $\kappa-$ bounded, then there is a terminal F-coalgebra

- ... choose $D = F(\kappa)$ and $\Sigma = \kappa$

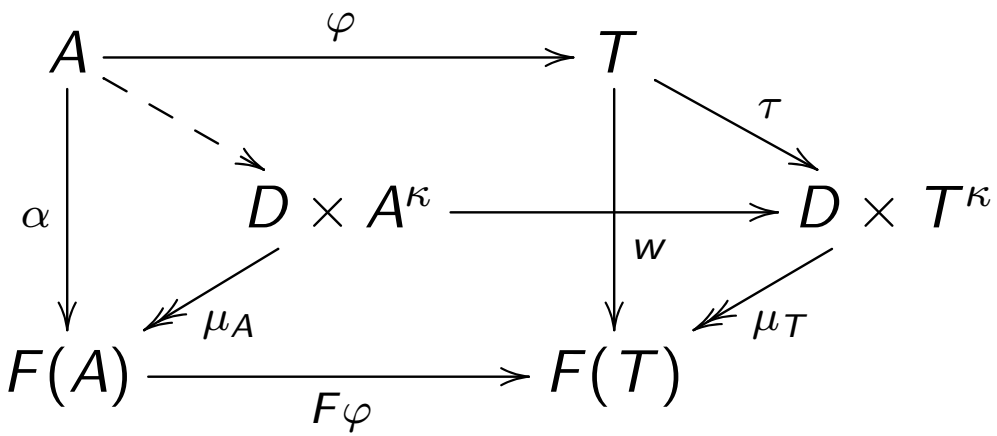


Existence of terminal algebras

Theorem

If F is κ – bounded, then there is a terminal F-coalgebra

- ... choose $D = F(\kappa)$ and $\Sigma = \kappa$

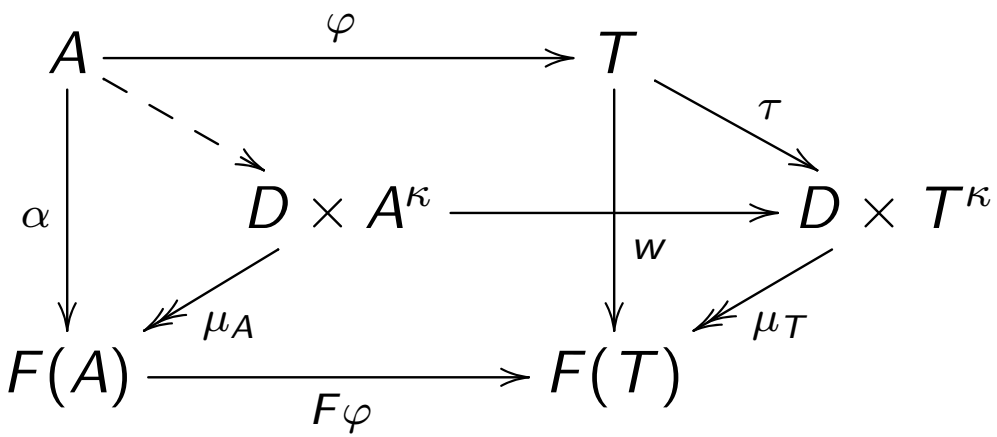


Existence of terminal algebras

Theorem

If F is κ – bounded, then there is a terminal F-coalgebra

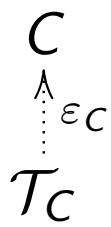
- ... choose $D = F(\kappa)$ and $\Sigma = \kappa$



$\mathcal{T} = (T, w)$ is weakly terminal, so $\mathcal{T}/\nabla$ is terminal

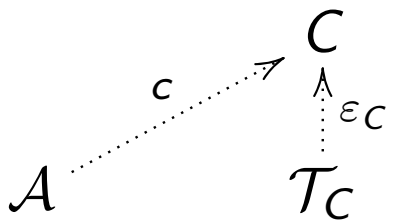
Cofree coalgebras

- Let C be any set (of colors) ...
- $\mathcal{T}_C = (T_C, \alpha)$ with "coloring" map $\varepsilon_C : T_C \to C$ is called *cofree* if
  - for each coalgebra $\mathcal{A} = (A, \alpha)$ with coloring $c : A \to C$
  - there is exactly one homomorphism $\varphi : \mathcal{A} \to \mathcal{T}_C$ with $c = \varepsilon_C \circ \varphi$.



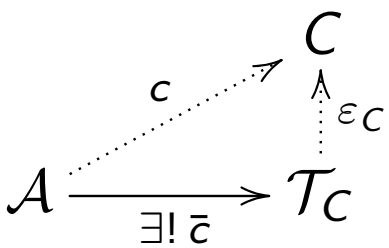
Cofree coalgebras

- Let C be any set (of colors) ...
- $\mathcal{T}_C = (T_C, \alpha)$ with "coloring" map $\varepsilon_C : T_C \to C$ is called *cofree* if
  - for each coalgebra $\mathcal{A} = (A, \alpha)$ with coloring $c : A \to C$
  - there is exactly one homomorphism $\varphi : \mathcal{A} \to \mathcal{T}_C$ with $c = \varepsilon_C \circ \varphi$.



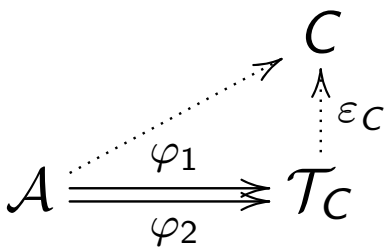
Cofree coalgebras

- Let C be any set (of colors) ...
- $\mathcal{T}_C = (T_C, \alpha)$ with "coloring" map $\varepsilon_C : T_C \to C$ is called *cofree* if
  - for each coalgebra $\mathcal{A} = (A, \alpha)$ with coloring $c : A \to C$
  - there is exactly one homomorphism $\varphi : \mathcal{A} \to \mathcal{T}_C$ with $c = \varepsilon_C \circ \varphi$.



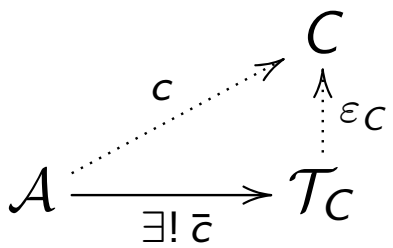
Cofree coalgebras

- ε_C separates homomorphisms
- Cofree F-coalgebras are terminal $C \times F(-)$-coalgebras
- If F is bounded then cofree coalgebras exist for each set C.



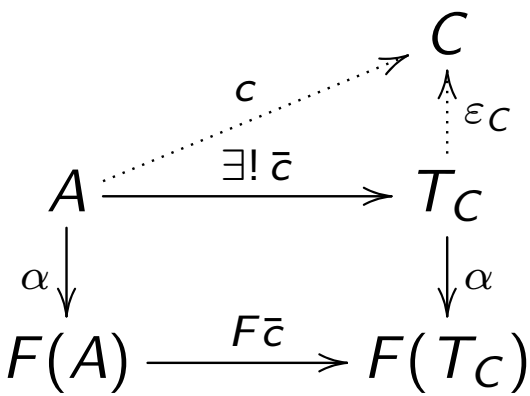
Cofree coalgebras

- ε_C separates homomorphisms
- Cofree F-coalgebras are terminal $C \times F(-)$-coalgebras
- If F is bounded then cofree coalgebras exist for each set C.



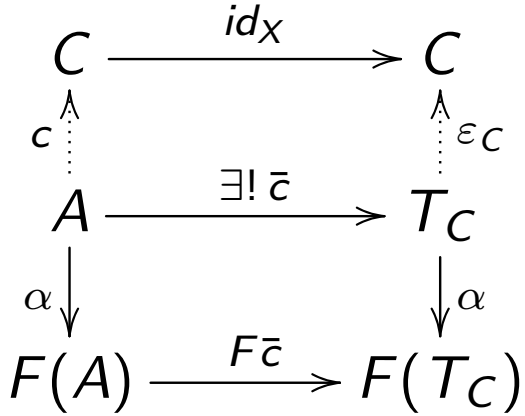
Cofree coalgebras

- ε_C separates homomorphisms
- Cofree F-coalgebras are terminal $C \times F(-)$-coalgebras
- If F is bounded then cofree coalgebras exist for each set C.



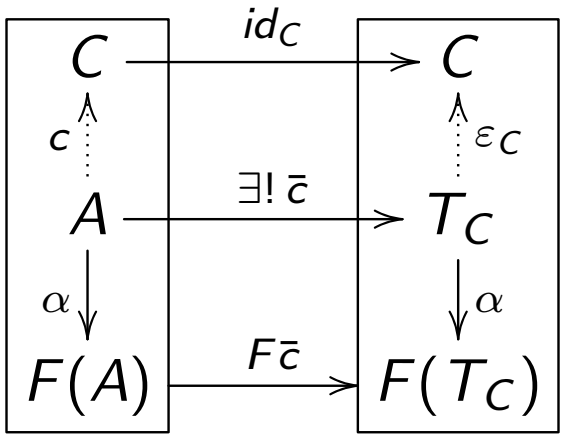
Cofree coalgebras

- ε_C separates homomorphisms
- Cofree F-coalgebras are terminal $C \times F(-)$-coalgebras
- If F is bounded then cofree coalgebras exist for each set C.



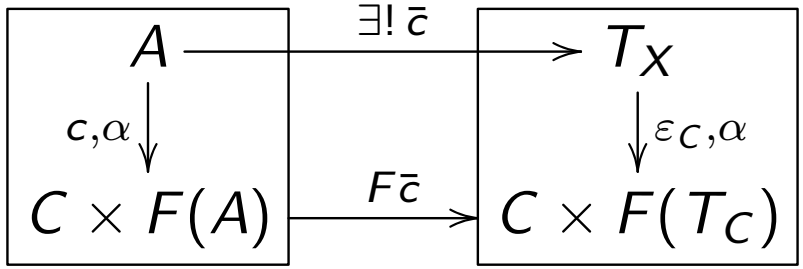
Cofree coalgebras

- ε_C separates homomorphisms
- Cofree F-coalgebras are terminal $C \times F(-)$-coalgebras
- If F is bounded then cofree coalgebras exist for each set C.



Cofree coalgebras

- ε_C separates homomorphisms
- Cofree F-coalgebras are terminal $C \times F(-)$-coalgebras
- If F is bounded then cofree coalgebras exist for each set C.



Coequations

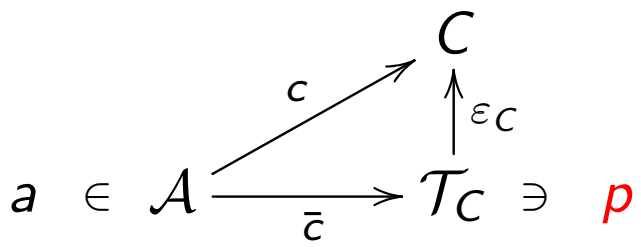
Definition

A *coequation* p is a "forbidden" element of a cofree coalgebra.

Coequations

Definition

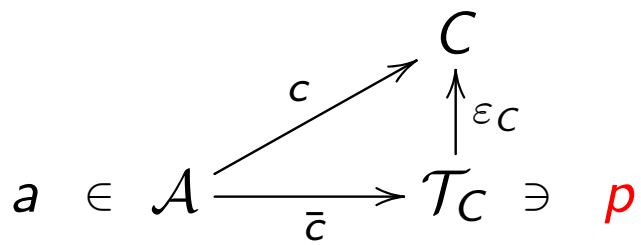
A *coequation* p is a "forbidden" element of a cofree coalgebra.



Coequations

Definition

A *coequation* p is a "forbidden" element of a cofree coalgebra.



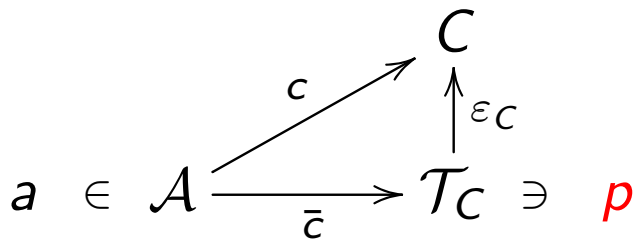
$$\mathcal{A}, a \models_c p \quad :\Longleftrightarrow \quad \bar{c}(a) \neq p$$

$$\mathcal{A} \models p \quad :\Longleftrightarrow \quad \forall a \in A. \forall c : A \to C. \mathcal{A}, a \models p$$

Coequations

Definition

A *coequation* p is a "forbidden" element of a cofree coalgebra.



$$\mathcal{A}, a \models_c p \quad :\Longleftrightarrow \quad \bar{c}(a) \neq p$$

$$\mathcal{A} \models p \quad :\Longleftrightarrow \quad \forall a \in A. \forall c : A \to C. \mathcal{A}, a \models p$$

If $\mathcal{A}, \mathcal{A}_i \models p$ then

- if $\mathcal{U} \leq \mathcal{A}$ then $\mathcal{U} \models p$.
- if $\varphi : \mathcal{A} \twoheadrightarrow \mathcal{B}$ then $\mathcal{B} \models p$
- $\Sigma \mathcal{A}_i \models p$

Covarieties

- $\mathfrak{K}$ a class of F-coalgebras.
  - $S(\mathfrak{K})$ all subcoalgebras
  - $H(\mathfrak{K})$ all homomorphic images
  - $\Sigma(\mathfrak{K})$ all sums
- $\mathfrak{K}$ a *is a covariety if* $\mathfrak{K}$ *is closed under* S, H, Σ
- if and only if $\mathfrak{K} = SH\Sigma(\mathfrak{K})$
- E a set of coequations, $Mod(E) := \{\mathcal{A} \mid \forall e \in P.\mathcal{A} \models p\}$ is covariety
- $\mathfrak{K}$ a class of coalgebras, $Eq_C(\mathfrak{K}) := \{p \in \mathcal{T}_C \mid \forall \mathcal{A} \in \mathfrak{K}.\mathcal{A} \models p\}$

Co-Birkhoff

Theorem

F bounded (by κ). Covarieties are the model classes for coequations

Co-Birkhoff

Theorem

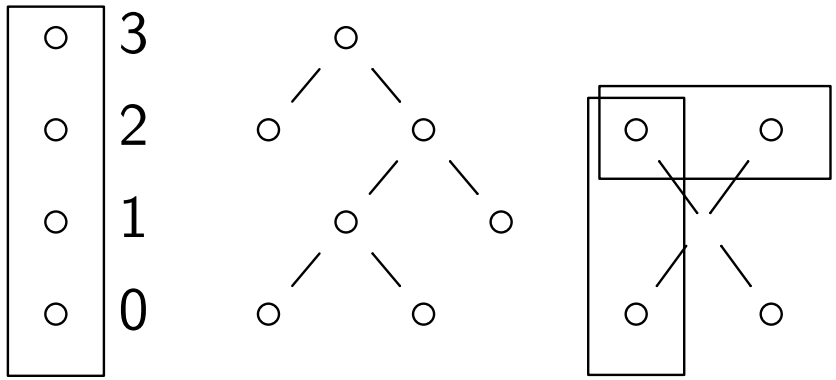
F bounded (by κ). Covarieties are the model classes for coequations

Proof.

- Clearly: $\mathfrak{K} \subseteq Mod(Eq_C(\mathfrak{K}))$ for $|C| \geq \kappa$.
- Conversely: Let $\mathcal{A} \in Mod(Eq_C(\mathfrak{K}))$:
- First reduce to case: $|\mathcal{A}| \leq \kappa$:
  - $\mathcal{A}$ is subdirect sum of $\mathcal{B}_i \leq \mathcal{A}$ with $|\mathcal{B}_i| \leq \kappa$.
  - each $\mathcal{B}_i \in Mod(Eq_C(\mathfrak{K}))$
  - if $\mathcal{B}_i \in \mathfrak{K}$, then $\mathcal{A} \in \mathfrak{K}$
- Now may assume $|\mathcal{A}| \leq \kappa$.
  - choose injective coloring $c : A \to C$, then $\mathcal{A} \cong \bar{c}(A) \leq \mathcal{T}_C$.
  - for each $u \in \bar{c}(A)$ there is $\mathcal{A}_u \in \mathfrak{K}$ and coloring c_u with $u \in \bar{c_u}(\mathcal{A}_u) \leq \mathcal{T}_C$
  - $\mathcal{A} \in SH\Sigma(\{\mathcal{A}_u \mid u \in \bar{c}(A)\}) \subseteq \mathfrak{K}$

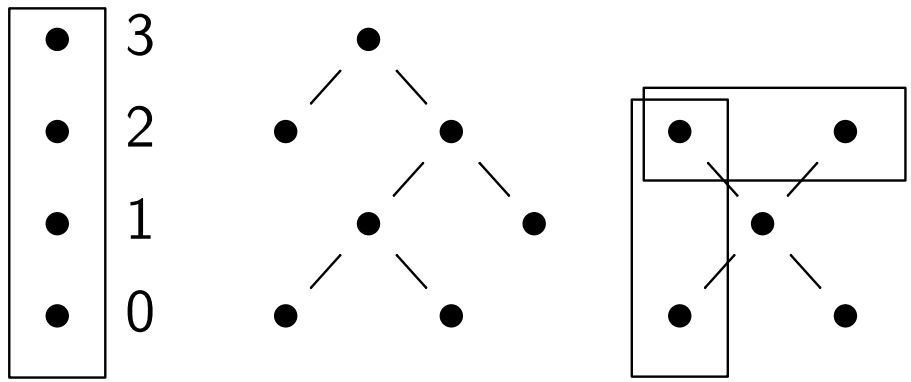
Useful Intuition for functors

- containers
  - $F(1)$: shapes
  - $F(2)$: 0-1-patterns
  - ...
  - $F(X)$: X -pattern.
- $F\sigma$ is substitution with σ
  - $\sigma : x_0, x_2 \mapsto \bullet, x_1 \mapsto \circ :$



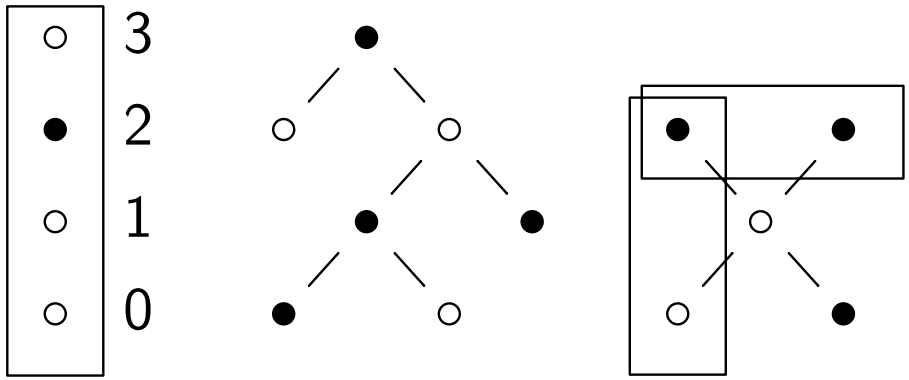
Useful Intuition for functors

- containers
  - $F(1)$: shapes
  - $F(2)$: 0-1-patterns
  - ...
  - $F(X)$: X -pattern.
- $F\sigma$ is substitution with σ
  - $\sigma : x_0, x_2 \mapsto \bullet, x_1 \mapsto \circ$:



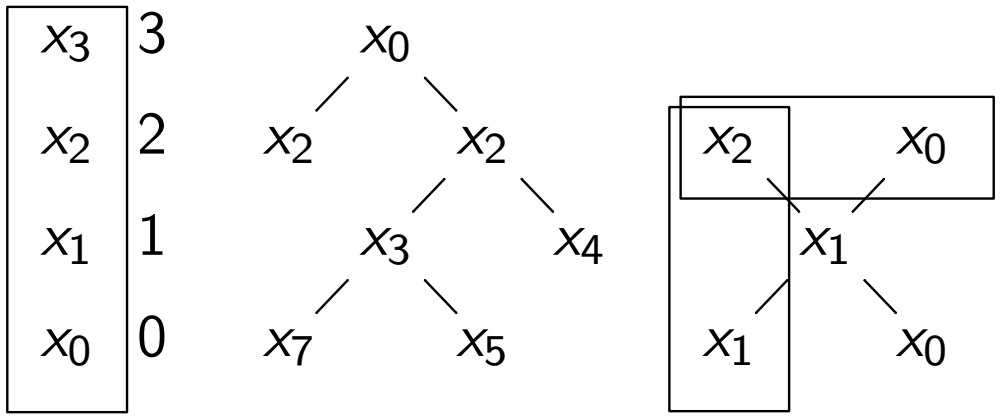
Useful Intuition for functors

- containers
  - $F(1)$: shapes
  - $F(2)$: 0-1-patterns
  - ...
  - $F(X)$: X -pattern.
- $F\sigma$ is substitution with σ
  - $\sigma : x_0, x_2 \mapsto \bullet, x_1 \mapsto \circ$:



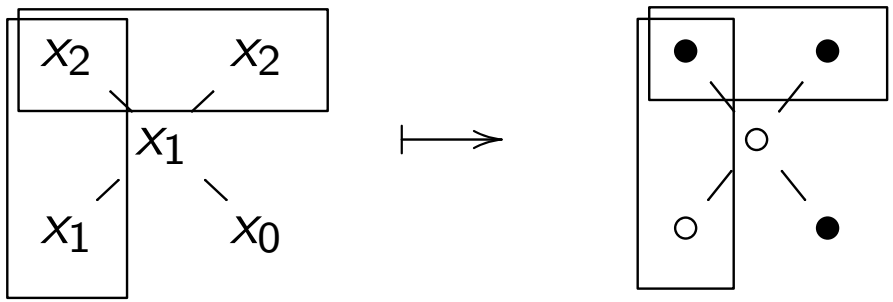
Useful Intuition for functors

- containers
  - $F(1)$: shapes
  - $F(2)$: 0-1-patterns
  - ...
  - $F(X)$: X -pattern.
- $F\sigma$ is substitution with σ
  - $\sigma : x_0, x_2 \mapsto \bullet, x_1 \mapsto \circ$:



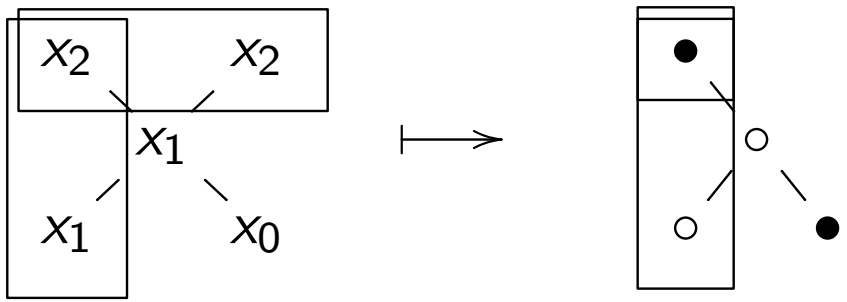
Useful Intuition for functors

- containers
  - $F(1)$: shapes
  - $F(2)$: 0-1-patterns
  - ...
  - $F(X)$: X -pattern.
- $F\sigma$ is substitution with σ
  - $\sigma : x_0, x_2 \mapsto \bullet,\ x_1 \mapsto \circ :$



Useful Intuition for functors

- containers
  - $F(1)$: shapes
  - $F(2)$: 0-1-patterns
  - ...
  - $F(X)$: X -pattern.
- $F\sigma$ is substitution with σ
  - $\sigma : x_0, x_2 \mapsto \bullet,\ x_1 \mapsto \circ :$



Patterns and substitutions

$X = \{x_1, ..., x_\kappa\}$ a set of variables

- $p \in F(X)$ is called $\kappa - pattern$
- Functor on maps is substitution

$$p \in F(X)$$

Patterns and substitutions

$X = \{x_1, ..., x_\kappa\}$ a set of variables

- $p \in F(X)$ is called $\kappa - pattern$
- Functor on maps is substitution

$$p(x_1, ..., x_n, ...) \quad = \quad p \quad \in \quad F(X)$$

Patterns and substitutions

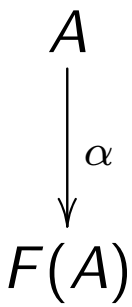
$X = \{x_1, ..., x_\kappa\}$ a set of variables

- $p \in F(X)$ is called $\kappa - \textit{pattern}$
- Functor on maps is substitution

$$\begin{array}{ccccccc} p(x_1, ..., x_n, ...) & = & p & \in & F(X) & X \\ \Big\downarrow F\sigma & & & & \Big\downarrow F\sigma & \Big\downarrow \sigma \\ p(x_1\sigma, ..., x_n\sigma, ...) & = & q & \in & F(Y) & Y \end{array}$$

Modalities

- Use formulae ϕ as test: $[\![\phi]\!] : A \to 2$
- F $[\![\phi]\!] : F(A) \to F(2)$ produces $0-1-$patterns
- judge $a \in A$ by the $0-1-$pattern of $\alpha(a)$
- For $\Box \subseteq F(2)$:

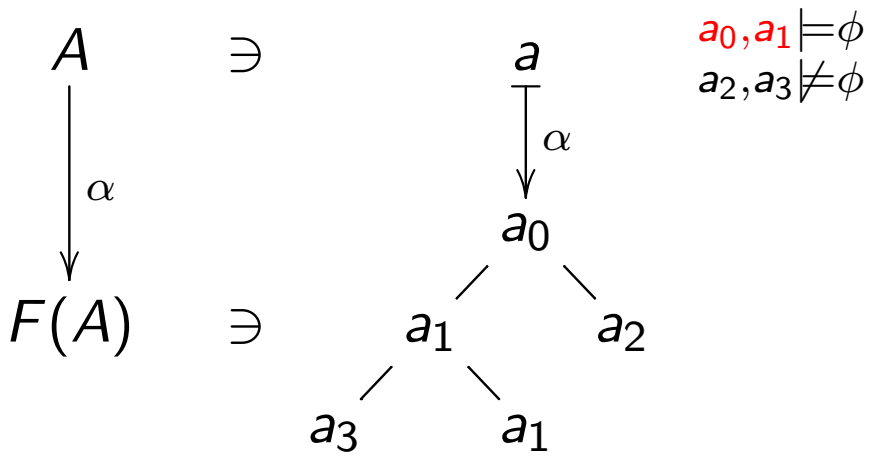


$a_0, a_1 \models \phi$
$a_2, a_3 \not\models \phi$

$$a \models \Box\phi :\iff (F\ [\![\phi]\!])(\alpha(a)) \in \Box$$

Modalities

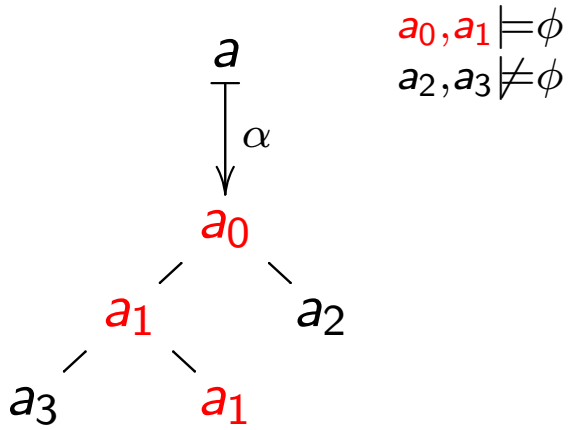
- Use formulae ϕ as test: $[\![\phi]\!] : A \to 2$
- F $[\![\phi]\!] : F(A) \to F(2)$ produces $0-1-$patterns
- judge $a \in A$ by the $0-1-$pattern of $\alpha(a)$
- For $\Box \subseteq F(2)$:



$$a \models \Box\phi :\iff (F\ [\![\phi]\!])(\alpha(a)) \in \Box$$

Modalities

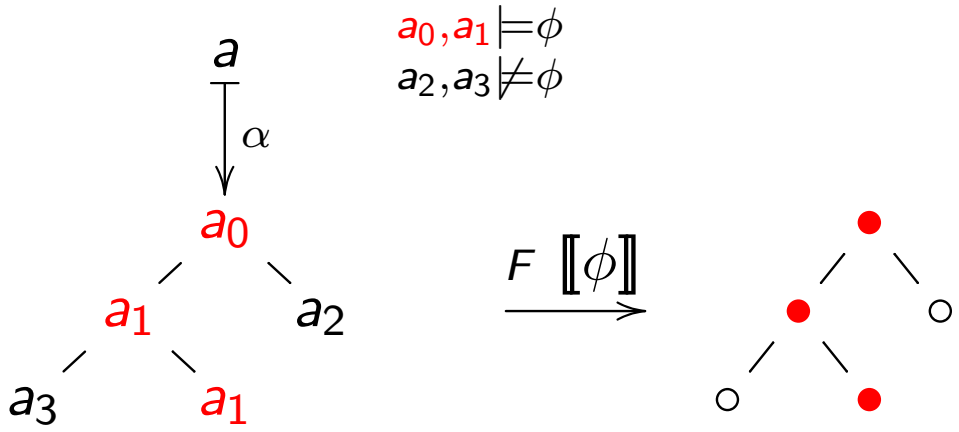
- Use formulae ϕ as test: $[\![\phi]\!] : A \to 2$
- F $[\![\phi]\!] : F(A) \to F(2)$ produces $0-1-$patterns
- judge $a \in A$ by the $0-1-$pattern of $\alpha(a)$
- For $\Box \subseteq F(2)$:



$$a \models \Box\phi :\iff (F\ [\![\phi]\!])(\alpha(a)) \in \Box$$

Modalities

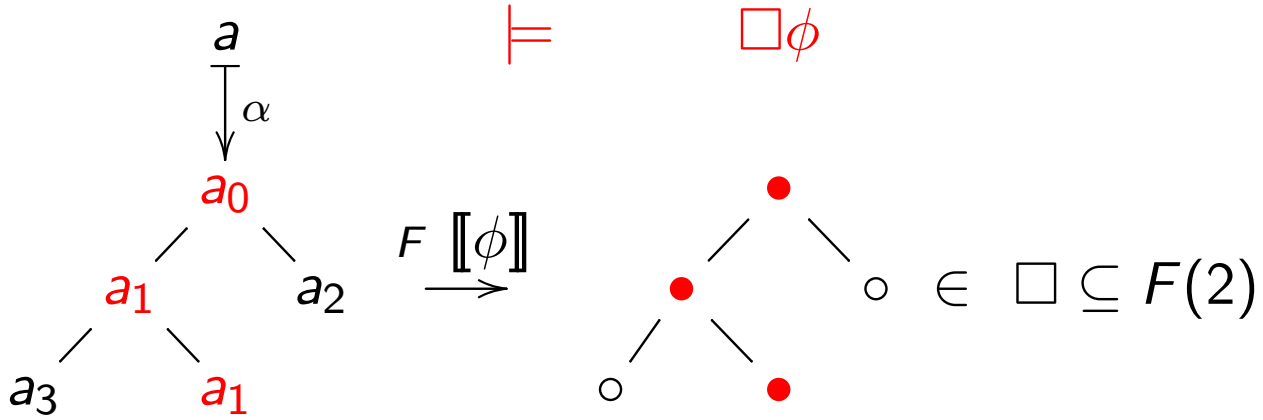
- Use formulae ϕ as test: $[\![\phi]\!] : A \to 2$
- F $[\![\phi]\!] : F(A) \to F(2)$ produces $0-1-$patterns
- judge $a \in A$ by the $0-1-$pattern of $\alpha(a)$
- For $\Box \subseteq F(2)$:



$$a \models \Box\phi :\iff (F\ [\![\phi]\!])(\alpha(a)) \in \Box$$

Modalities

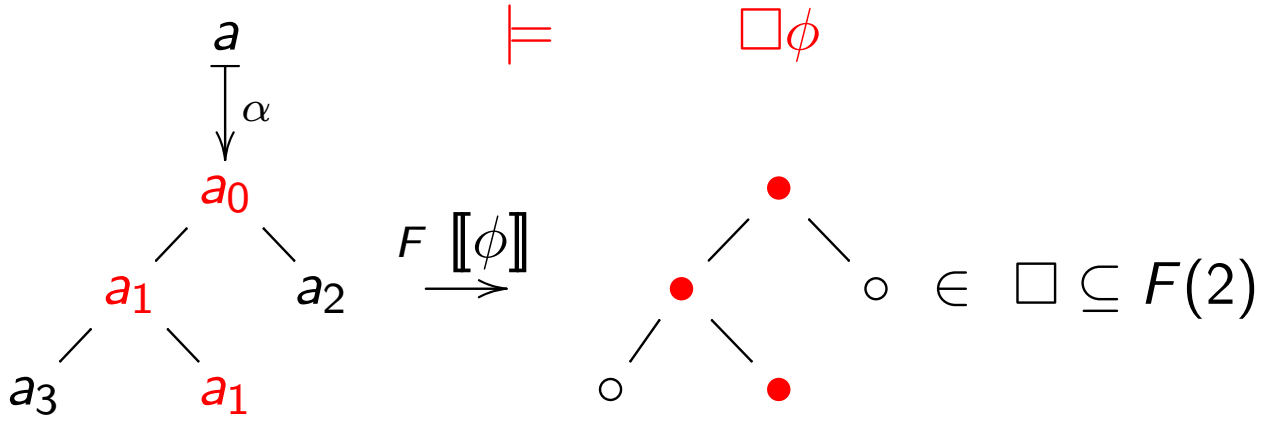
- Use formulae ϕ as test: $[\![\phi]\!] : A \to 2$
- F $[\![\phi]\!] : F(A) \to F(2)$ produces $0-1-$patterns
- judge $a \in A$ by the $0-1-$pattern of $\alpha(a)$
- For $\Box \subseteq F(2)$:



$$a \models \Box\phi :\iff (F\,[\![\phi]\!])(\alpha(a)) \in \Box$$

Modalities

- Use formulae ϕ as test: $[\![\phi]\!] : A \to 2$
- F $[\![\phi]\!] : F(A) \to F(2)$ produces $0-1-$patterns
- judge $a \in A$ by the $0-1-$pattern of $\alpha(a)$
- For $\Box \subseteq F(2)$:



$$a \models \Box\phi :\iff (F\ [\![\phi]\!])(\alpha(a)) \in \Box$$

Modal Logic

- Formulae
  - pick $\Box_i \subseteq F(2)$

$$\begin{array}{rcl} \phi & :: & \mathit{true} \mid \neg\phi \mid \bigwedge_{i\in I} \phi_i \\ & \mid & \Box\phi \quad \text{for } \Box \subseteq F(2) \end{array}$$

- Semantics
  - $x \models \bigwedge_{i\in I} \phi_i :\Longleftrightarrow \forall i \in I. x \models \phi_i$
  - $x \models \Box\phi :\Longleftrightarrow F\,[\![\phi]\!](\alpha(x)) \in \Box$

Modal Logic

- Formulae
  - pick $\Box_i \subseteq F(2)$

$$\phi \quad :: \quad true \mid \neg\phi \mid \bigwedge_{i \in I} \phi_i$$
$$\mid \quad \Box\phi \quad \text{for } \Box \subseteq F(2)$$

$$\overset{x}{\dashrightarrow} A \xrightarrow{[\![\phi]\!]} 2$$

- Semantics
  - $x \models \bigwedge_{i \in I} \phi_i \iff \forall i \in I. x \models \phi_i$
  - $x \models \Box\phi \iff F\,[\![\phi]\!](\alpha(x)) \in \Box$

Modal Logic

- Formulae
  - pick $\Box_i \subseteq F(2)$

$$\phi \quad :: \quad true \mid \neg\phi \mid \bigwedge_{i \in I} \phi_i$$
$$\mid \quad \Box\phi \quad \text{for } \Box \subseteq F(2)$$

$$\overset{x}{\dashrightarrow} A \xrightarrow{[\![\phi]\!]} 2$$

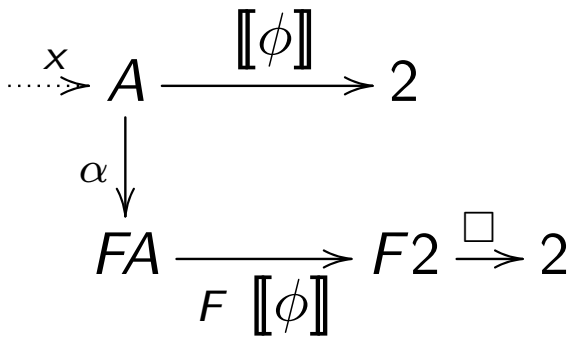
$$F2 \xrightarrow{\Box} 2$$

- Semantics
  - $x \models \bigwedge_{i \in I} \phi_i \iff \forall i \in I. x \models \phi_i$
  - $x \models \Box\phi \iff F\,[\![\phi]\!](\alpha(x)) \in \Box$

Modal Logic

- Formulae
  - pick $\Box_i \subseteq F(2)$

$$\begin{array}{rcl} \phi & :: & \mathit{true} \mid \neg\phi \mid \bigwedge_{i\in I} \phi_i \\ & \mid & \Box\phi \quad \text{for } \Box \subseteq F(2) \end{array}$$



- Semantics
  - $x \models \bigwedge_{i\in I} \phi_i \iff \forall i \in I. x \models \phi_i$
  - $x \models \Box\phi \iff F\,[\![\phi]\!](\alpha(x)) \in \Box$

Stability: $\nabla \subseteq \approx$

- $\models$ is homomorphism stable

$$\varphi : \mathcal{A} \to \mathcal{B} \implies (\forall a \in A.a \models \phi \iff \varphi(a) \models \phi)$$

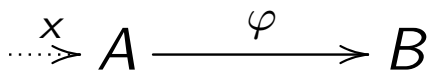
- Proof by formula induction

Stability: $\nabla \subseteq \approx$

- $\models$ is homomorphism stable

$$\varphi : \mathcal{A} \to \mathcal{B} \implies (\forall a \in A.a \models \phi \iff \varphi(a) \models \phi)$$

- Proof by formula induction

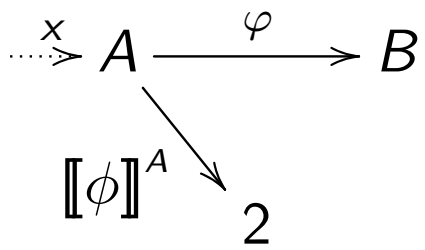


Stability: $\nabla \subseteq \approx$

- $\models$ is homomorphism stable

$$\varphi : \mathcal{A} \to \mathcal{B} \implies (\forall a \in A.a \models \phi \iff \varphi(a) \models \phi)$$

- Proof by formula induction

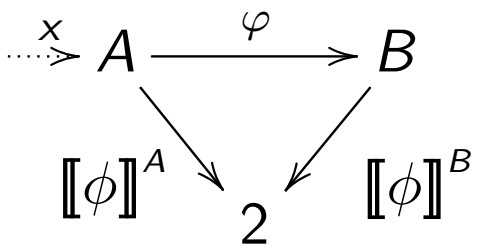


Stability: $\nabla \subseteq \approx$

- $\models$ is homomorphism stable

$$\varphi : \mathcal{A} \to \mathcal{B} \implies (\forall a \in A.a \models \phi \iff \varphi(a) \models \phi)$$

- Proof by formula induction

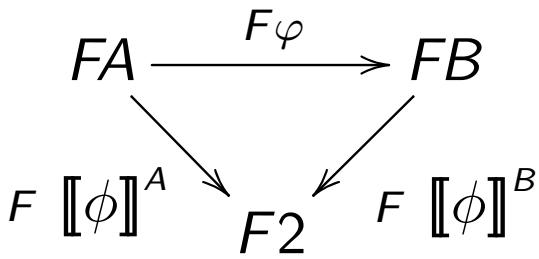


Stability: $\nabla \subseteq \approx$

- $\models$ is homomorphism stable

$$\varphi : \mathcal{A} \to \mathcal{B} \implies (\forall a \in A.a \models \phi \iff \varphi(a) \models \phi)$$

- Proof by formula induction

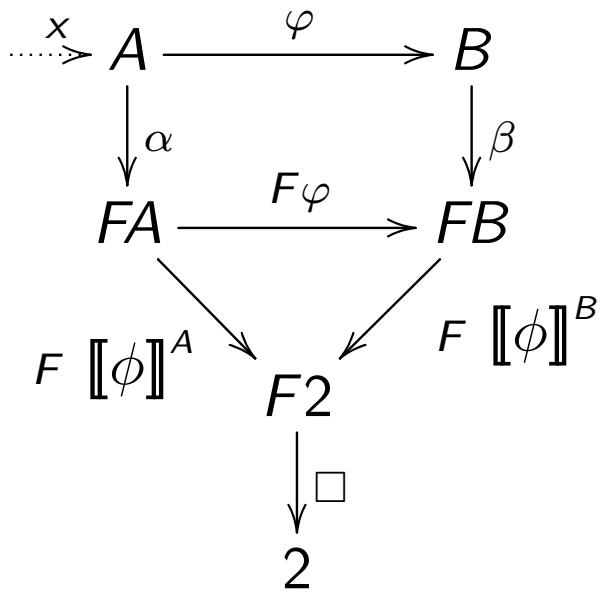


Stability: $\nabla \subseteq \approx$

- $\models$ is homomorphism stable

$$\varphi : \mathcal{A} \to \mathcal{B} \implies (\forall a \in A.a \models \phi \iff \varphi(a) \models \phi)$$

- Proof by formula induction



Completeness: $\approx \subseteq \nabla$

- F is $2-$*separable* if $\{Ff \mid f : X \to 2\}$ separates points

1. For $a \not\approx b$ find $\phi_{a,b}$ with
   - $a \models \phi_{a,b}$ but $b \not\models \phi_{a,b}$.
2. $\phi_a = \bigwedge_{a \not\approx b} \phi_{a,b}$ defines a:
   - $x \approx a \iff x \models \phi_a$.
3. For $f : A/\approx \to 2$
   - $\phi_f := \bigvee\{\phi_a \mid f \circ \pi_\approx(a) = 1\}$

$$X \xrightarrow{f} 2$$

$$\overset{a}{\underset{b}{\rightrightarrows}} F(X) \xrightarrow{Ff} F(2)$$

Completeness: $\approx \subseteq \nabla$

- F is $2-$*separable* if
  $\{Ff \mid f : X \to 2\}$ separates points

1. For $a \not\approx b$ find $\phi_{a,b}$ with
   - $a \models \phi_{a,b}$ but $b \not\models \phi_{a,b}$.
2. $\phi_a = \bigwedge_{a \not\approx b} \phi_{a,b}$ defines a:
   - $x \approx a \iff x \models \phi_a$.
3. For $f : A/\approx \to 2$
   - $\phi_f := \vee\{\phi_a \mid f \circ \pi_\approx(a) = 1\}$

$$A \xrightarrow{\pi_\approx} A/\approx$$

Completeness: $\approx \subseteq \nabla$

- F is $2-$*separable* if
  $\{Ff \mid f : X \to 2\}$ separates points

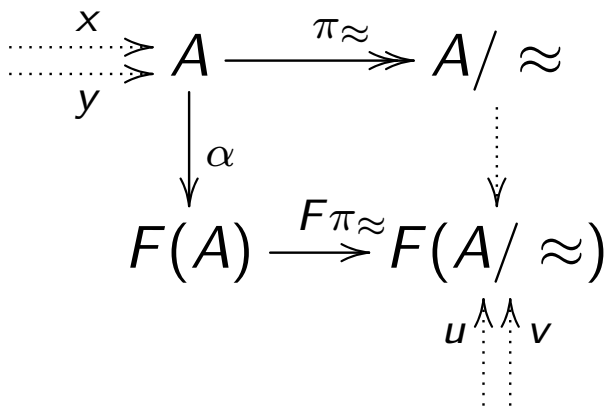
1. For $a \not\approx b$ find $\phi_{a,b}$ with
   - $a \models \phi_{a,b}$ but $b \not\models \phi_{a,b}$.
2. $\phi_a = \bigwedge_{a \not\approx b} \phi_{a,b}$ defines a:
   - $x \approx a \iff x \models \phi_a$.
3. For $f : A/\approx \to 2$
   - $\phi_f := \bigvee\{\phi_a \mid f \circ \pi_\approx(a) = 1\}$

$$\begin{array}{ccc} A & \xrightarrow{\pi_\approx} & A/\approx \\ \downarrow{\scriptstyle \alpha} & & \vdots \\ F(A) & \xrightarrow{F\pi_\approx} & F(A/\approx) \end{array}$$

Completeness: $\approx \subseteq \nabla$

- F is $2-$*separable* if $\{Ff \mid f : X \to 2\}$ separates points

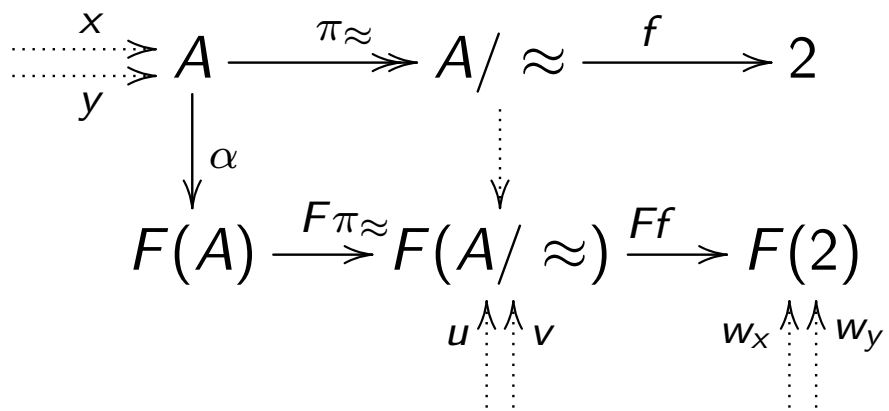
1. For $a \not\approx b$ find $\phi_{a,b}$ with
   - $a \models \phi_{a,b}$ but $b \not\models \phi_{a,b}$.
2. $\phi_a = \bigwedge_{a \not\approx b} \phi_{a,b}$ defines a:
   - $x \approx a \iff x \models \phi_a$.
3. For $f : A/\approx \to 2$
   - $\phi_f := \bigvee\{\phi_a \mid f \circ \pi_\approx(a) = 1\}$



Completeness: $\approx \subseteq \nabla$

- F is $2-$*separable* if $\{Ff \mid f : X \to 2\}$ separates points

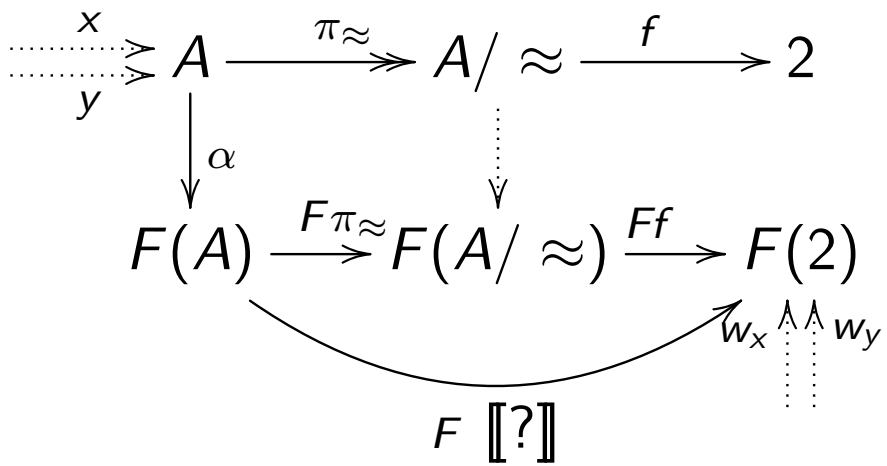
1. For $a \not\approx b$ find $\phi_{a,b}$ with
   - $a \models \phi_{a,b}$ but $b \not\models \phi_{a,b}$.
2. $\phi_a = \bigwedge_{a \not\approx b} \phi_{a,b}$ defines a:
   - $x \approx a \iff x \models \phi_a$.
3. For $f : A/\approx \to 2$
   - $\phi_f := \bigvee\{\phi_a \mid f \circ \pi_\approx(a) = 1\}$



Completeness: $\approx \subseteq \nabla$

- F is $2-$*separable* if
  $\{Ff \mid f : X \to 2\}$ separates points

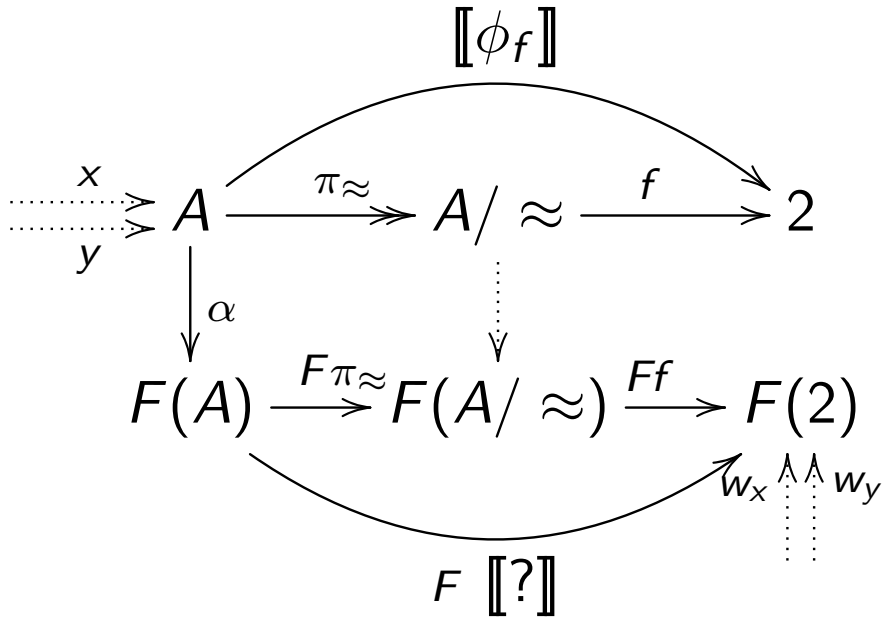
1. For $a \not\approx b$ find $\phi_{a,b}$ with
   - $a \models \phi_{a,b}$ but $b \not\models \phi_{a,b}$.
2. $\phi_a = \bigwedge_{a \not\approx b} \phi_{a,b}$ defines a:
   - $x \approx a \iff x \models \phi_a$.
3. For $f : A/\approx \to 2$
   - $\phi_f := \vee\{\phi_a \mid f \circ \pi_\approx(a) = 1\}$



Completeness: $\approx \subseteq \nabla$

- F is $2-$*separable* if $\{Ff \mid f : X \to 2\}$ separates points

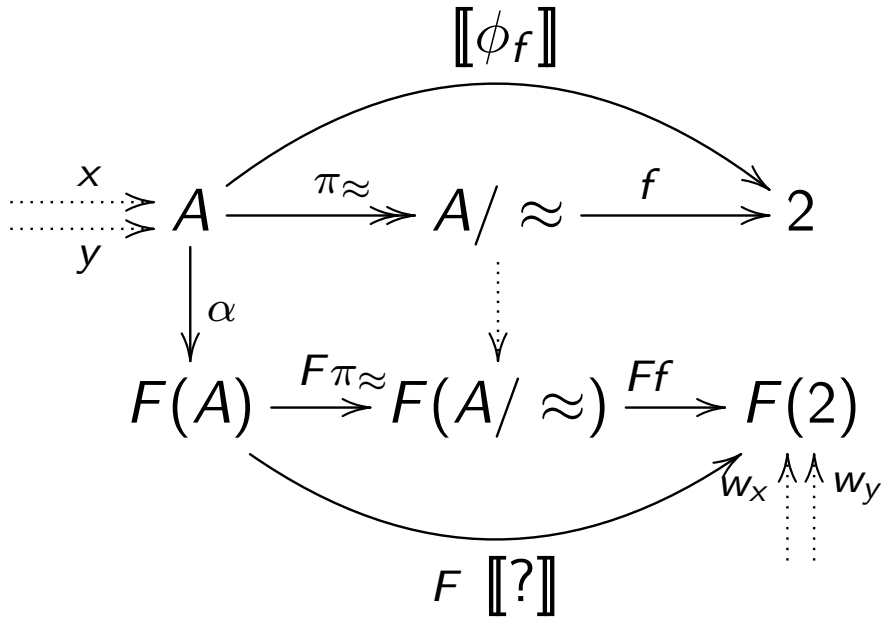
1. For $a \not\approx b$ find $\phi_{a,b}$ with
   - $a \models \phi_{a,b}$ but $b \not\models \phi_{a,b}$.
2. $\phi_a = \bigwedge_{a \not\approx b} \phi_{a,b}$ defines a:
   - $x \approx a \iff x \models \phi_a$.
3. For $f : A/\approx \to 2$
   - $\phi_f := \bigvee\{\phi_a \mid f \circ \pi_\approx(a) = 1\}$



Completeness: $\approx \subseteq \nabla$

- F is $2-$*separable* if $\{Ff \mid f : X \to 2\}$ separates points

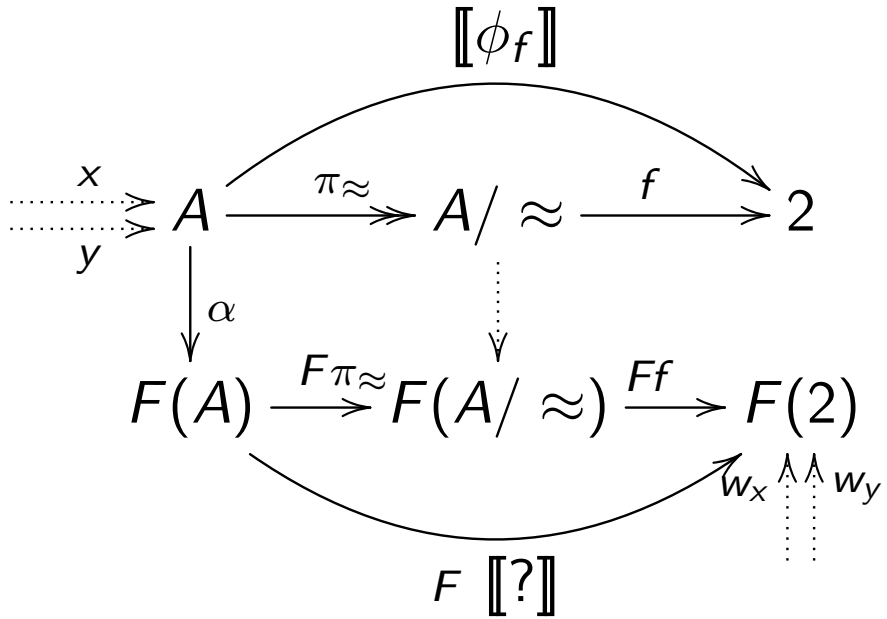
1. For $a \not\approx b$ find $\phi_{a,b}$ with
   - $a \models \phi_{a,b}$ but $b \not\models \phi_{a,b}$.
2. $\phi_a = \bigwedge_{a \not\approx b} \phi_{a,b}$ defines a:
   - $x \approx a \iff x \models \phi_a$.
3. For $f : A/\approx \to 2$
   - $\phi_f := \bigvee\{\phi_a \mid f \circ \pi_\approx(a) = 1\}$



Completeness: $\approx \subseteq \nabla$

- F is $2-$*separable* if $\{Ff \mid f : X \to 2\}$ separates points

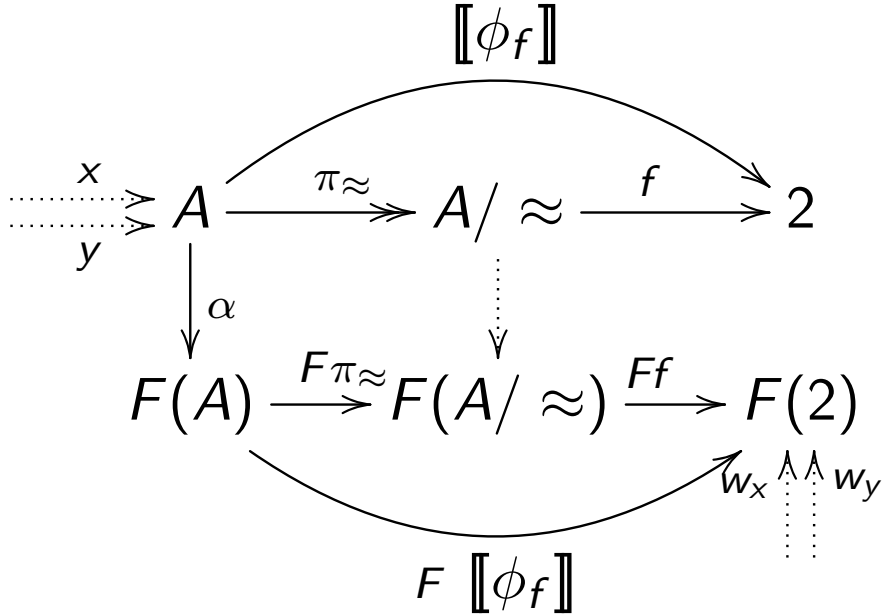
1. For $a \not\approx b$ find $\phi_{a,b}$ with
   - $a \models \phi_{a,b}$ but $b \not\models \phi_{a,b}$.
2. $\phi_a = \bigwedge_{a \not\approx b} \phi_{a,b}$ defines a:
   - $x \approx a \iff x \models \phi_a$.
3. For $f : A/\approx \to 2$
   - $\phi_f := \bigvee\{\phi_a \mid f \circ \pi_\approx(a) = 1\}$



Completeness: $\approx \subseteq \nabla$

- F is $2-$*separable* if $\{Ff \mid f : X \to 2\}$ separates points

1. For $a \not\approx b$ find $\phi_{a,b}$ with
   - $a \models \phi_{a,b}$ but $b \not\models \phi_{a,b}$.
2. $\phi_a = \bigwedge_{a \not\approx b} \phi_{a,b}$ defines a:
   - $x \approx a \iff x \models \phi_a$.
3. For $f : A/\approx \to 2$
   - $\phi_f := \vee\{\phi_a \mid f \circ \pi_\approx(a) = 1\}$



Example: Probabilistic systems

- $\mathcal{D}_\omega(X) =$ finitary prob. distr. on X
- Assumption satisfied:
  - $\{\mathcal{D}_\omega f \mid f : X \to 2\}$ separate points
- Formulae

$$\begin{array}{rcl} \phi & :: & [\bar{w}]\phi \text{ where } \bar{w} = [w, 1] \\ & \mid & \phi \wedge \phi \mid \neg\phi \mid \mathit{true} \end{array}$$

- Semantics
  - $x \models [\bar{w}]\phi \iff \sum\{p \mid x \xrightarrow{p} y \models \phi\} \geq w$

Example: Probabilistic systems

- $\mathcal{D}_\omega(X) =$ finitary prob. distr. on X
- Assumption satisfied:
  - $\{\mathcal{D}_\omega f \mid f : X \to 2\}$ separate points
- Formulae

$$\begin{array}{rcl} \phi & :: & [\bar{w}]\phi \text{ where } \bar{w} = [w, 1] \\ & | & \phi \wedge \phi \mid \neg\phi \mid \mathit{true} \end{array}$$

- Semantics
  - $x \models [\bar{w}]\phi \iff \sum\{p \mid x \xrightarrow{p} y \models \phi\} \geq w$

Example: Probabilistic systems

- $\mathcal{D}_\omega(X) =$ finitary prob. distr. on X
- Assumption satisfied:
  - $\{\mathcal{D}_\omega f \mid f : X \to 2\}$ separate points
- Formulae

$$\begin{array}{rcl} \phi & :: & [\bar{w}]\phi \text{ where } \bar{w} = [w, 1] \\ & \mid & \phi \wedge \phi \mid \neg\phi \mid \mathit{true} \end{array}$$

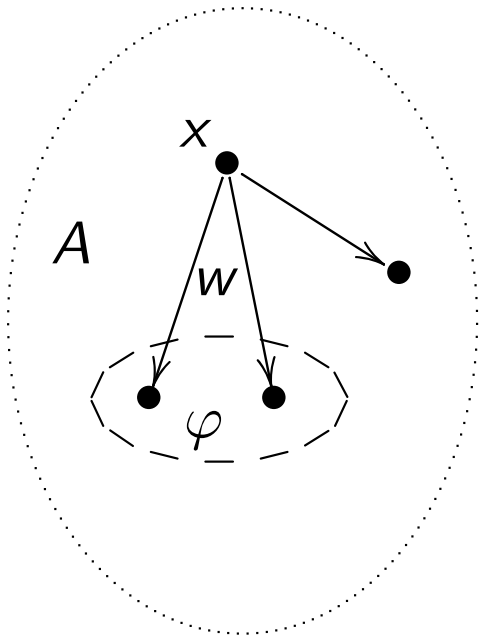
- Semantics
  - $x \models [\bar{w}]\phi \iff \sum\{p \mid x \xrightarrow{p} y \models \phi\} \geq w$

Example: Probabilistic systems

- $\mathcal{D}_\omega(X) =$ finitary prob. distr. on X
- Assumption satisfied:
  - $\{\mathcal{D}_\omega f \mid f : X \to 2\}$ separate points
- Formulae

$$\begin{array}{rcl} \phi & :: & [\bar{w}]\phi \text{ where } \bar{w} = [w, 1] \\ & \mid & \phi \wedge \phi \mid \neg\phi \mid \mathit{true} \end{array}$$

- Semantics
  - $x \models [\bar{w}]\phi \iff \sum\{p \mid x \xrightarrow{p} y \models \phi\} \geq w$



Towards Finitary Logic

- Formulae

$$\begin{aligned} \phi \quad :: \quad & \mathit{true} \mid \neg\phi \mid \bigwedge_{i \in I} \phi_i \\ \mid \quad & \Box\phi \quad \text{for } \Box \subseteq F(2) \end{aligned}$$

- Finite conjunctions

Towards Finitary Logic

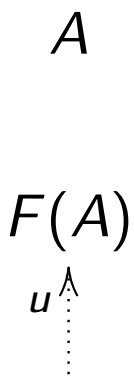
- Formulae
- Finite conjunctions

$$\begin{array}{rcl} \phi & :: & \mathit{true} \mid \neg\phi \mid \phi \wedge \phi \\ & \mid & \Box\phi \quad \text{for } \Box \subseteq F(2) \end{array}$$

Finitary completeness

- Assume F is $\omega - bounded$
  - $F(X) = \bigcup\{F[U]X \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_{\approx}) = 1\}$

Wrap up



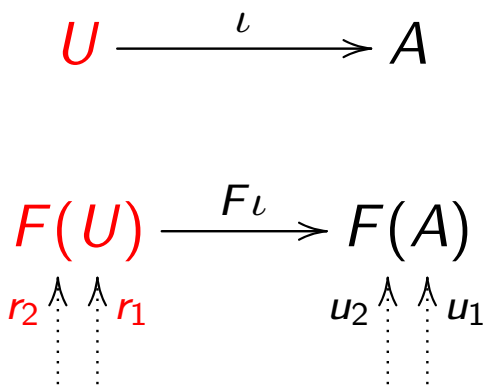
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]X \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$

$$\begin{array}{ccc} U & \xrightarrow{\quad\iota\quad} & A \\ & & \\ F(U) & \xrightarrow{\quad F\iota\quad} & F(A) \\ r \uparrow & & u \uparrow \end{array}$$

Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]X \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$



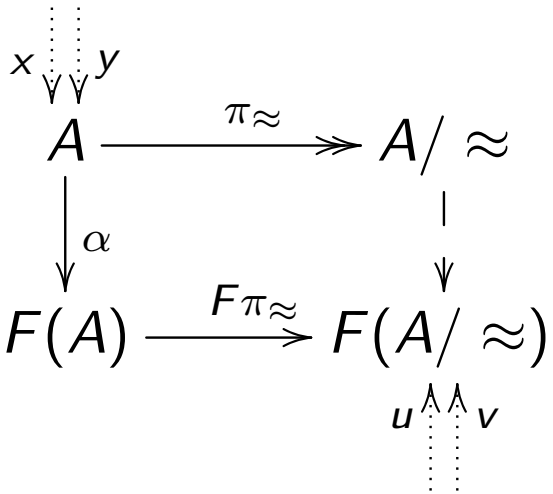
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]X \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$

$$\begin{array}{ccc} A & \xrightarrow{\pi_\approx} & A/\approx \\ \downarrow{\scriptstyle \alpha} & & \downarrow \\ F(A) & \xrightarrow{F\pi_\approx} & F(A/\approx) \end{array}$$

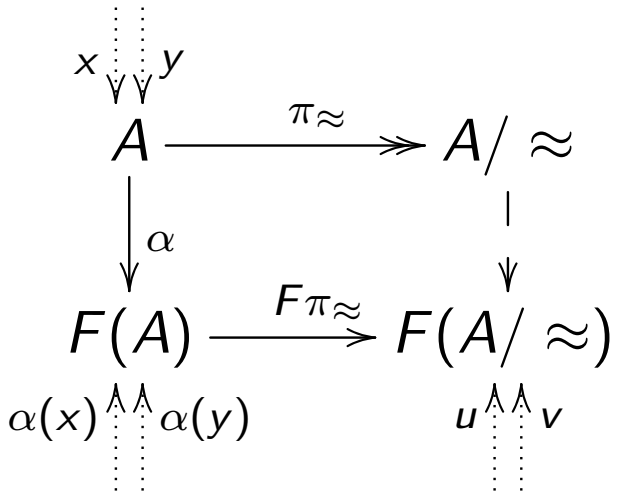
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]x \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$



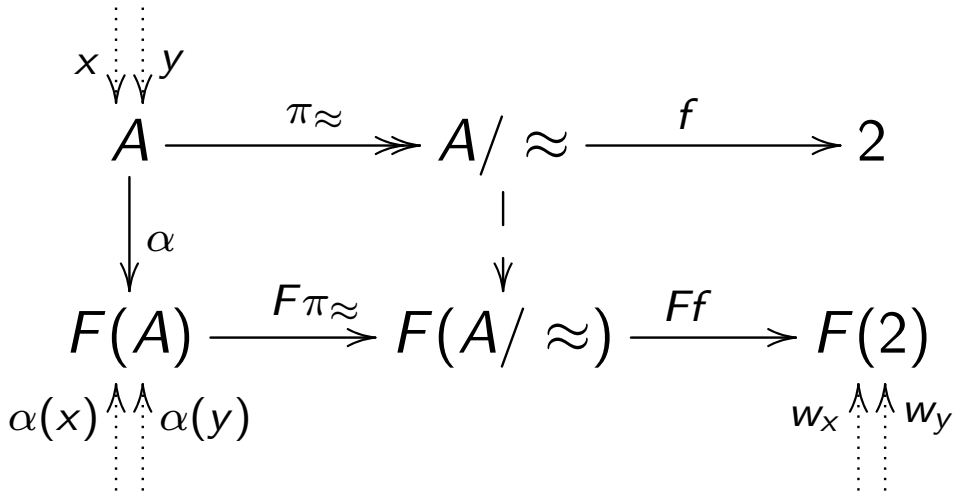
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]x \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$



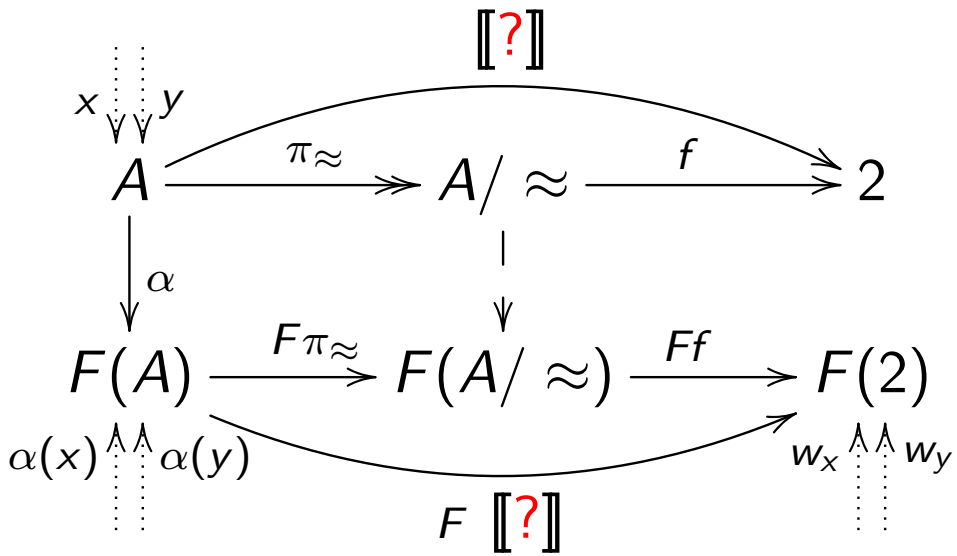
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]x \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$



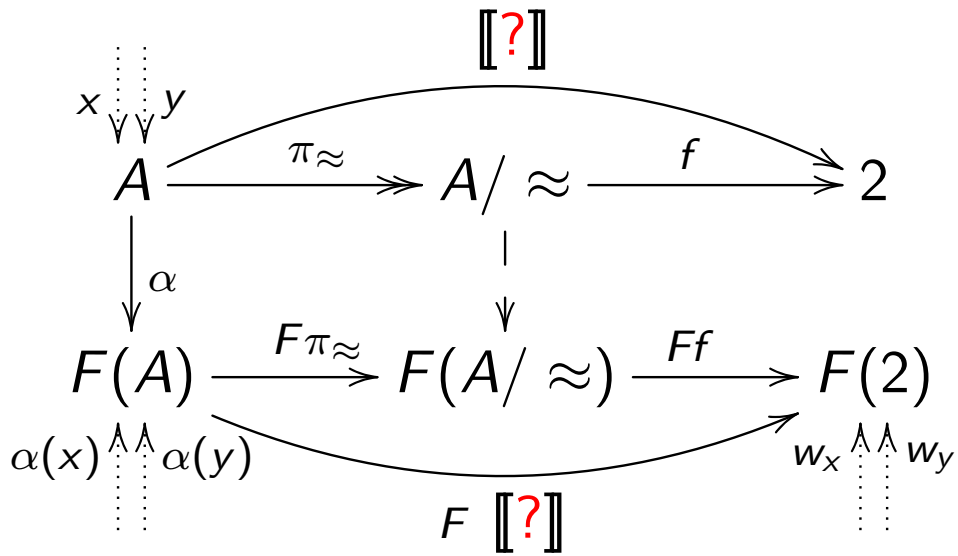
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]x \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$



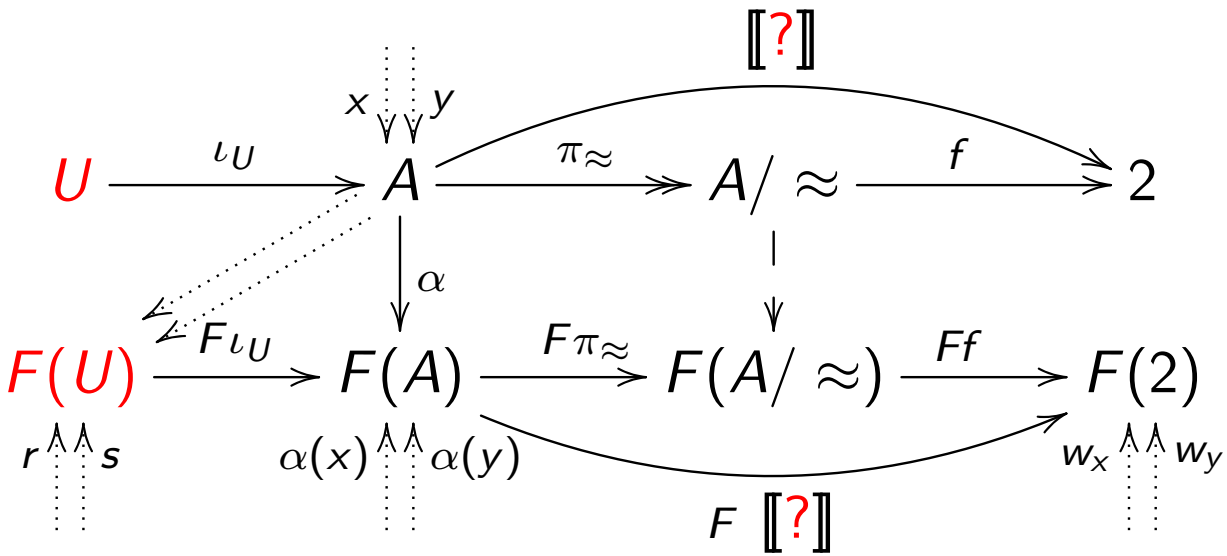
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]x \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$



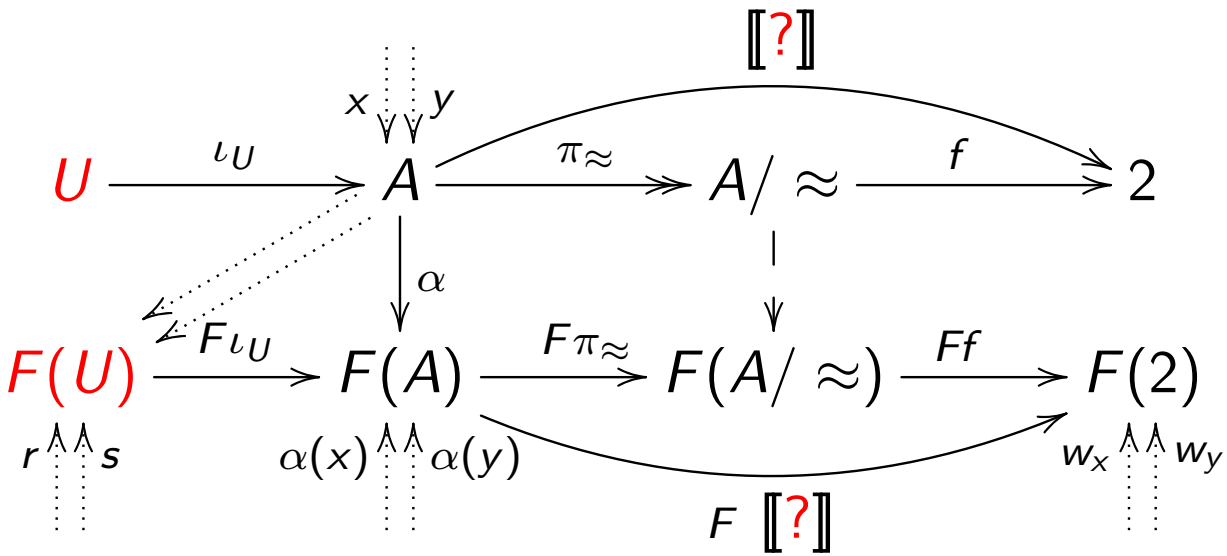
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]x \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$



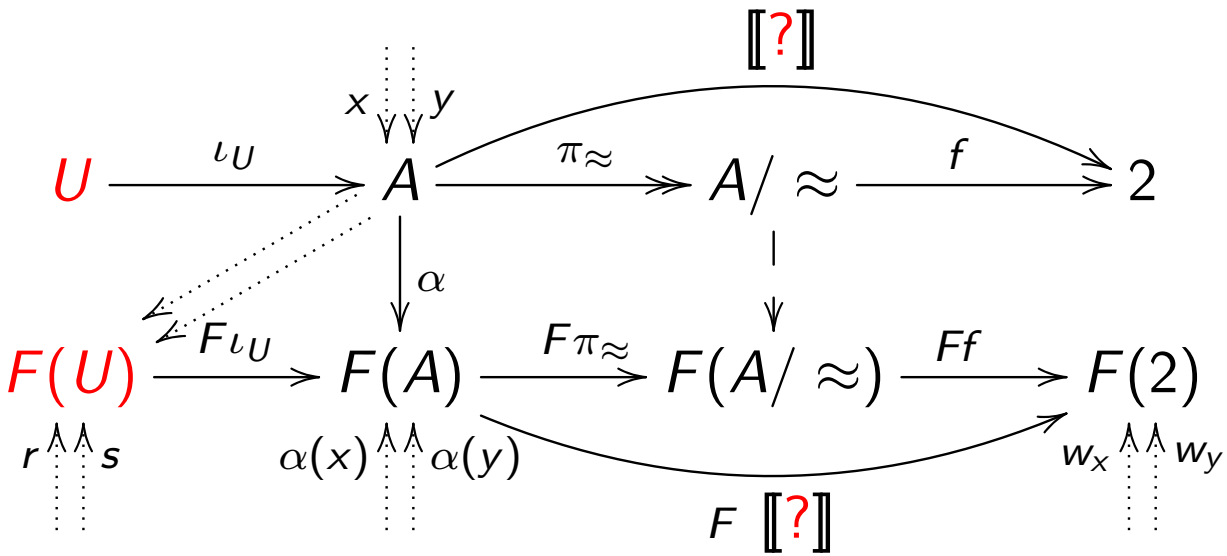
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]x \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$



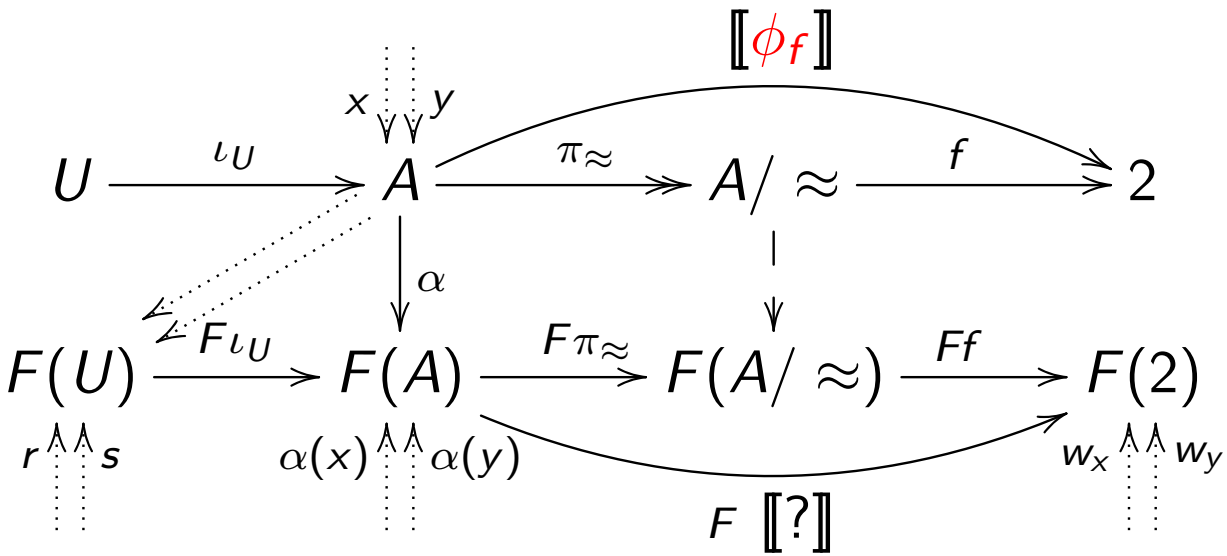
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]x \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$



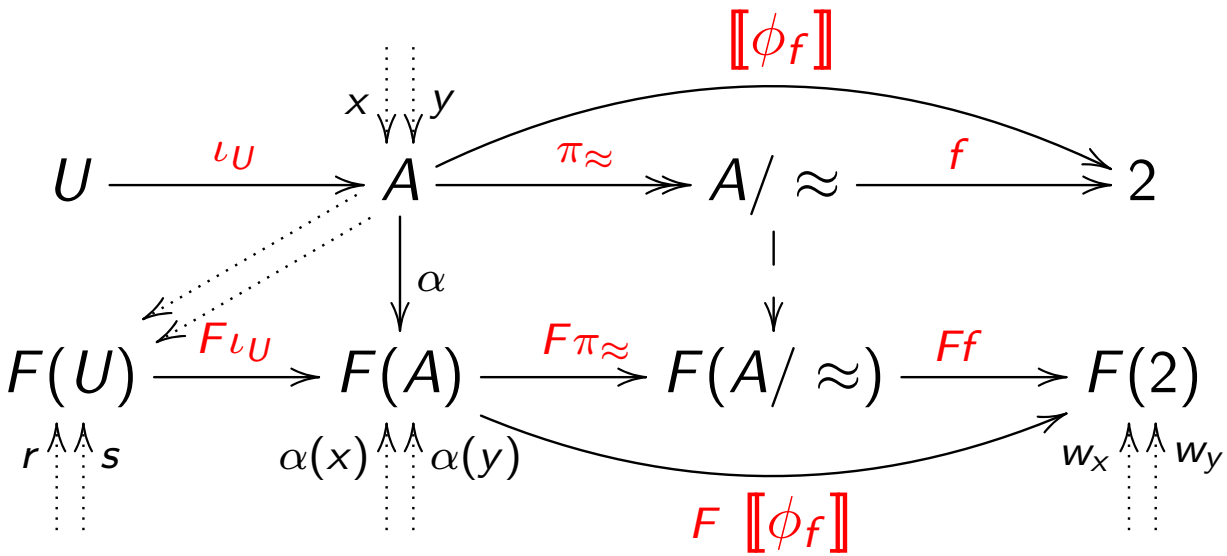
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]x \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$



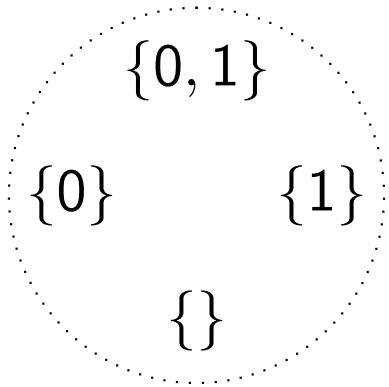
Finitary completeness

- Assume F is $\omega - \textit{bounded}$
  - $F(X) = \bigcup\{F[U]x \mid U \subseteq_{fin} X\}$
  - Choose $U \subseteq X$ with $\alpha(x), \alpha(y) \in F(U)$
  - $\phi_a := \bigvee\{\phi_{a,b} \mid a \not\approx b \in U\}$ defines a relative to U
  - $\phi_f = \bigwedge\{\phi_u \mid u \in U, f(u_\approx) = 1\}$



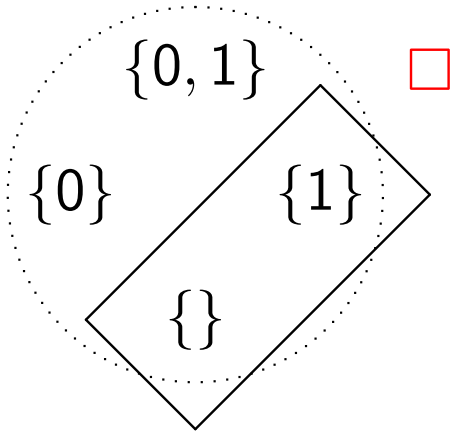
Modalities for $\mathbb{P}(-)$

$$x \models \Box\phi : \Longleftrightarrow \; x \models [\{\}]\phi \;\vee\; x \models [\{1\}]\phi$$



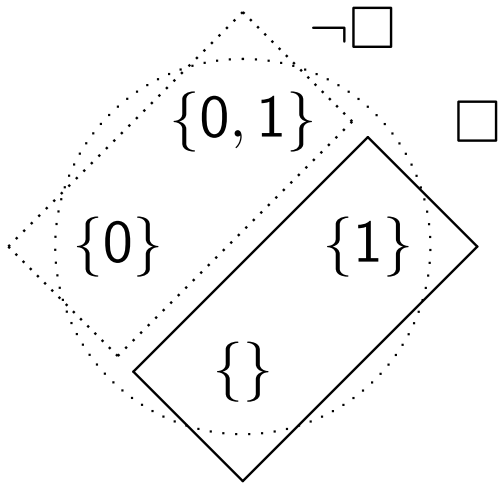
Modalities for $\mathbb{P}(-)$

$$x \models \Box\phi :\iff x \models [\{\}]\phi \ \vee \ x \models [\{1\}]\phi$$



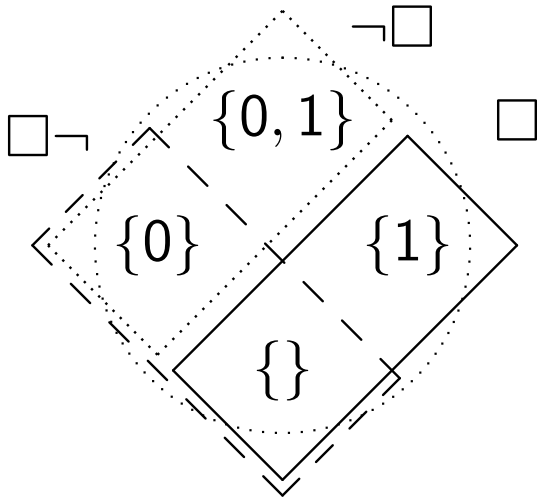
Modalities for $\mathbb{P}(-)$

$$x \models \Box\phi :\Longleftrightarrow\ x \models [\{\}]\phi\ \vee\ x \models [\{1\}]\phi$$



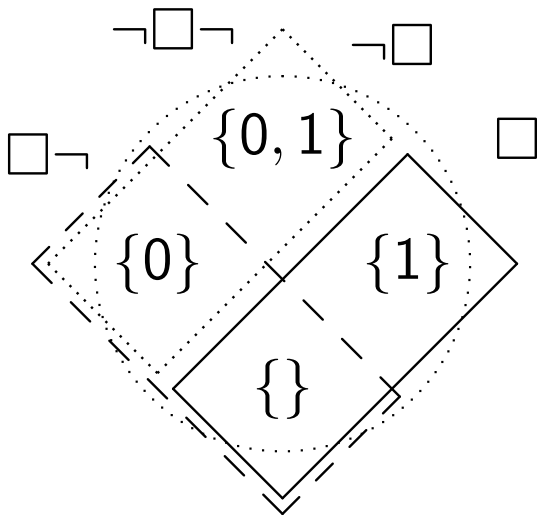
Modalities for $\mathbb{P}(-)$

$$x \models \Box\phi :\Longleftrightarrow\ x \models [\{\}]\phi\ \vee\ x \models [\{1\}]\phi$$



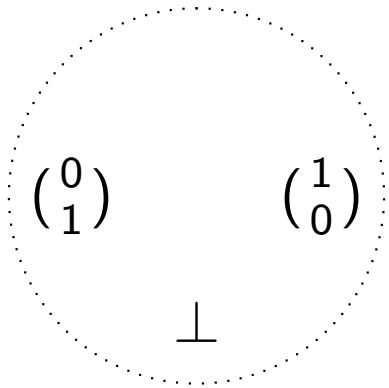
Modalities for $\mathbb{P}(-)$

$$x \models \Box\phi :\iff x \models [\{\}]\phi \ \vee \ x \models [\{1\}]\phi$$



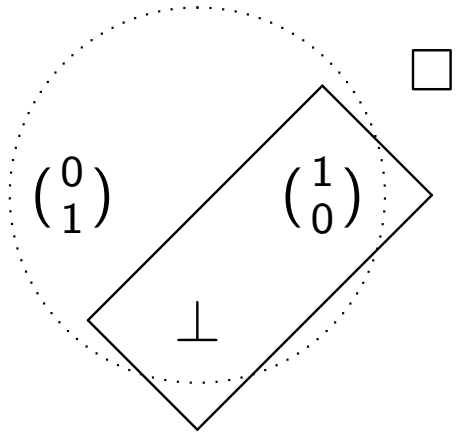
Modalities for $F(X) = X^2 - X + 1$

$$x \models \Box\phi :\iff x \models [\bot]\phi \ \vee \ x \models [\binom{1}{0}]\phi$$



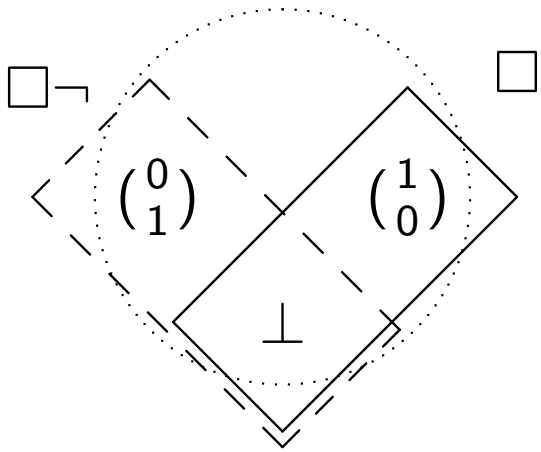
Modalities for $F(X) = X^2 - X + 1$

$$x \models \Box\phi :\iff x \models [\bot]\phi \ \vee \ x \models [\binom{1}{0}]\phi$$



Modalities for $F(X) = X^2 - X + 1$

$$x \models \Box\phi :\iff x \models [\bot]\phi \ \vee \ x \models [\binom{1}{0}]\phi$$



Some frame axioms

$$\begin{array}{rcl} & & \Box T \\ \Box v_1 \wedge \Box v_2 & \Longrightarrow & \Box(v_1 \wedge v_2) \\ (v_1 \Longrightarrow v_2) & \Longrightarrow & (\Box v_1 \Longrightarrow \Box v_2) \end{array}$$

Validity depends on choice of $\Box \subseteq F(2)$.

Theorem

- $\models \Box T$ **iff** $\frac{}{p(1) \in \Box}$
- $\models \Box v_1 \wedge \Box v_2 \Longrightarrow \Box(v_1 \wedge v_2)$ **iff** $\frac{p(1,1,0) \in \Box, p(1,0,1) \in \Box}{p(1,0,0) \in \Box}$
- $\models (v_1 \Longrightarrow v_2) \Longrightarrow (\Box v_1 \Longrightarrow \Box v_2)$ **iff** $\frac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

Some frame axioms

$$\begin{array}{rcl} & & \Box T \\ \Box v_1 \wedge \Box v_2 & \Longrightarrow & \Box(v_1 \wedge v_2) \\ (v_1 \Longrightarrow v_2) & \Longrightarrow & (\Box v_1 \Longrightarrow \Box v_2) \end{array}$$

Validity depends on choice of $\Box \subseteq F(2)$.

Theorem

- $\models \Box T$ **iff** $\frac{}{p(1) \in \Box}$
- $\models \Box v_1 \wedge \Box v_2 \Longrightarrow \Box(v_1 \wedge v_2)$ **iff** $\frac{p(1,1,0) \in \Box, p(1,0,1) \in \Box}{p(1,0,0) \in \Box}$
- $\models (v_1 \Longrightarrow v_2) \Longrightarrow (\Box v_1 \Longrightarrow \Box v_2)$ **iff** $\frac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

Some frame axioms

$$\begin{array}{rcl} & & \Box T \\ \Box v_1 \wedge \Box v_2 & \Longrightarrow & \Box(v_1 \wedge v_2) \\ (v_1 \Longrightarrow v_2) & \Longrightarrow & (\Box v_1 \Longrightarrow \Box v_2) \end{array}$$

Validity depends on choice of $\Box \subseteq F(2)$.

Theorem

- $\models \Box T$ **iff** $\frac{}{p(1) \in \Box}$
- $\models \Box v_1 \wedge \Box v_2 \Longrightarrow \Box(v_1 \wedge v_2)$ **iff** $\frac{p(1,1,0) \in \Box, p(1,0,1) \in \Box}{p(1,0,0) \in \Box}$
- $\models (v_1 \Longrightarrow v_2) \Longrightarrow (\Box v_1 \Longrightarrow \Box v_2)$ **iff** $\frac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

Some frame axioms

$$\begin{array}{rcl} & & \Box T \\ \Box v_1 \wedge \Box v_2 & \Longrightarrow & \Box(v_1 \wedge v_2) \\ (v_1 \Longrightarrow v_2) & \Longrightarrow & (\Box v_1 \Longrightarrow \Box v_2) \end{array}$$

Validity depends on choice of $\Box \subseteq F(2)$.

Theorem

- $\models \Box T$ **iff** $\frac{}{p(1) \in \Box}$
- $\models \Box v_1 \wedge \Box v_2 \Longrightarrow \Box(v_1 \wedge v_2)$ **iff** $\frac{p(1,1,0) \in \Box, p(1,0,1) \in \Box}{p(1,0,0) \in \Box}$
- $\models (v_1 \Longrightarrow v_2) \Longrightarrow (\Box v_1 \Longrightarrow \Box v_2)$ **iff** $\frac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

An examle proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

“$\Longrightarrow$”

An examle proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

"$\Longrightarrow$"

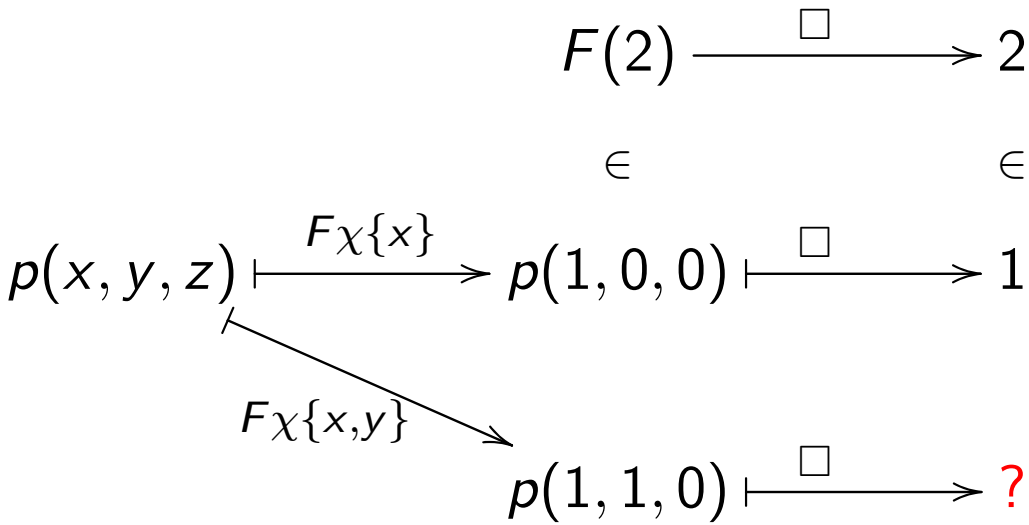
$$\begin{array}{ccc} F(2) & \xrightarrow{\Box} & 2 \\ \in & & \in \\ p(1,0,0) & \xmapsto{\Box} & 1 \\ \\ p(1,1,0) & \xmapsto{\Box} & \end{array}$$

An examle proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

"$\Longrightarrow$"

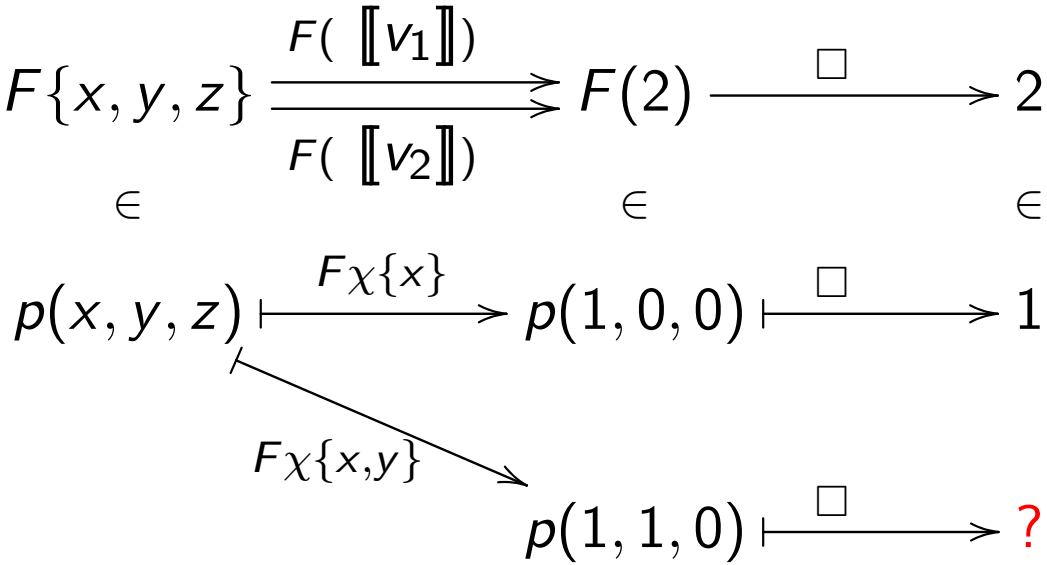


An examle proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

"$\Longrightarrow$"



An examle proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

"$\Longrightarrow$"

$$\{x,y,z\} \underset{[\![v_2]\!]=\chi\{x,y\}}{\overset{[\![v_1]\!]=\chi\{x\}}{\rightrightarrows}} 2$$

$$F\{x,y,z\} \underset{F([\![v_2]\!])}{\overset{F([\![v_1]\!])}{\rightrightarrows}} F(2) \xrightarrow{\Box} 2$$

$$\in \qquad\qquad \in \qquad\qquad \in$$

$$p(x,y,z) \xmapsto{F\chi\{x\}} p(1,0,0) \xmapsto{\Box} 1$$

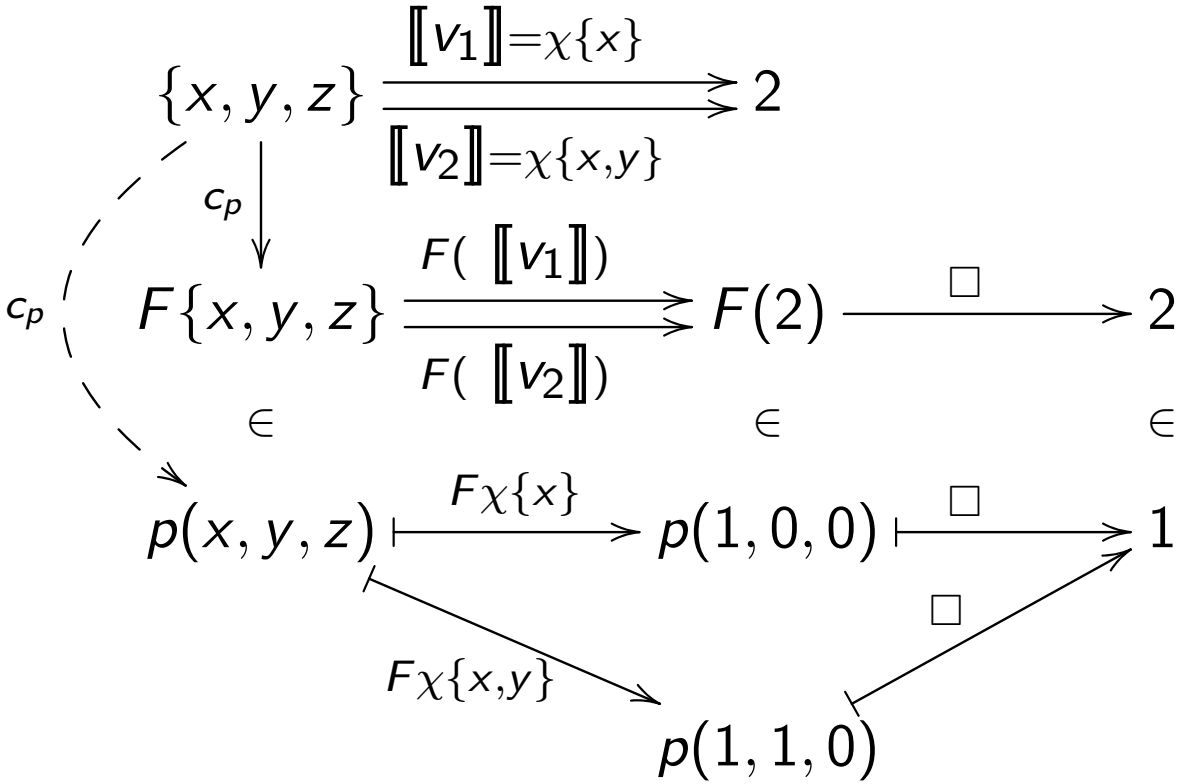
$$p(x,y,z) \xmapsto{F\chi\{x,y\}} p(1,1,0) \xmapsto{\Box} ?$$

An examle proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

"$\Longrightarrow$"



An example proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

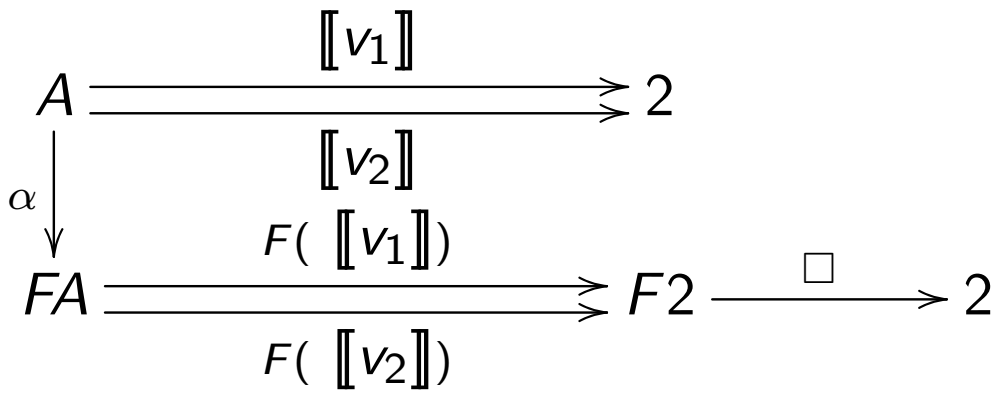
"$\Longleftarrow$"

An example proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

"$\Longleftarrow$"



An example proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

"$\Longleftarrow$"

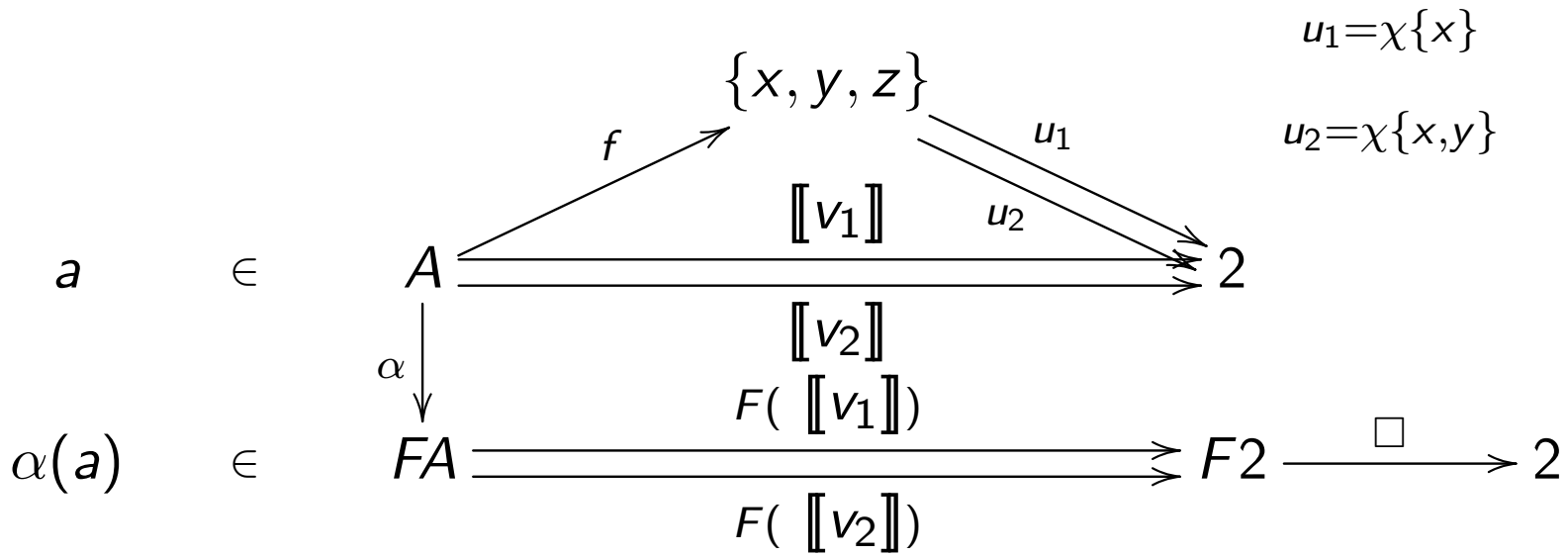
$$
\begin{array}{ccc}
a & \in & A \underset{[\![v_2]\!]}{\overset{[\![v_1]\!]}{\rightrightarrows}} 2 \\
 & & \alpha \downarrow \\
\alpha(a) & \in & FA \underset{F([\![v_2]\!])}{\overset{F([\![v_1]\!])}{\rightrightarrows}} F2 \xrightarrow{\Box} 2
\end{array}
$$

An example proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

"$\Longleftarrow$"

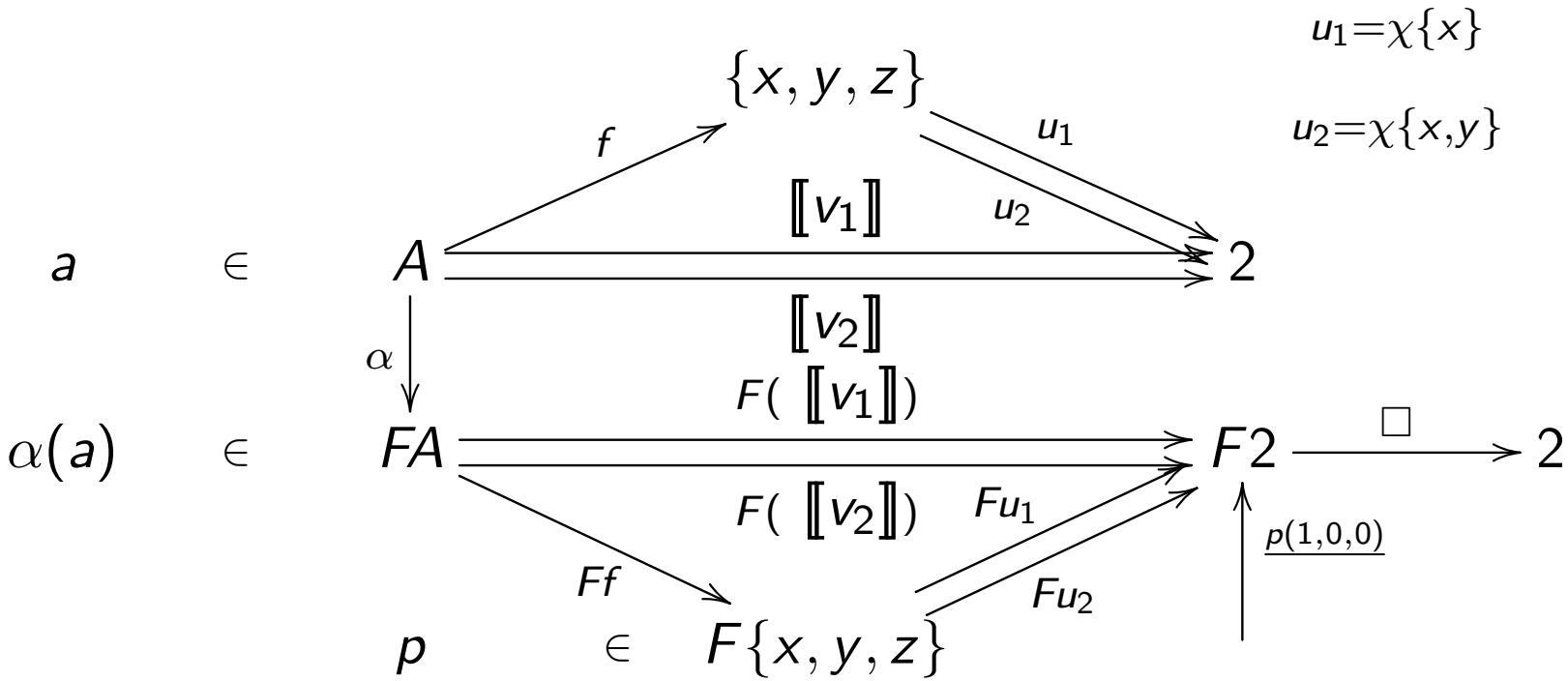


An example proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

"$\Longleftarrow$"

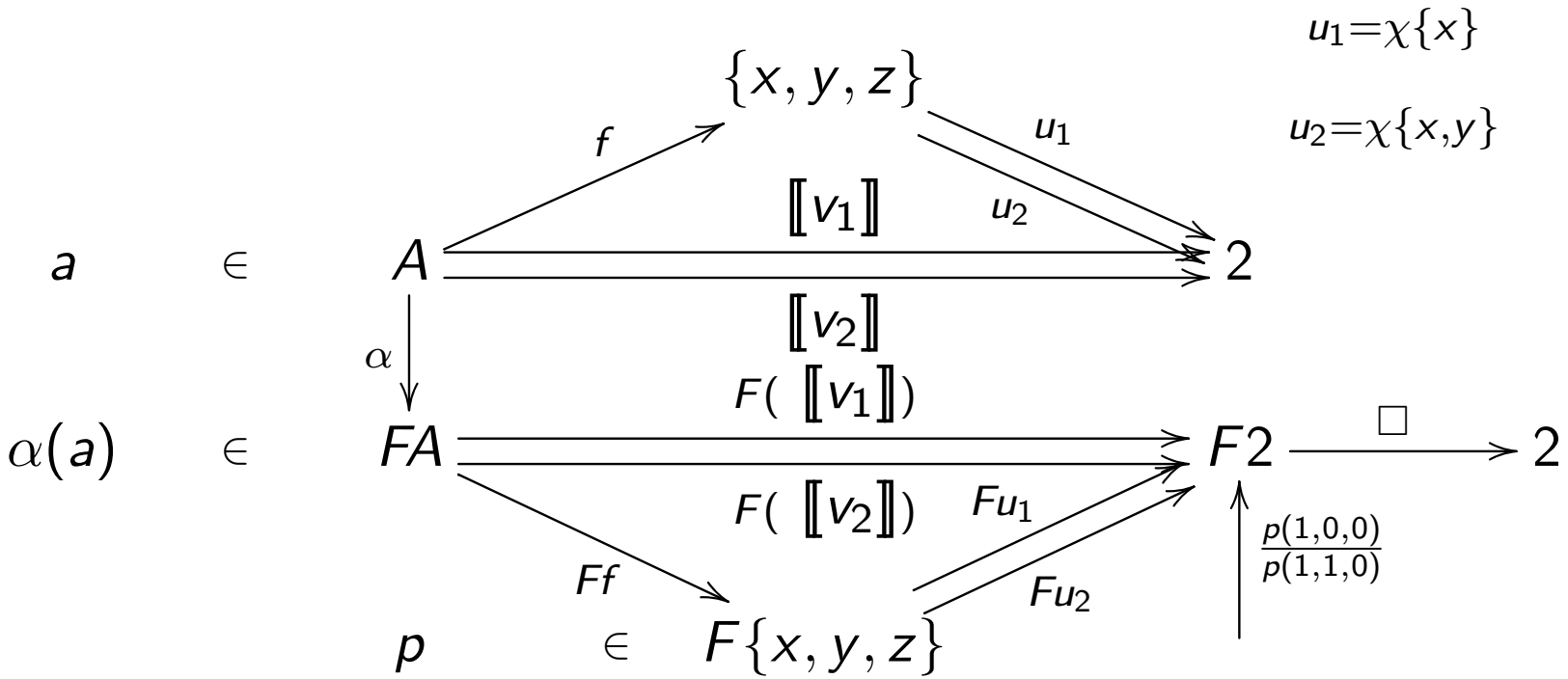


An example proof

Theorem

$(F, \Box)$ *satisfies* $\dfrac{v_1 \Longrightarrow v_2}{\Box v_1 \Longrightarrow \Box v_2} \iff \dfrac{p(1,0,0) \in \Box}{p(1,1,0) \in \Box}$

"$\Longleftarrow$"



Topological Spaces

- Topological Spaces

$$T = (X, \tau)$$

- Neighbourhoods

$$\mathfrak{N}(x) := \{U \subseteq X \mid \exists \mathcal{O} \in \tau . a \in \mathcal{O} \subseteq U\}$$

- Topologies as Coalgebras

Topological Spaces

- Topological Spaces

$$T = (X, \tau)$$

- Neighbourhoods

$$\mathfrak{N}(x) := \{U \subseteq X \mid \exists \mathcal{O} \in \tau . a \in \mathcal{O} \subseteq U\}$$

- Topologies as Coalgebras

Topological Spaces

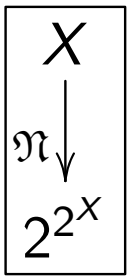
- Topological Spaces

$$T = (X, \tau)$$

- Neighbourhoods

$$\mathfrak{N}(x) := \{U \subseteq X \mid \exists \mathcal{O} \in \tau . a \in \mathcal{O} \subseteq U\}$$

- Topologies as Coalgebras



The functor 2^{2^-}

On objects:

$$2^{2^X} = \mathbb{PP}(X)$$

On maps:

$$f : X \longrightarrow Y \qquad \text{yields} \qquad 2^{2^f} : 2^{2^X} \longrightarrow 2^{2^Y}$$

where

$$2^{2^f}(\sigma) = \{V \subseteq Y \mid f^{-1}[V] \in \sigma\}$$

The functor 2^{2^-}

On objects:

$$2^{2^X} = \mathbb{PP}(X)$$

On maps:

$$f : X \longrightarrow Y \qquad \textit{yields} \qquad 2^{2^f} : 2^{2^X} \longrightarrow 2^{2^Y}$$

where

$$2^{2^f}(\sigma) = \{V \subseteq Y \mid f^{-1}[V] \in \sigma\}$$

The functor 2^{2^-}

On objects:

$$2^{2^X} = \mathbb{PP}(X)$$

On maps:

$$f : X \longrightarrow Y \qquad \textit{yields} \qquad 2^{2^f} : 2^{2^X} \longrightarrow 2^{2^Y}$$

where

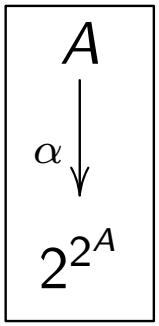
$$2^{2^f}(\sigma) = \{V \subseteq Y \mid f^{-1}[V] \in \sigma\}$$

2^{2^-}-coalgebras: Neighbourhood systems

- 2^{2^-}-coalgebra:
- Notation:

  $a \xrightarrow[\alpha]{} \triangleright U \quad :\Longleftrightarrow$

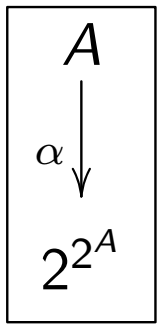
  $U \in \alpha(a)$
- Homomorphism



2^{2^-}-coalgebras: Neighbourhood systems

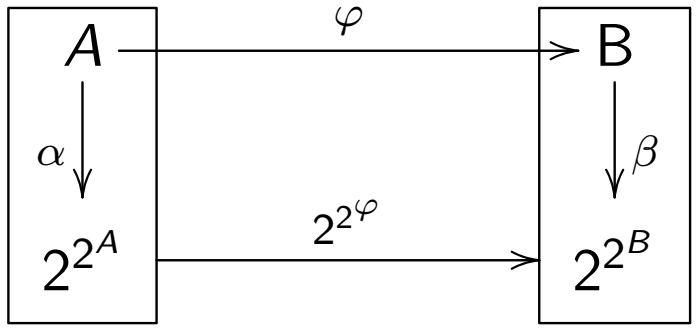
- 2^{2^-}-coalgebra:
- Notation:

  $a \underset{\alpha}{\multimap} \triangleright U \quad :\Longleftrightarrow \quad U \in \alpha(a)$
- Homomorphism



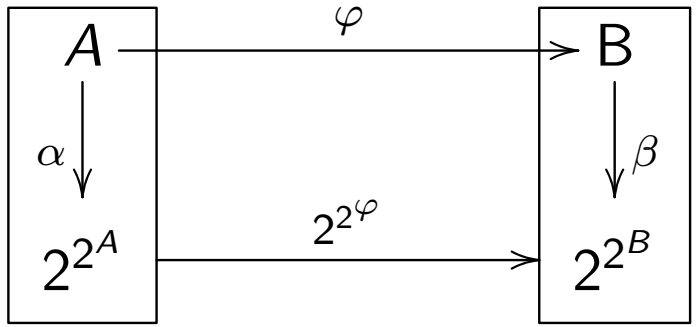
2^{2^-}-coalgebras: Neighbourhood systems

- 2^{2^-}-coalgebra:
- Notation:
  $a \underset{\alpha}{\multimap} U \quad :\Longleftrightarrow \quad U \in \alpha(a)$
- Homomorphism



2^{2^-}-coalgebras: Neighbourhood systems

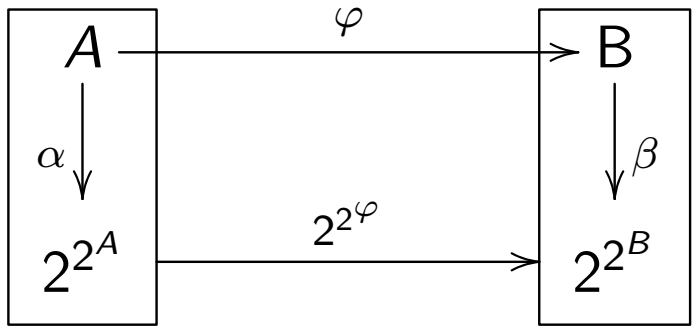
- 2^{2^-}-coalgebra:
- Notation:
  $a \underset{\alpha}{\longrightarrow\!\!\!\!\!\!\triangleright}\ U \quad :\Longleftrightarrow\quad U \in \alpha(a)$
- Homomorphism



$$a \longrightarrow\!\!\!\!\!\!\triangleright\ \varphi^{-1}[V] \iff \varphi(a) \longrightarrow\!\!\!\!\!\!\triangleright\ V$$

2^{2^-}-coalgebras: Neighbourhood systems

- 2^{2^-}-coalgebra:
- Notation:
  $a \underset{\alpha}{\multimap} U \quad :\Longleftrightarrow \quad U \in \alpha(a)$
- Homomorphism



Topological spaces with open maps form a subcovariety of neighbourhood systems.

UnCurry-Curry

Curry:

$$X \times Y \to Z \cong X \to Z^Y$$

UnCurry-Curry

Curry:

$$X \times Y \to Z \cong X \to Z^Y$$

Functor action:

$$A \xrightarrow{f} 2 \qquad \mapsto \qquad F(A) \xrightarrow{Ff} F(2)$$

UnCurry-Curry

Curry:

$$X \times Y \to Z \cong X \to Z^Y$$

Functor action:

$$2^A \xrightarrow{\quad F \quad} F(2)^{F(A)}$$

UnCurry-Curry

Curry:

$$X \times Y \to Z \cong X \to Z^Y$$

Functor action:

$$2^A \xrightarrow{\quad F \quad} F(2)^{F(A)}$$

Uncurried:

$$2^A \times F(A) \longrightarrow F(2)$$

UnCurry-Curry

Curry:

$$X \times Y \to Z \cong X \to Z^Y$$

Functor action:

$$2^A \xrightarrow{\quad F \quad} F(2)^{F(A)}$$

Uncurried:

$$2^A \times F(A) \longrightarrow F(2)$$

Curried:

$$F(A) \longrightarrow F(2)^{2^A}$$

UnCurry-Curry

Curry:

$$X \times Y \to Z \cong X \to Z^Y$$

Functor action:

$$2^A \xrightarrow{\quad F \quad} F(2)^{F(A)}$$

Uncurried:

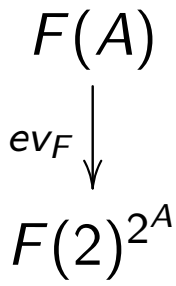
$$2^A \times F(A) \longrightarrow F(2)$$

Curried:

$$F(A) \xrightarrow{\quad ev_F \quad} F(2)^{2^A}$$

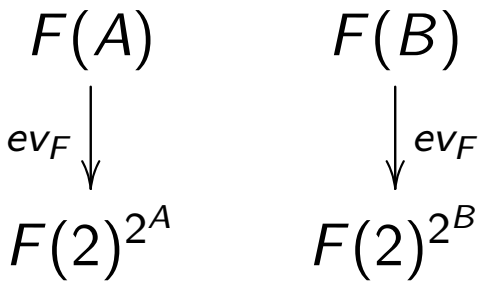
Naturality

It's a natural transformation ...



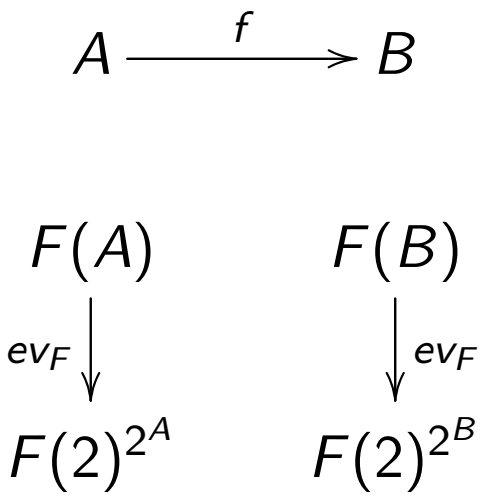
Naturality

It's a natural transformation ...



Naturality

It's a natural transformation ...



Naturality

It's a natural transformation ...

$$A \xrightarrow{f} B$$

$$\begin{array}{ccc} F(A) & \xrightarrow{Ff} & F(B) \\ {\scriptstyle ev_F}\downarrow & & \downarrow{\scriptstyle ev_F} \\ F(2)^{2^A} & \xrightarrow{2^{2^f}} & F(2)^{2^B} \end{array}$$

Naturality

It's a natural transformation ...

$$\begin{array}{ccc} A & \xrightarrow{f} & B \end{array}$$

$$\begin{array}{ccc} F(A) & \xrightarrow{Ff} & F(B) \\ {\scriptstyle ev_F}\downarrow & & \downarrow{\scriptstyle ev_F} \\ F(2)^{2^A} & \xrightarrow{2^{2^f}} & F(2)^{2^B} \end{array} \qquad \begin{array}{c} F(2) \\ \downarrow \\ \square \\ \downarrow \\ 2 \end{array}$$

Naturality

It's a natural transformation ...

$$A \xrightarrow{f} B$$

$$\begin{array}{ccc} F(A) & \xrightarrow{Ff} & F(B) \\ {\scriptstyle ev_F}\downarrow & & \downarrow{\scriptstyle ev_F} \\ F(2)^{2^A} & \xrightarrow[2^{2^f}]{} & F(2)^{2^B} \\ {\scriptstyle \hat{\Box}}\downarrow & & \\ 2^{2^A} & & \end{array} \qquad \begin{array}{c} F(2) \\ \downarrow \\ \Box \\ \downarrow \\ 2 \end{array}$$

Naturality

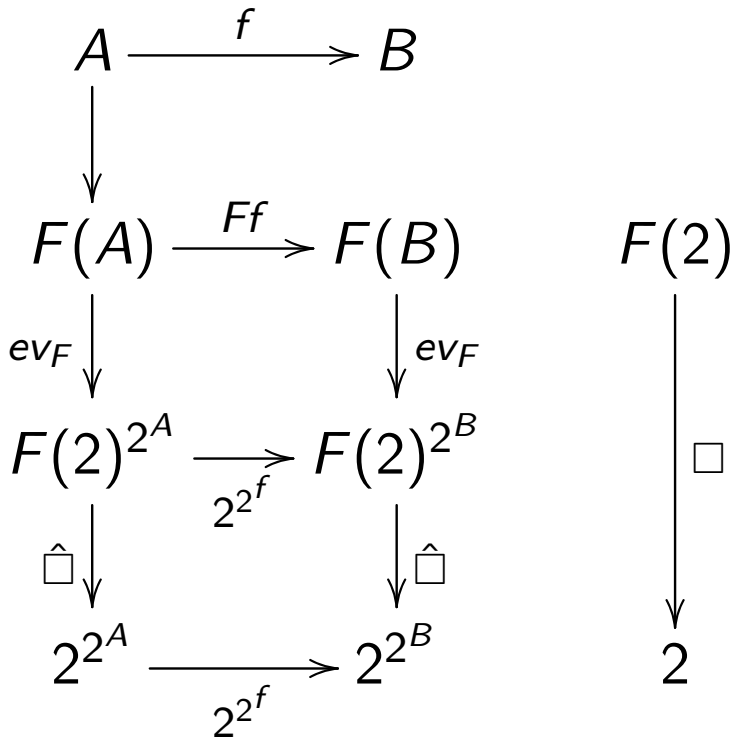
It's a natural transformation ...

$$A \xrightarrow{f} B$$

$$\begin{array}{ccc} F(A) & \xrightarrow{Ff} & F(B) \\ {\scriptstyle ev_F}\downarrow & & \downarrow{\scriptstyle ev_F} \\ F(2)^{2^A} & \xrightarrow[2^{2^f}]{} & F(2)^{2^B} \\ {\scriptstyle \hat{\Box}}\downarrow & & \downarrow{\scriptstyle \hat{\Box}} \\ 2^{2^A} & \xrightarrow[2^{2^f}]{} & 2^{2^B} \end{array} \qquad \begin{array}{c} F(2) \\ \downarrow{\scriptstyle \Box} \\ 2 \end{array}$$

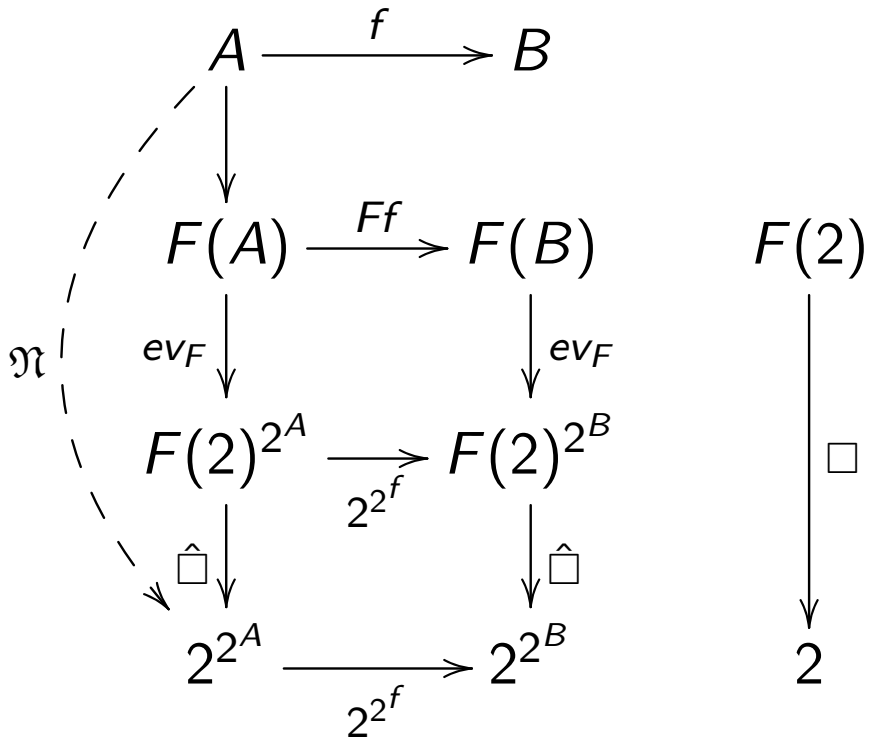
Naturality

It's a natural transformation ...



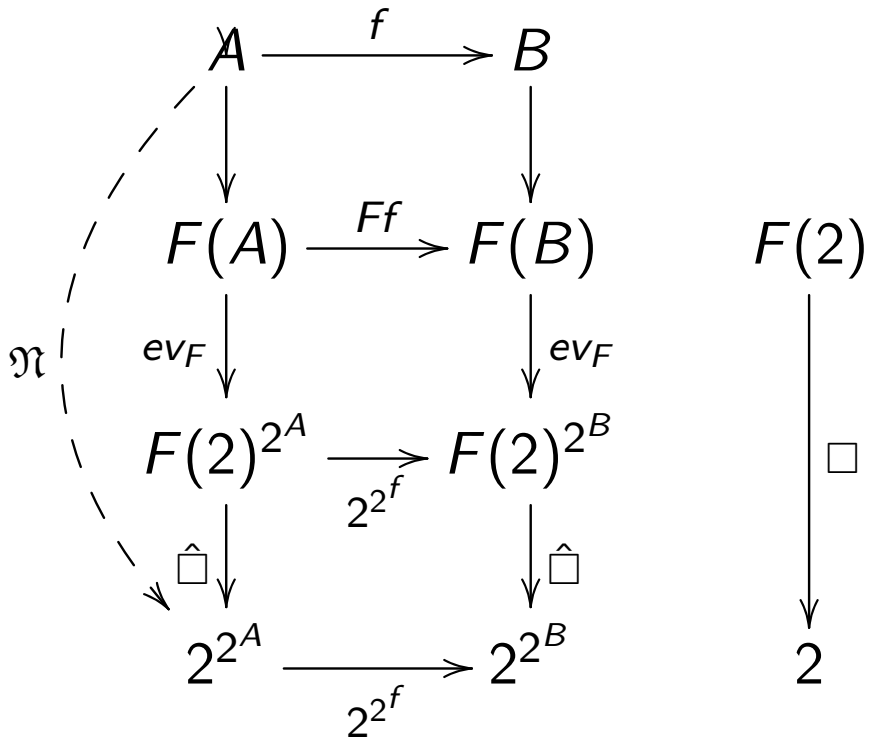
Naturality

It's a natural transformation ...



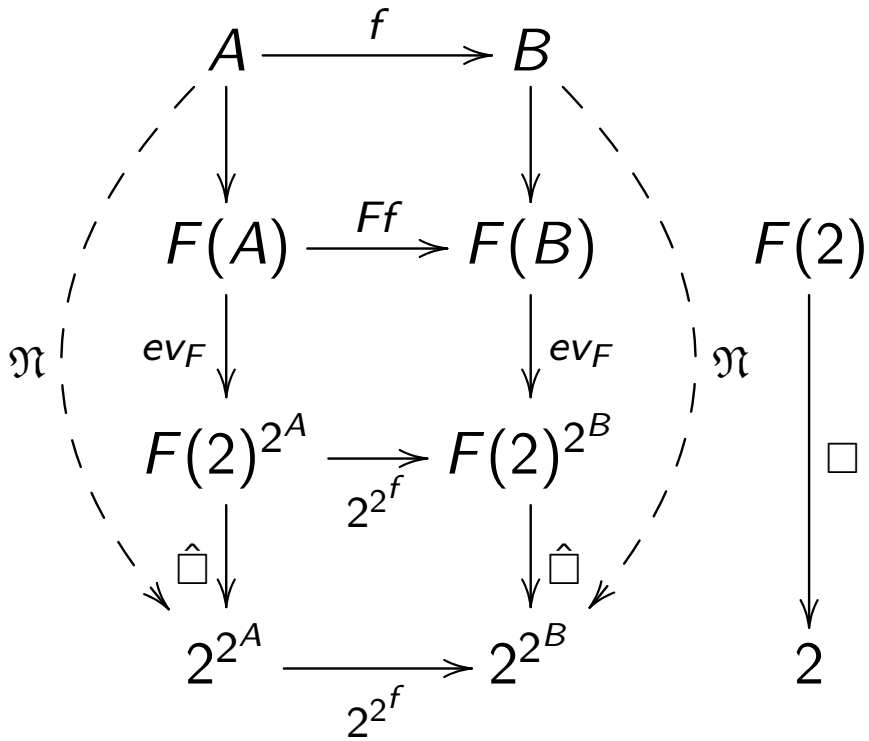
Naturality

It's a natural transformation ...



Naturality

It's a natural transformation ...



A full embedding

- Each $\Box_i : F(2) \to 2$ yields a functor

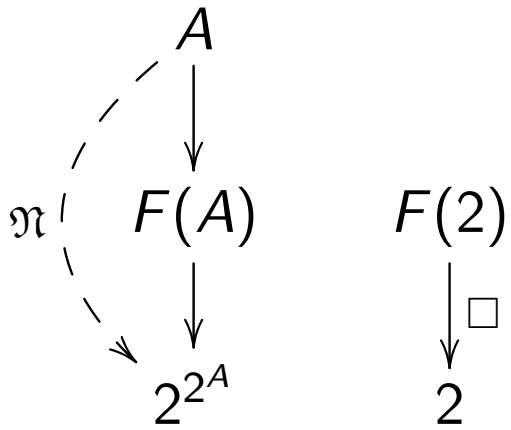
$$\mathcal{S}et_F \to \mathcal{S}et_{2^{2^-}}$$

- each bit-encoding $\Box : F(2) \hookrightarrow 2^\kappa$ where $\Box_i = \pi_i \circ \Box$ yields a full embedding:

$$\mathcal{S}et_F \hookrightarrow (\mathcal{S}et_{2^{2^-}})^\kappa$$

- Relation to modal logic:

$$a \models \Box_i \phi \qquad \Longleftrightarrow \qquad a \xrightarrow{\quad}_i \triangleright [\![\phi]\!]$$



A full embedding

- Each $\Box_i : F(2) \to 2$ yields a functor

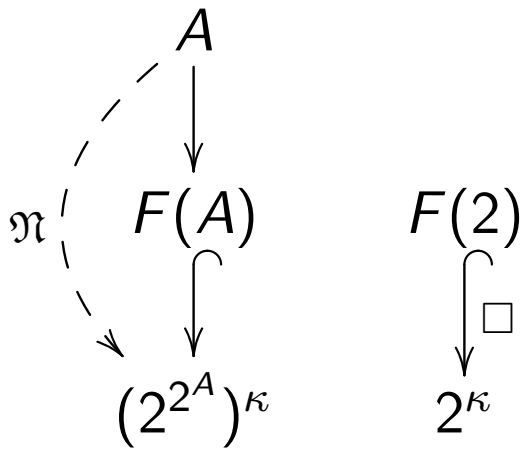
$$\mathcal{S}et_F \to \mathcal{S}et_{2^{2^-}}$$

- each bit-encoding $\Box : F(2) \hookrightarrow 2^\kappa$ where $\Box_i = \pi_i \circ \Box$ yields a full embedding:

$$\mathcal{S}et_F \hookrightarrow (\mathcal{S}et_{2^{2^-}})^\kappa$$

- Relation to modal logic:

$$a \models \Box_i \phi \qquad \Longleftrightarrow \qquad a \xrightarrow[i]{} \triangleright \; [\![\phi]\!]$$



A full embedding

- Each $\Box_i : F(2) \to 2$ yields a functor

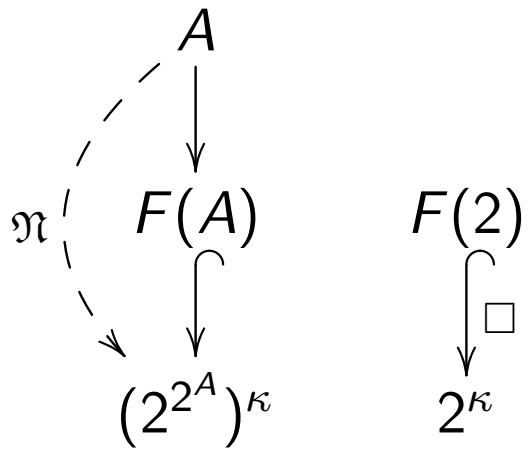
$$\mathcal{S}et_F \to \mathcal{S}et_{2^{2-}}$$

- each bit-encoding $\Box : F(2) \hookrightarrow 2^\kappa$ where $\Box_i = \pi_i \circ \Box$ yields a full embedding:

$$\mathcal{S}et_F \hookrightarrow (\mathcal{S}et_{2^{2-}})^\kappa$$

- Relation to modal logic:

$$a \models \Box_i \phi \qquad \Longleftrightarrow \qquad a \xrightarrow[i]{} \triangleright \; [\![\phi]\!]$$



Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - minimality
    - observational equivalence
    - coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
    - ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - minimality
    - observational equivalence
    - coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
    - ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - ⋆ minimality
    - ⋆ observational equivalence
    - ⋆ coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
  - ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - ★ minimality
    - ★ observational equivalence
    - ★ coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
    - ★ ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - ★ minimality
    - ★ observational equivalence
    - ★ coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
    - ★ ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - ★ minimality
    - ★ observational equivalence
    - ★ coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
    - ★ ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - ★ minimality
    - ★ observational equivalence
    - ★ coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
    - ★ ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - ★ minimality
    - ★ observational equivalence
    - ★ coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
    - ★ ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - ★ minimality
    - ★ observational equivalence
    - ★ coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
  - ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - ★ minimality
    - ★ observational equivalence
    - ★ coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
    - ★ ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - ★ minimality
    - ★ observational equivalence
    - ★ coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
    - ★ ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - ★ minimality
    - ★ observational equivalence
    - ★ coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
    - ★ ... but that's another story ...

Conclusion

- Coalgebras model state based systems
  - elegant algebraic theory
  - covers all familiar systems
    - ★ minimality
    - ★ observational equivalence
    - ★ coequational (global) logic
- Modal logic
  - framework for arbitrary coalgebra
  - adequate and expressive
  - finitary for finitary type functor
- A lot more
  - Algebra-coalgebra maps
  - Recursive coalgebras
    - ★ ... but that's another story ...

Thanks

$$T^h @_n x$$