How to Pull Back Open Maps along Semantics Functors

Uli Fahrenberg

Department of Computer Science
Aalborg University
Denmark

ACCAT 2008
1 Motivation
- Example: Bisimulation for timed automata
- Open maps
- Generalization

2 Open maps
- Definition
- Open maps and bisimulation
- Open maps and paths
- Summary

3 Open maps for timed automata
- Definition
- Semantics
- Region quotient
- Open maps
- Conclusion
Example

- timed automata $\leadsto$ 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 $\sim \sim$ operational semantics: transition systems
- transition systems $\sim \sim$ notion of bisimulation
  $\sim \sim$ bisimulation for timed automata
- transition systems $\sim \sim$ 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 \(\sim\) operational semantics: transition systems
- transition systems \(\sim\) notion of bisimulation
  \(\sim\) bisimulation for timed automata
- transition systems \(\sim\) 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

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

bisimulation in $\mathcal{T}$ used for defining bisimulation in $\mathcal{M}$
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
  - \( \Sigma \) 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 \((S_1, s_1^0, \Sigma, E_1), (S_2, s_2^0, \Sigma, E_2)\) : 
\(f : S_1 \rightarrow S_2\) 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 \((S_1, s^0_1, \Sigma, E_1), (S_2, s^0_2, \Sigma, E_2) : f : S_1 \rightarrow S_2\) such that

\[
f(s^0_1) = s^0_2 \\
(s, a, s') \in E_1 \implies (f(s), a, f(s')) \in E_2
\]

\(\leadsto\) category of transition systems

- well-behaved category; natural constructions are well-known; relates to other formalisms by (reflective) functors

Open maps

- transition system: \((S, s^0, \Sigma, E \subseteq S \times \Sigma \times S)\)
- morphism of transition systems \((S_1, s_1^0, \Sigma, E_1), (S_2, s_2^0, \Sigma, E_2) : f : S_1 \rightarrow S_2\) 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 \(f : (S_1, s_1^0, \Sigma, E_1) \rightarrow (S_2, s_2^0, \Sigma, E_2)\) is open if
Open maps

- transition system: \((S, s^0, \Sigma, E \subseteq S \times \Sigma \times S)\)
- morphism of transition systems \((S_1, s_1^0, \Sigma, E_1), (S_2, s_2^0, \Sigma, E_2) : f : S_1 \rightarrow S_2\) 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 \(f : (S_1, s_1^0, \Sigma, E_1) \rightarrow (S_2, s_2^0, \Sigma, E_2)\) is open if
  \[
  \forall \text{ reachable } s_1 \in S_1
  \]
  \[
  S_1
  \]
  \[
  f
  \]
  \[
  S_2
  \]
  \[
  s_1
  \]

\[\]
Open maps

- transition system: \((S, s^0, \Sigma, E \subseteq S \times \Sigma \times S)\)
- morphism of transition systems \((S_1, s^0_1, \Sigma, E_1), (S_2, s^0_2, \Sigma, E_2)\) : 
  \(f : S_1 \rightarrow S_2\) such that
  \[
  f(s^0_1) = s^0_2 \\
  (s, a, s'_1) \in E_1 \implies (f(s), a, f(s')) \in E_2
  \]

\(\leadsto\) category of transition systems

- a morphism \(f : (S_1, s^0_1, \Sigma, E_1) \rightarrow (S_2, s^0_2, \Sigma, E_2)\) is open if
  \[
  \forall \text{ reachable } s_1 \in S_1 \quad \forall \text{ edges } (f(s_1), a, s'_2) \in E_2
  \]

\[
\begin{array}{c}
S_1 \\
\downarrow f \\
S_2
\end{array}
\quad \quad
\begin{array}{c}
s_1 \\
\downarrow f
\end{array}
\quad \quad
\begin{array}{c}
s_1 \\
\downarrow f
\end{array}
\quad \quad
\begin{array}{c}
f(s_1) \\
\stackrel{a}{\rightarrow} s'_2
\end{array}
\]
Open maps

- transition system: \((S, s^0, \Sigma, E \subseteq S \times \Sigma \times S)\)
- morphism of transition systems \((S_1, s^0_1, \Sigma, E_1), (S_2, s^0_2, \Sigma, E_2) : f : S_1 \rightarrow S_2\) such that
  
  \[
  f(s^0_1) = s^0_2
  \]
  \[
  (s, a, s') \in E_1 \implies (f(s), a, f(s')) \in E_2
  \]

\[\implies\] category of transition systems

- a morphism \(f : (S_1, s^0_1, \Sigma, E_1) \rightarrow (S_2, s^0_2, \Sigma, E_2)\) is open if
  
  \[
  \forall \text{reachable } s_1 \in S_1, \exists \text{edge } (s_1, a, s'_1) \in E_1 \text{ for which } s'_2 = f(s'_1)
  \]

\[
S_1 \xrightarrow{f} S_2
\]
\[
S_1 
\xrightarrow{a} s'_1
\]
\[
S_2
\xrightarrow{f(s_1)} s'_2
\]
Open maps and bisimulation

- (again:) a morphism \( f : (S_1, s_1^0, \Sigma E_1) \rightarrow (S_2, s_2^0, \Sigma, E_2) \) is open if
  \[
  \forall \text{ reachable } s_1 \in S_1 \Rightarrow \forall \text{ edges } (f(s_1), a, s'_2) \in E_2 \Rightarrow \exists \text{ edge } (s_1, a, s'_1) \in E_1 \text{ for which } s'_2 = f(s'_1)
  \]
- open map \( f : (S_1, s_1^0, \Sigma, E_1) \rightarrow (S_2, s_2^0, \Sigma, E_2) \) ⇛ bisimulation
  \[
  R = \{(s, f(s)) \mid s \in S_1 \text{ reachable}\}
  \]
- conversely: bisimulation \( R \subseteq S_1 \times S_2 \) ⇛ span of open maps

\[
\begin{array}{c}
\text{S}_1 \\
R
\end{array} \quad \begin{array}{c}
\text{S}_2
\end{array}
\]
Open maps and paths

- a path transition system:
  \[ s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{\ldots} s_n \]

- \( P \): the category of paths and inclusion morphisms

- (a full subcategory of transition systems)
Open maps and paths

- a path transition system:

\[
\begin{align*}
  s_0 & \xrightarrow{a_1} s_1 & \xrightarrow{a_2} s_2 & \cdots & \xrightarrow{a_n} s_n \\
\end{align*}
\]

- \( \mathcal{P} \): the category of paths and inclusion morphisms
- a path in a transition system \( T \) is a morphism \( P \rightarrow T \), for \( P \in \mathcal{P} \)
Open maps and paths

- A path transition system:
  \[ s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \longrightarrow \ldots \xrightarrow{a_n} s_n \]

- \( P \): the category of paths and inclusion morphisms
- A path in a transition system \( T \) is a morphism \( P \rightarrow T \), for \( P \in P \)
- A morphism \( f : T_1 \rightarrow T_2 \) is open if and only if:
  \[ \forall m : P_1 \rightarrow P_2 \in P \quad \forall p_1 : P_1 \rightarrow T_1, p_2 : P_2 \rightarrow T_2 \text{ with } p_2 \circ m = f \circ p_1 \exists q : P_2 \rightarrow T_1 \text{ such that } q \circ m = p_1 \text{ and } f \circ q = p_2 \]
Open maps and paths

- a **path** transition system:

\[ s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{} \ldots \xrightarrow{a_n} s_n \]

- \( P \): the category of paths and inclusion morphisms
- a path in a transition system \( T \) is a morphism \( P \rightarrow T \), for \( P \in P \)
- a morphism \( f : T_1 \rightarrow T_2 \) is open if and only if:

\[
\forall \ m : P_1 \rightarrow P_2 \in P
\]

\[
\begin{array}{ccc}
P_1 & \xrightarrow{m} & P_2 \\
\downarrow & & \downarrow \\
T_1 & \xrightarrow{f} & T_2
\end{array}
\]
Open maps and paths

- A path transition system:
  \[
  s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{} \ldots \xrightarrow{a_n} s_n
  \]

- \( P \): the category of paths and inclusion morphisms

- A path in a transition system \( T \) is a morphism \( P \to T \), for \( P \in P \)

- A morphism \( f : T_1 \to T_2 \) is open if and only if:
  \[
  \forall \ m : P_1 \to P_2 \in P
  \]
  \[
  \forall \ p_1 : P_1 \to T_1, \ p_2 : P_2 \to T_2
  \]
  with \( p_2 \circ m = f \circ p_1 \)
Open maps and paths

- a path transition system:

\[ s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \rightarrow \ldots \xrightarrow{a_n} s_n \]

- \( \mathcal{P} \): the category of paths and inclusion morphisms
- a path in a transition system \( T \) \( \triangleright \) a morphism \( P \rightarrow T \), for \( P \in \mathcal{P} \)
- a morphism \( f : T_1 \rightarrow T_2 \) is open if and only if:

\[
\forall \ m : P_1 \rightarrow P_2 \in \mathcal{P} \\
\forall \ p_1 : P_1 \rightarrow T_1, \ p_2 : P_2 \rightarrow T_2 \\
\text{with } p_2 \circ m = f \circ p_1 \\
\exists \ q : P_2 \rightarrow T_1 \text{ such that } q \circ m = p_1 \text{ and } f \circ q = p_2
\]
Open maps and paths

- a path transition system:

\[ s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \rightarrow \ldots \xrightarrow{a_n} s_n \]

- \( \mathbf{P} \): the category of paths and inclusion morphisms
- a path in a transition system \( T \xleftarrow{\triangle} \) a morphism \( P \rightarrow T \), for \( P \in \mathbf{P} \)
- a morphism \( f : T_1 \rightarrow T_2 \) is open if and only if:

\[
\forall \ m : P_1 \rightarrow P_2 \in \mathbf{P} \\
\forall \ p_1 : P_1 \rightarrow T_1, \ p_2 : P_2 \rightarrow T_2 \text{ with } p_2 \circ m = f \circ p_1 \\
\exists \ q : P_2 \rightarrow T_1 \text{ such that } q \circ m = p_1 \text{ and } f \circ q = p_2
\]

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

...
Open maps and paths

- a path transition system:

  \[ s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{\ldots} s_n \]

- \( P \): the category of paths and inclusion morphisms

- a path in a transition system \( T \) \( \Downarrow \) a morphism \( P \to T \), for \( P \in P \)

- a morphism \( f : T_1 \to T_2 \) is open if and only if:

  \[
  \forall m : P_1 \to P_2 \in P \\
  \forall p_1 : P_1 \to T_1, p_2 : P_2 \to T_2 \\
  \text{with } p_2 \circ m = f \circ p_1 \\
  \exists q : P_2 \to T_1 \text{ such that } q \circ m = p_1 \text{ and } f \circ q = p_2
  \]

- a.k.a. open maps = \( RLP(P) = P^\square \)

- generalization to higher-dimensional transition systems: [Fahrenberg: A category of higher-dimensional automata. FOSSACS 2005]
How to introduce and use open maps, “standard” version:
1. Given a category $\mathcal{M}$,
2. identify (usually full) subcategory $\mathcal{P}$ of paths (from denotational semantics, usually),
3. and let open maps be $\mathcal{O} = \mathcal{P} \Box$.
4. Then $\Box \mathcal{O}$ is the colimit closure of $\mathcal{P}$, $(\Box \mathcal{O}) \Box = \mathcal{O}$, and $(\Box \mathcal{O}, \mathcal{O})$ is a weak factorization system.
5. $\leadsto$ can introduce model category structures on $\mathcal{M}$; interesting!
How to introduce and use open maps, “standard” version:

1. Given a category \( \mathcal{M} \),
2. identify (usually full) subcategory \( \mathcal{P} \) of paths (from denotational semantics, usually),
3. and let open maps be \( \mathcal{O} = \mathcal{P} \Box \).
4. Then \( \Box \mathcal{O} \) is the colimit closure of \( \mathcal{P} \), \( (\Box \mathcal{O}) \Box = \mathcal{O} \), and \( (\Box \mathcal{O}, \mathcal{O}) \) is a weak factorization system.
5. \( \mapsto \) 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 \( (\Box \mathcal{O}, (\Box \mathcal{O}) \Box) \) is interesting,
5. and \( (\Box \mathcal{O}) \Box = \mathcal{O} \) is a useful property to be checked.
Timed automata

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

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

\[ \varphi ::= x \triangleleft k \mid x - y \triangleleft k \mid \varphi_1 \land \varphi_2 \quad (x \in C, k \in \mathbb{Z}, \triangleleft \in \{\leq, <, \geq, >\}) \]

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

$$S = \{(q, \nu) \in Q \times \mathbb{R}_{\geq 0}^C \mid \nu \vdash \nu(q)\} \quad \quad s^0 = (q^0, \nu^0)$$

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

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

Our version:
Semantics of $A$ is the natural transition system morphism $\llbracket A \rrbracket \to A$
Semantics

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

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

$$s^0 = (q^0, \nu^0)$$

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

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

Our version:
Semantics of $A$ is the natural transition system morphism $\semantics{A} \to A$

Nothing changed, only emphasized structure: Semantics is now the usual timed transition system with a “backwards” book-keeping mapping
The timed transition systems arising as semantics of timed automata have finite region quotient:

- Given $\lbrack A \rbrack$, say that two valuations $\nu_1, \nu_2$ are $K$-region equivalent ($\equiv_K$), 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 $\lbrack A \rbrack/\equiv_K$
  - 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 $\lbrack A \rbrack \leftarrow R \rightarrow \lbrack B \rbrack$ 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 isomorph.

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 $\varphi$ below, and the circle is identity:

$$
\begin{array}{c}
\varphi \\
\downarrow \\
T \\
\sim \\
\downarrow \\
T/\sim \\
\sim \\
\downarrow \\
[A]/\sim \\
\end{array}
$$

$$
\begin{array}{c}
A \\
\uparrow \\
\sim \\
\uparrow \\
\downarrow \\
[A] \\
\downarrow \\
\sim \\
\downarrow \\
T/\sim \\
\end{array}
$$
**Theorem 10:** If $A$ and $B$ are timed automata which are timed bisimilar, then the diagram below defines mappings $A \leftarrow C \rightarrow B$.

\[\begin{array}{ccc}
A & \sim & B \\
\downarrow & & \downarrow \\
[A] & \sim & [B] \\
\downarrow & & \downarrow \\
[A]/\sim & \sim & [B]/\sim
\end{array}\]
Collecting the pieces

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

\[ \begin{array}{c}
A \leftarrow \quad C \quad \rightarrow B \\
\downarrow \quad \quad \quad \quad \downarrow \quad \quad \quad \quad \downarrow \quad \quad \quad \quad \downarrow \\
[A] \quad \quad \quad \quad [C] \quad \quad \quad \quad [B] \\
\downarrow \sim \quad \quad \quad \quad \downarrow \sim \quad \quad \quad \quad \downarrow \sim \\
[A]/\sim \quad \rightarrow \quad R/\sim \quad \rightarrow \quad [B]/\sim \\
\end{array} \]

– 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:

$$\begin{align*}
A & \quad B & \quad M \\
\downarrow & \quad \downarrow & \quad \downarrow \\
[A] & \quad R & \quad [B] \\
\downarrow & \quad \downarrow & \quad \downarrow \\
J & \quad A & \quad K \\
\downarrow & \quad \downarrow & \quad \downarrow \\
J & \quad B & \quad K \\
\downarrow & \quad \downarrow & \quad \downarrow \\
J & \quad R & \quad J \\
\downarrow & \quad \downarrow & \quad \downarrow \\
J & \quad B & \quad K \\
\downarrow & \quad \downarrow & \quad \downarrow \\
J & \quad R & \quad J \\
\downarrow & \quad \downarrow & \quad \downarrow \\
J & \quad B & \quad K \\
\end{align*}$$
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 $\mathcal{M}$.

\[
\begin{array}{ccc}
A & \xrightarrow{\sim} & B \\
\downarrow & & \downarrow \\
[A] & \xleftrightarrow{R} & [B] \\
\downarrow & & \downarrow \\
[A]/\sim & \xleftrightarrow{R/\sim} & [B]/\sim
\end{array}
\]
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 $(\square O \square) = O$
- (more general) try out Howto for other formalisms