How to Pull Back Open Maps along Semantics Functors

Uli Fahrenberg

Department of Computer Science
Aalborg University
Denmark

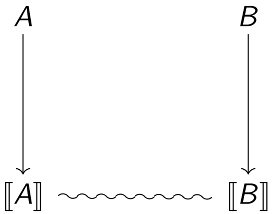
ACCAT 2008

Example

$$\begin{array}{c} A \\ \big\downarrow \\ [\![A]\!] \end{array}$$

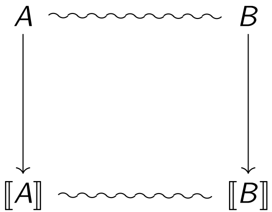
- timed automata $\rightsquigarrow$ operational semantics: transition systems

Example



- timed automata $\rightsquigarrow$ operational semantics: transition systems
- transition systems $\rightsquigarrow$ notion of bisimulation

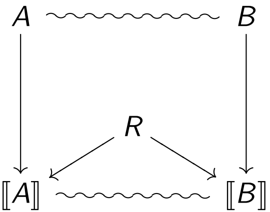
Example



- timed automata $\rightsquigarrow$ operational semantics: transition systems
- transition systems $\rightsquigarrow$ notion of bisimulation

$\rightsquigarrow$ bisimulation for timed automata

Example

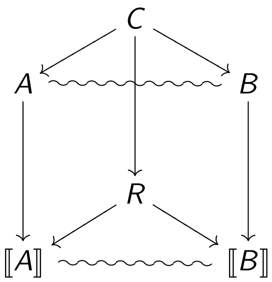


- timed automata $\rightsquigarrow$ operational semantics: transition systems
- transition systems $\rightsquigarrow$ notion of bisimulation

$\rightsquigarrow$ bisimulation for timed automata

- transition systems $\rightsquigarrow$ notion of open maps
- Two transition systems are bisimilar if and only if they are connected by a "span" of open maps.

Example



- timed automata $\rightsquigarrow$ operational semantics: transition systems
- transition systems $\rightsquigarrow$ notion of bisimulation

$\rightsquigarrow$ bisimulation for timed automata

- transition systems $\rightsquigarrow$ notion of open maps
- want to "pull back" these open maps "along the semantics functor"

Open maps

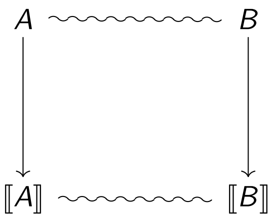
- [Joyal, Nielsen, Winskel: *Bisimulation from open maps.* Information and Computation 127(2), 1996]
- standard models (presheaves)
- standard logics
- relations between different formalisms ((co)reflective functors)
- connection to algebraic topology (weak factorization systems, model categories)

Generalization

$$\begin{array}{c} A \\ \big\downarrow \\ [\![A]\!] \end{array}$$

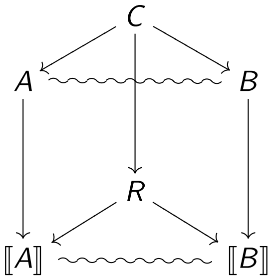
- some formalism $\mathcal{M}$ $\rightsquigarrow$ operational semantics in a category $\mathcal{T}$ with open maps

Generalization



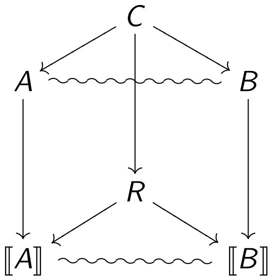
- some formalism $\mathcal{M} \rightsquigarrow$ operational semantics in a category $\mathcal{T}$ with open maps
- bisimulation in $\mathcal{T}$ used for defining bisimulation in $\mathcal{M}$

Generalization



- some formalism $\mathcal{M} \rightsquigarrow$ operational semantics in a category $\mathcal{T}$ with open maps
- bisimulation in $\mathcal{T}$ used for defining bisimulation in $\mathcal{M}$
- use open maps in $\mathcal{T}$ for introducing open maps in $\mathcal{M}$

Generalization



- some formalism $\mathcal{M}$ $\rightsquigarrow$ operational semantics in a category $\mathcal{T}$ with open maps
- bisimulation in $\mathcal{T}$ used for defining bisimulation in $\mathcal{M}$
- use open maps in $\mathcal{T}$ for introducing open maps in $\mathcal{M}$
- (and possibly morphisms in $\mathcal{M}$ as such)

Open maps

- transition system:
  - S states
  - $s^0 \in S$ initial state
  - Σ labels
  - $E \subseteq S \times \Sigma \times S$ transitions

Open maps

- transition system: $(S, s^0, \Sigma, E \subseteq S \times \Sigma \times S)$
- morphism of transition systems $(S_1, s_1^0, \Sigma, E_1)$, $(S_2, s_2^0, \Sigma, E_2)$: $f : S_1 \to S_2$ such that

$$f(s_1^0) = s_2^0$$
$$(s, a, s') \in E_1 \implies (f(s), a, f(s')) \in E_2$$

- morphisms are *functional simulations*
- (in actual fact, morphisms can also change the labeling. We don't need this here)

Open maps

- transition system: $(S, s^0, \Sigma, E \subseteq S \times \Sigma \times S)$
- morphism of transition systems $(S_1, s_1^0, \Sigma, E_1)$, $(S_2, s_2^0, \Sigma, E_2)$: $f : S_1 \to S_2$ such that

$$f(s_1^0) = s_2^0$$
$$(s, a, s') \in E_1 \quad \Longrightarrow \quad (f(s), a, f(s')) \in E_2$$

$\leadsto$ category of transition systems

- well-behaved category; natural constructions are well-known; relates to other formalisms by (reflective) functors
- [Winskel, Nielsen: *Models for concurrency*. In Handbook of Logic in Computer Science, Oxford Univ. Press 1995]

Open maps

- transition system: $(S, s^0, \Sigma, E \subseteq S \times \Sigma \times S)$
- morphism of transition systems $(S_1, s_1^0, \Sigma, E_1), (S_2, s_2^0, \Sigma, E_2)$: $f : S_1 \to S_2$ such that

$$f(s_1^0) = s_2^0$$
$$(s, a, s') \in E_1 \implies (f(s), a, f(s')) \in E_2$$

$\leadsto$ category of transition systems

- a morphism $f : (S_1, s_1^0, \Sigma, E_1) \to (S_2, s_2^0, \Sigma, E_2)$ is open if

Open maps

- transition system: $(S, s^0, \Sigma, E \subseteq S \times \Sigma \times S)$
- morphism of transition systems $(S_1, s_1^0, \Sigma, E_1)$, $(S_2, s_2^0, \Sigma, E_2)$: $f : S_1 \to S_2$ such that

$$f(s_1^0) = s_2^0$$
$$(s, a, s') \in E_1 \quad \Longrightarrow \quad (f(s), a, f(s')) \in E_2$$

⇝ category of transition systems

- a morphism $f : (S_1, s_1^0, \Sigma, E_1) \to (S_2, s_2^0, \Sigma, E_2)$ is open if

$\forall$ reachable $s_1 \in S_1$

$$\begin{array}{ccc} S_1 & & s_1 \\ {\scriptstyle f}\downarrow & & \\ S_2 & & \end{array}$$

Open maps

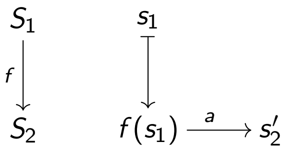
- transition system: $(S, s^0, \Sigma, E \subseteq S \times \Sigma \times S)$
- morphism of transition systems $(S_1, s_1^0, \Sigma, E_1)$, $(S_2, s_2^0, \Sigma, E_2)$: $f : S_1 \to S_2$ such that

$$f(s_1^0) = s_2^0$$
$$(s, a, s') \in E_1 \implies (f(s), a, f(s')) \in E_2$$

⇝ category of transition systems

- a morphism $f : (S_1, s_1^0, \Sigma, E_1) \to (S_2, s_2^0, \Sigma, E_2)$ is open if

$\forall$ reachable $s_1 \in S_1$
$\forall$ edges $(f(s_1), a, s_2') \in E_2$



Open maps

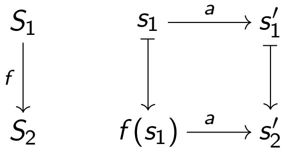
- transition system: $(S, s^0, \Sigma, E \subseteq S \times \Sigma \times S)$
- morphism of transition systems $(S_1, s_1^0, \Sigma, E_1)$, $(S_2, s_2^0, \Sigma, E_2)$: $f : S_1 \to S_2$ such that

$$f(s_1^0) = s_2^0$$
$$(s, a, s') \in E_1 \implies (f(s), a, f(s')) \in E_2$$

$\rightsquigarrow$ category of transition systems

- a morphism $f : (S_1, s_1^0, \Sigma, E_1) \to (S_2, s_2^0, \Sigma, E_2)$ is open if

$\forall$ reachable $s_1 \in S_1$
$\forall$ edges $(f(s_1), a, s_2') \in E_2$
$\exists$ edge $(s_1, a, s_1') \in E_1$ for which $s_2' = f(s_1')$



Open maps and bisimulation

- (again:) a morphism $f : (S_1, s_1^0, \Sigma E_1) \to (S_2, s_2^0, \Sigma, E_2)$ is open if
  - $\forall$ reachable $s_1 \in S_1$
  - $\forall$ edges $\big(f(s_1), a, s_2'\big) \in E_2$
  - $\exists$ edge $(s_1, a, s_1') \in E_1$ for which $s_2' = f(s_1')$
- open map $f : (S_1, s_1^0, \Sigma, E_1) \to (S_2, s_2^0, \Sigma, E_2) \rightsquigarrow$ bisimulation

$$R = \Big\{ \big(s, f(s)\big) \;\Big|\; s \in S_1 \text{ reachable} \Big\}$$

- conversely: bisimulation $R \subseteq S_1 \times S_2 \rightsquigarrow$ span of open maps

$$S_1 \longleftarrow R \longrightarrow S_2$$

Open maps and paths

- a path transition system:

$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \longrightarrow \dots \xrightarrow{a_n} s_n$$

- **P** : the category of paths and inclusion morphisms
- (a *full* subcategory of transition systems)

Open maps and paths

- a path transition system:

$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \longrightarrow \dots \xrightarrow{a_n} s_n$$

- $\mathbf{P}$: the category of paths and inclusion morphisms
- a path in a transition system $T \mathrel{\hat{=}}$ a morphism $P \to T$, for $P \in \mathbf{P}$

Open maps and paths

- a path transition system:

$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \longrightarrow \dots \xrightarrow{a_n} s_n$$

- **P** : the category of paths and inclusion morphisms
- a path in a transition system $T \mathrel{\hat{=}}$ a morphism $P \to T$, for $P \in \mathbf{P}$
- a morphism $f : T_1 \to T_2$ is open if and only if:

Open maps and paths

- a path transition system:

$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \longrightarrow \dots \xrightarrow{a_n} s_n$$

- $\mathbf{P}$: the category of paths and inclusion morphisms
- a path in a transition system $T \mathrel{\hat{=}}$ a morphism $P \to T$, for $P \in \mathbf{P}$
- a morphism $f : T_1 \to T_2$ is open if and only if:

$$\forall\ m : P_1 \to P_2 \in \mathbf{P}$$

$$\begin{array}{ccc} P_1 & & T_1 \\ {\scriptstyle m}\downarrow & & \downarrow{\scriptstyle f} \\ P_2 & & T_2 \end{array}$$

Open maps and paths

- a path transition system:

$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \longrightarrow \dots \xrightarrow{a_n} s_n$$

- $\mathbf{P}$: the category of paths and inclusion morphisms
- a path in a transition system $T \mathrel{\hat{=}}$ a morphism $P \to T$, for $P \in \mathbf{P}$
- a morphism $f : T_1 \to T_2$ is open if and only if:

$\forall\ m : P_1 \to P_2 \in \mathbf{P}$

$\forall\ p_1 : P_1 \to T_1,\ p_2 : P_2 \to T_2$
with $p_2 \circ m = f \circ p_1$

$$\begin{array}{ccc} P_1 & \xrightarrow{p_1} & T_1 \\ {\scriptstyle m}\downarrow & & \downarrow{\scriptstyle f} \\ P_2 & \xrightarrow[p_2]{} & T_2 \end{array}$$

Open maps and paths

- a path transition system:

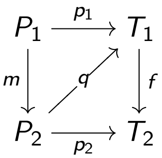
$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \longrightarrow \dots \xrightarrow{a_n} s_n$$

- $\mathbf{P}$: the category of paths and inclusion morphisms
- a path in a transition system $T \triangleq$ a morphism $P \to T$, for $P \in \mathbf{P}$
- a morphism $f : T_1 \to T_2$ is open if and only if:

$\forall\; m : P_1 \to P_2 \in \mathbf{P}$

$\forall\; p_1 : P_1 \to T_1,\; p_2 : P_2 \to T_2$
with $p_2 \circ m = f \circ p_1$

$\exists\; q : P_2 \to T_1$ such that
$q \circ m = p_1$ and $f \circ q = p_2$



Open maps and paths

- a path transition system:

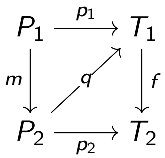
$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \longrightarrow \dots \xrightarrow{a_n} s_n$$

- $\mathbf{P}$: the category of paths and inclusion morphisms
- a path in a transition system $T \mathrel{\hat{=}}$ a morphism $P \to T$, for $P \in \mathbf{P}$
- a morphism $f : T_1 \to T_2$ is open if and only if:

$\forall\ m : P_1 \to P_2 \in \mathbf{P}$

$\forall\ p_1 : P_1 \to T_1,\ p_2 : P_2 \to T_2$
with $p_2 \circ m = f \circ p_1$

$\exists\ q : P_2 \to T_1$ such that
$q \circ m = p_1$ and $f \circ q = p_2$



- a.k.a. open maps $= RLP(\mathbf{P}) = \mathbf{P}^{\square}$

Open maps and paths

- a path transition system:

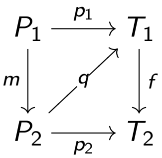
$$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \longrightarrow \dots \xrightarrow{a_n} s_n$$

- $\mathbf{P}$: the category of paths and inclusion morphisms
- a path in a transition system $T \triangleq$ a morphism $P \to T$, for $P \in \mathbf{P}$
- a morphism $f : T_1 \to T_2$ is open if and only if:

  $\forall\ m : P_1 \to P_2 \in \mathbf{P}$

  $\forall\ p_1 : P_1 \to T_1,\ p_2 : P_2 \to T_2$
  with $p_2 \circ m = f \circ p_1$

  $\exists\ q : P_2 \to T_1$ such that
  $q \circ m = p_1$ and $f \circ q = p_2$



- a.k.a. open maps $= RLP(\mathbf{P}) = \mathbf{P}^{\square}$
- generalization to higher-dimensional transition systems: [Fahrenberg: *A category of higher-dimensional automata.* FOSSACS 2005]

Summary

How to introduce and use open maps, "standard" version:

1. Given a category $\mathcal{M}$,
2. identify (usually full) subcategory $\mathbf{P}$ of paths (from denotational semantics, usually),
3. and let open maps be $\mathbf{O} = \mathbf{P}^{\square}$.
4. Then ${}^{\square}\mathbf{O}$ is the colimit closure of $\mathbf{P}$, $({}^{\square}\mathbf{O})^{\square} = \mathbf{O}$, and $({}^{\square}\mathbf{O}, \mathbf{O})$ is a weak factorization system.
5. $\rightsquigarrow$ can introduce model category structures on $\mathcal{M}$; interesting!
6. [Kurz, Rosický: *Weak Factorizations, Fractions and Homotopies.* Applied Categorical Structures 13, 2005]

Summary

How to introduce and use open maps, “standard” version:

1. Given a category $\mathcal{M}$,
2. identify (usually full) subcategory $\mathbf{P}$ of paths (from denotational semantics, usually),
3. and let open maps be $\mathbf{O} = \mathbf{P}^{\square}$.
4. Then ${}^{\square}\mathbf{O}$ is the colimit closure of $\mathbf{P}$, $({}^{\square}\mathbf{O})^{\square} = \mathbf{O}$, and $({}^{\square}\mathbf{O}, \mathbf{O})$ is a weak factorization system.
5. $\rightsquigarrow$ can introduce model category structures on $\mathcal{M}$; interesting!

How to introduce and use open maps, our version:

1. Given a set $\mathcal{M}$ and a (semantics, usually) mapping $\mathcal{M} \to \mathcal{T}$, where $\mathcal{T}$ has open maps,
2. “pull back” open maps to $\mathcal{M}$,
3. and relax conditions on open maps to find morphisms in $\mathcal{M}$.
4. Then the weak factorization system $({}^{\square}\mathbf{O}, ({}^{\square}\mathbf{O})^{\square})$ is interesting,
5. and $({}^{\square}\mathbf{O})^{\square} = \mathbf{O}$ is a useful property to be checked

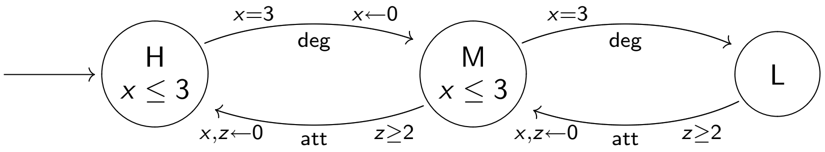
Timed automata

- Finite transition system $(Q, E, q^0, \Sigma_\perp, \ell)$,
- finite set (of clocks) C,
- location invariants $\iota : Q \to \Phi(C)$,
- edge constraints $c : E \to \Phi(C)$,
- and edge reset sets $R : E \to 2^C$.

- $\Phi(C)$: clock constraints:

$$\varphi ::= x \bowtie k \mid x - y \bowtie k \mid \varphi_1 \land \varphi_2 \qquad (x \in C, k \in \mathbb{Z}, \bowtie \in \{\leq, <, \geq, >\})$$

Example:



Semantics

"Standard" version:
Semantics of timed automaton $A = (Q, E, q^0, \Sigma_\perp, \ell, C, \iota, c, R)$ is a timed transition system $[\![A]\!] = (S, E', s^0, \Sigma \cup \mathbb{R}_{\geq 0}, \ell')$ given by

$$S = \{(q, \nu) \in Q \times \mathbb{R}_{\geq 0}^C \mid \nu \vdash \iota(q)\} \qquad s^0 = (q^0, \nu^0)$$
$$E'_s = \{(e, \nu) \in E \times \mathbb{R}_{\geq 0}^C \mid \nu \vdash \iota(\delta_0 e) \wedge c(e), \nu[R(e) \leftarrow 0] \vdash \iota(\delta_1 e)\}$$
$$E'_d = \{(q, \nu, t) \in Q \times \mathbb{R}_{\geq 0}^C \times \mathbb{R}_{\geq 0} \mid \forall t' \in [0, t] : \nu + t' \vdash \iota(q)\}$$

Our version:
Semantics of A is the natural transition system morphism $[\![A]\!] \to A$

Semantics

"Standard" version:

Semantics of timed automaton $A = (Q, E, q^0, \Sigma_\perp, \ell, C, \iota, c, R)$ is a timed transition system $[\![A]\!] = (S, E', s^0, \Sigma \cup \mathbb{R}_{\geq 0}, \ell')$ given by

$$S = \{(q, \nu) \in Q \times \mathbb{R}^C_{\geq 0} \mid \nu \vdash \iota(q)\} \qquad s^0 = (q^0, \nu^0)$$

$$E'_s = \{(e, \nu) \in E \times \mathbb{R}^C_{\geq 0} \mid \nu \vdash \iota(\delta_0 e) \wedge c(e), \nu[R(e) \leftarrow 0] \vdash \iota(\delta_1 e)\}$$

$$E'_d = \{(q, \nu, t) \in Q \times \mathbb{R}^C_{\geq 0} \times \mathbb{R}_{\geq 0} \mid \forall t' \in [0, t] : \nu + t' \vdash \iota(q)\}$$

Our version:

Semantics of A is the natural transition system morphism $[\![A]\!] \to A$

Nothing changed, only emphasized structure: Semantics is now the usual timed transition system with a "backwards" book-keeping mapping

Region quotient

The timed transition systems arising as semantics of timed automata have finite region quotient:

- Given $[\![A]\!]$, say that two valuations ν_1, ν_2 are K-region equivalent ($\simeq_k$), for $K \in \mathbb{N}$, if
  - the integer parts of their clocks are equal,
  - and the fractional orderings of their clocks are equal,
  - or they all exceed K.
- Then $[\![A]\!]/{\simeq_K}$
  - is a "bisimulation quotient" (*i.e.* captures the semantics of A),
  - and is finite.

Observation: Given two timed bisimilar timed automata A, B, then the timed transition system R in $[\![A]\!] \leftarrow R \rightarrow [\![B]\!]$ has the same property.

"Inverse" semantics

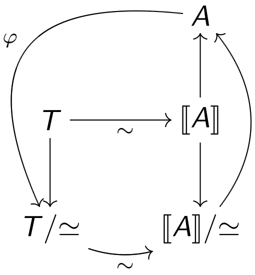
Theorem 6: If T is a timed transition system whose region quotient is a bisimulation quotient, then there is a timed automaton A such that $[\![A]\!]$ and T are isomorphic.

Proof idea: Take the region quotient of T and equip it with constraints and invariants such that locations and transitions are enabled exactly when the valuation is in the region inherent in the location/transition.

"Inverse" semantics

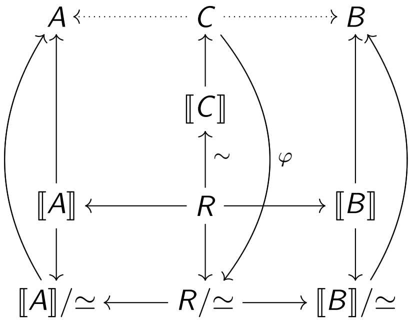
Theorem 6: If T is a timed transition system whose region quotient is a bisimulation quotient, then there is a timed automaton A such that $[\![A]\!]$ and T are isomorphic.

Some book-keeping: If T comes equipped with a book-keeping mapping to a finite transition system (*i.e.* is a "LVTTS" as the timed transition systems arising as semantics of timed automata are), then we can choose the isomorphism so that we have φ below, and the circle is identity:



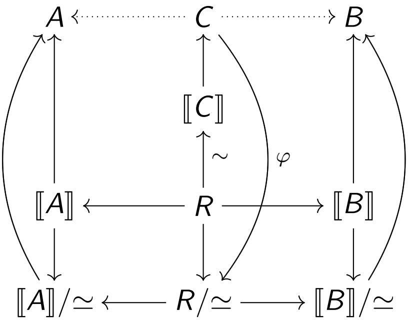
Collecting the pieces

Theorem 10: If A and B are timed automata which are timed bisimilar, then the diagram below defines mappings $A \leftarrow C \rightarrow B$.



Collecting the pieces

Theorem 10: If A and B are timed automata which are timed bisimilar, then the diagram below defines mappings $A \leftarrow C \rightarrow B$.



– and this is what we call open maps.

(Turns out this is the same notion of (morphism and) open map as introduced by Nielsen and Hune in '99 (*Fundam.Inform.* 38), so we must have done *something* right. . .)

Conclusion

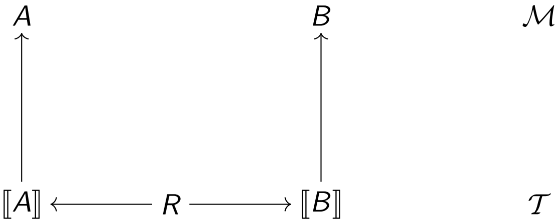
How to pull back open maps along semantics functors:

$$\begin{array}{ccccc} A & & B & & \mathcal{M} \\ \\ [\![A]\!] & \longleftarrow R \longrightarrow & [\![B]\!] & & \mathcal{T} \end{array}$$

Conclusion

How to pull back open maps along semantics functors:

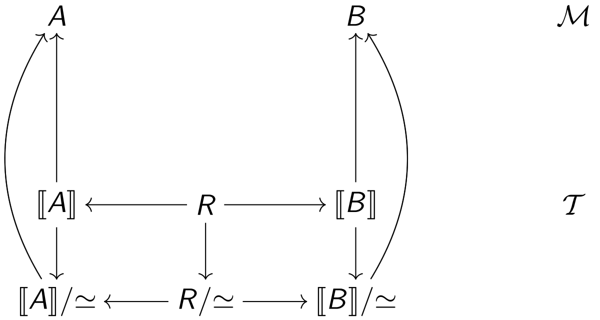
1. View semantics of an object of $\mathcal{M}$ as a morphism into A



Conclusion

How to pull back open maps along semantics functors:

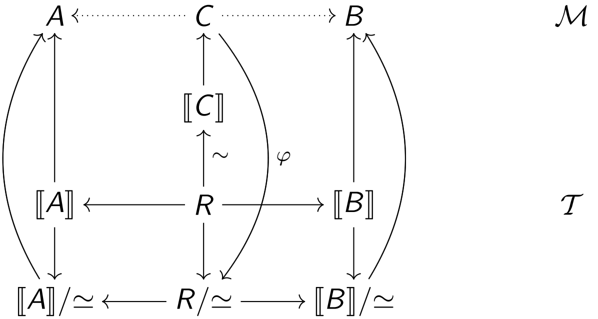
1. View semantics of an object of $\mathcal{M}$ as a morphism into A
2. Identify sufficient conditions for an object in $\mathcal{T}$ to be isomorphic to the semantics of something in $\mathcal{M}$



Conclusion

How to pull back open maps along semantics functors:

1. View semantics of an object of $\mathcal{M}$ as a morphism into A
2. Identify sufficient conditions for an object in $\mathcal{T}$ to be isomorphic to the semantics of something in $\mathcal{M}$
3. Given these conditions, construct an "inverse" to the semantics morphism



Conclusion

How to pull back open maps along semantics functors:

1. View semantics of an object of $\mathcal{M}$ as a morphism into A
2. Identify sufficient conditions for an object in $\mathcal{T}$ to be isomorphic to the semantics of something in $\mathcal{M}$
3. Given these conditions, construct an "inverse" to the semantics morphism

Todo:

- (for timed automata) Check whether $({}^{\square}\mathbf{O})^{\square} = \mathbf{O}$
- (more general) try out Howto for other formalisms