How to Pull Back Open Maps along Semantics Functors
Extended Abstract

Uli Fahrenberg*

March 23, 2008

This paper is part of an exploration into the following setting: Given a formalism $\mathcal{M}$ for which an operational semantics is defined using transition systems, or some other category $\mathcal{T}$ in which there is a notion of *open maps* in the sense of [2], under which conditions can the notion open map be "pulled back" from $\mathcal{T}$ to $\mathcal{M}$?

This is a quite important problem, as there are plenty of widely-used formalisms which fit the above description, yet for which no notion of open map (or even morphism) has been defined, or has been defined but in a rather *ad-hoc* manner. On the other hand, open maps are a useful tool: they lead to such notions as *presheaf models* and *characteristic path logics*, they aid in comparing different formalisms, and one should also not forget the recently explored relation to *topological model categories* [3].

The problem stated above is quite general, hence it is appropriate to start by considering some examples. In this paper, we are concerned with the particular example where $\mathcal{M}$ is timed automata [1] and $\mathcal{T}$ is (timed) transition systems. This also makes it possible to compare our notions with the ones introduced in [4].

1 Semantics of Timed Automata

It shall be convenient to use models slightly different from the standard exposition which avoid certain *extensionality* properties. Hence,

- a (directed) *graph* is a pair of sets (V, E) together with mappings $\delta_0, \delta_1 : E \to V$, $\varepsilon : V \to E$ (called *face maps* and *degeneracies*, respectively) satisfying $\delta_i \circ \varepsilon = \mathrm{id}$ for $i = 0, 1$. These structure maps will be implicit in what follows. The graph is *finite* if both V and E are finite sets.

*Dept. of Computer Science, Aalborg University, Denmark. Email: uli@cs.aau.dk

Note that this representation of graphs is standard in, *e.g.*, algebraic topology. What follows is a list of standard definitions, included only to make explicit how our different representation of graphs affects them:

- A (labeled) *transition system* consists of a graph (S, E), $s^0 \in S$, a pointed set $\Sigma_\perp$, and a mapping $\ell : E \to \Sigma_\perp$, with the provision that $\ell(e) = \perp$ if and only if $e = \varepsilon\delta_0 e$. The transition system is *finite* if both (S, E) and Σ are finite.

- A *timed transition system* is a transition system $(S, E, s^0, \Sigma \cup \mathbb{R}_{\geq 0}, \ell)$, where $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers, $\Sigma \cap \mathbb{R}_{\geq 0} = \emptyset$, and we let $\perp = 0 \in \mathbb{R}_{\geq 0}$. The following properties are required, where we customarily denote by $s \xrightarrow{t} s'$ the property that there exists $e \in E$ with $\delta_0 e = s$, $\delta_1 e = s'$, and $\ell(e) = t \in \mathbb{R}_{\geq 0}$:

  1. whenever $s \xrightarrow{t} s'$ and $s' \xrightarrow{t'} s''$, then also $s \xrightarrow{t+t'} s''$,
  2. whenever $s \xrightarrow{t} s'$ and $t' \leq t$, then also $s \xrightarrow{t'} s'' \xrightarrow{t-t'} s'$ for some $s'' \in S$, and
  3. whenever $s \xrightarrow{t} s'$ and $s \xrightarrow{t} s''$, then $s' = s''$.

  As a shorthand, we shall use $E_s = \ell^{-1}(\Sigma) \subseteq E$ and $E_d = \ell^{-1}(\mathbb{R}_{\geq 0}) \subseteq E$ to denote the sets of *switch* and *delay* transitions, respectively.

- A *timed automaton* consists of a finite transition system $(Q, E, q^0, \Sigma_\perp, \ell)$ and a finite set C, together with mappings $\iota : Q \to \Phi(C)$, $c : E \to \Phi(C)$, and $R : E \to 2^C$. Here the set $\Phi(C)$ of *clock constraints* over C is defined by the grammar

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

  and the mappings are to satisfy the conditions that $c(\varepsilon q) = tt$ and $R(\varepsilon q) = \emptyset$ for all $q \in Q$.

Definition 1 The *semantics* of a timed automaton $A = (Q, E, q^0, \Sigma_\perp, \ell, C, \iota, c, R)$ is given by a timed transition system $[\![A]\!] = (S, E', s^0, \Sigma \cup \mathbb{R}_{\geq 0}, \ell')$ and a transition system morphism $(\sigma, \lambda) : [\![A]\!] \to (Q, E, q^0, \Sigma_\perp, \ell)$, which are defined as follows:

$$S = \left\{(q,\nu) \in Q \times \mathbb{R}^C_{\geq 0} \;\middle|\; \nu \vdash \iota(q)\right\} \qquad s^0 = (q^0, \nu^0)$$
$$E'_s = \left\{(e,\nu) \in E \times \mathbb{R}^C_{\geq 0} \;\middle|\; \nu \vdash \iota(\delta_0 e) \wedge c(e), \nu[R(e) \leftarrow 0] \vdash \iota(\delta_1 e)\right\}$$
$$E'_d = \left\{(q,\nu,t) \in Q \times \mathbb{R}^C_{\geq 0} \times \mathbb{R}_{\geq 0} \;\middle|\; \forall t' \in [0,t] : \nu + t' \vdash \iota(q)\right\}$$
$$\delta_0(e,\nu) = (\delta_0 e, \nu) \qquad \delta_1(e,\nu) = (\delta_1 e, \nu[R(e) \leftarrow 0]) \qquad \ell(e,\nu) = \ell(e)$$
$$\delta_0(q,\nu,t) = (q,\nu) \qquad \delta_1(q,\nu,t) = (q,\nu + t) \qquad \ell(q,\nu,t) = t$$
$$\varepsilon(q,\nu) = (q,\nu,0) \qquad \sigma(q,\nu) = q \qquad \sigma(e,\nu) = e$$
$$\sigma(q,\nu,t) = \varepsilon q \qquad \lambda(x) = \begin{cases} x & \text{if } x \in \Sigma \\ \bot & \text{if } x \in \mathbb{R}_{\geq 0} \end{cases}$$

Note how, except for our "book-keeping" mapping (σ, λ) and the mentioned differences regarding presentation of graphs, this is just the standard definition from [1].

2 From Timed Transition Systems to Timed Automata

The timed transition systems which occur as the semantics of timed automata have a special structure which we need to make explicit:

Definition 2 A *location-valuation timed transition system* (LVTTS) consists of a timed transition system $T = (S, E, s^0, \Sigma \cup \mathbb{R}_{\geq 0}, \ell)$, a finite transition system $T' = (Q, E', q^0, \Sigma_\bot, \ell')$, and a finite set C, together with a transition system morphism $(\sigma, \lambda) : T \to T'$ for which

$$\lambda(x) = \begin{cases} x & \text{if } x \in \Sigma \\ \bot & \text{if } x \in \mathbb{R}_{\geq 0} \end{cases}$$

and a mapping $\rho : S \to \mathbb{R}^C_{\geq 0}$ for which $\rho(s^0) = \nu^0$. Additionally, the two following properties are required:

1. For all $s \xrightarrow{a} s' \in E_s$ and all $c \in C$: $\rho(s')(c) = \rho(s)(c)$ or $\rho(s')(c) = 0$.
2. For all $s \xrightarrow{t} s' \in E_d$: $\rho(s') = \rho(s) + t$.

The next definition uses the notion of K-region equivalence $\simeq_K \subseteq \mathbb{R}^C_{\geq 0} \times \mathbb{R}^C_{\geq 0}$ from [1]:

Definition 3 A LVTTS $(S, E, s^0, \Sigma \cup \mathbb{R}_{\geq 0}, \ell, Q, E', q^0, \ell', \sigma, \lambda, C, \rho)$ is said to be *K-region stable*, for $K \in \mathbb{N}$, provided that

1. for all $s_1 \in S$ and all $\nu_2 \simeq_K \rho(s_1)$, there exists $s_2 \in S$ for which $\sigma(s_2) = \sigma(s_1)$ and $\rho(s_2) = \nu_2$, and

2. for all $e_1 \in E$ and all $\nu_2 \simeq_K \rho(\delta_0 e)$, there exists $e_2 \in E$ for which $\sigma(e_2) = \sigma(e_1)$, $\rho(\delta_0 e_2) = \nu_2$, and $\rho(\delta_1 e_1) \simeq_K \rho(\delta_1 e_2)$.

It is well-known [1] that $[\![A]\!]$ is K-region stable for some K, for any timed automaton A. Theorem 6 below provides a converse.

The next definition adapts standard quotient constructions from [1] for our purpose; the only difference is that we do not identify states or transitions in the quotient unless their images by σ are equal. Hence our quotients are bigger than the ones in [1], but they are still *finite*.

Definition 4 The *K-region quotient* of a K-region stable LVTTS $T = (S, E, s^0, \Sigma \cup \mathbb{R}_{\geq 0}, \ell, Q, E', q^0, \ell', \sigma, \lambda, C, \rho)$ is the transition system $\big(S', E'', \bar{s}^0, \Sigma_\perp \cup \{\tau\}, \ell''\big)$ given as follows: Extend $\simeq_K$ to S and E by

$$\begin{aligned} s \simeq_K s' \quad &\text{iff} \quad \sigma(s) = \sigma(s') \text{ and } \rho(s) \simeq_K \rho(s') \\ e \simeq_K e' \quad &\text{iff} \quad \sigma(e) = \sigma(e'), \delta_0 e \simeq_K \delta_0 e', \text{ and } \delta_1 e \simeq_K \delta_1 e' \end{aligned}$$

and let

$$S' = S/{\simeq_K} \qquad E'' = E/{\simeq_K} \qquad \bar{s}^0 = \langle s^0 \rangle \qquad \varepsilon\langle s \rangle = \langle \varepsilon s \rangle$$

$$\delta_0\langle e \rangle = \langle \delta_0 e \rangle \qquad \delta_1\langle e \rangle = \langle \delta_1 e \rangle \qquad \ell''\langle e \rangle = \begin{cases} \ell(e) & \text{if } \ell(e) \in \Sigma \\ \perp & \text{if } e \simeq_K \varepsilon \delta_0 e \\ \tau & \text{else} \end{cases}$$

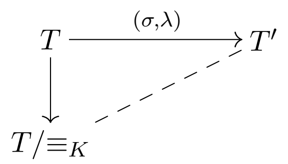
The *delay K-quotient* of T is the transition system $T/{\equiv_K} = \big(S'', E''', \bar{\bar{s}}^0, \Sigma_\perp \cup \{\tau\}, \ell'''\big)$ given as follows: Define an equivalence relation $\equiv \subseteq S' \times S'$, and extend it (minimally) to E'', by

$$\begin{aligned} s \equiv s' \quad &\text{iff} \quad s = s' \text{ or } s \xrightarrow{\varepsilon} s' \text{ or } s' \xrightarrow{\varepsilon} s \\ e \equiv e' \quad &\text{iff} \quad e = e', \text{ or } e = \varepsilon \delta_0 e, e' = \varepsilon \delta_0 e', \text{ and } \delta_0 e \equiv \delta_0 e' \end{aligned}$$

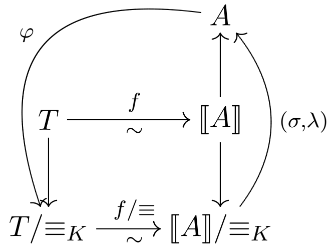
and let

$$S'' = S'/{\equiv} \qquad E''' = E''/{\equiv} \qquad \bar{\bar{s}}^0 = \langle \bar{s}^0 \rangle \qquad \varepsilon\langle s \rangle = \langle \varepsilon s \rangle$$
$$\delta_0\langle e \rangle = \langle \delta_0 e \rangle \qquad \delta_1\langle e \rangle = \langle \delta_1 e \rangle \qquad \ell'''\langle e \rangle = \langle \ell''(e) \rangle$$

Lemma 5 *For any K-region stable LVTTS $(\sigma, \lambda) : T \to T'$, the morphism (σ, λ) lifts to the quotient $T/{\equiv_K}$:*



Theorem 6 *Let T be a K-region stable LVTTS. There exists a timed automaton A, an isomorphism $f : T \to [\![A]\!]$, and a mapping $\varphi : A \to T/{\equiv_K}$ such that in the diagram below, $\varphi \circ (\sigma, \lambda) \circ f/{\equiv} = \mathrm{id}$:*

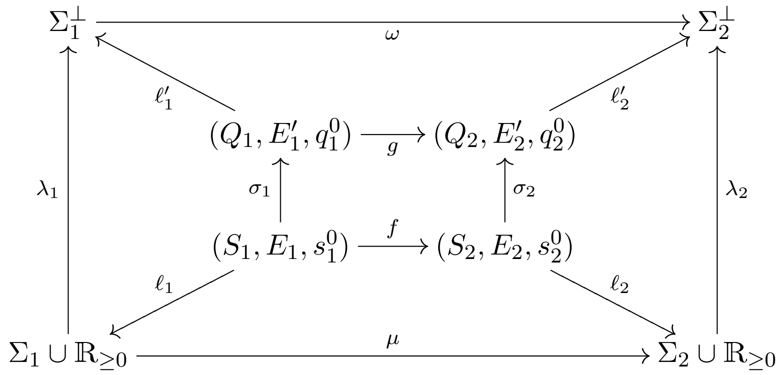


Proof Idea: The underlying graph of A is $T/{\equiv_K}$. This is then equipped with invariants ι, constraints c and reset sets R to ensure that the semantics is as required.

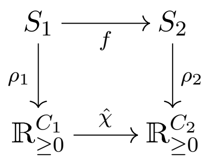
3 Open Maps for Timed Automata

The main part of our work is now done; we only need to collect the pieces:

Definition 7 A *morphism* of LVTTS consists of transition system morphisms (f, μ), (g, ω) as in the commutative diagram



with the property that $\mu(a) \in \Sigma_2 \cup \{0\}$ for all $a \in \Sigma_1$, together with a mapping $\chi : C_2 \to C_1$ such that



commutes, where $\hat{\chi}$ is the induced mapping.

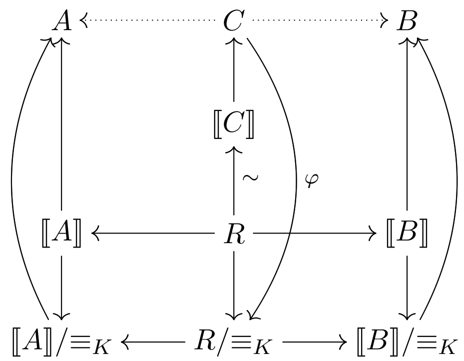
Definition 8 A label-preserving morphism (f, g) of LVTTS is said to be *open* if both f and g are open.

Timed bisimilarity of LVTTS $T_1 \to T_1'$, $T_2 \to T_2'$ is defined as timed bisimilarity of the timed transition systems T_1 and T_2, *i.e.* with no special demands on the "book-keeping" transition systems T_1', T_2'.

Lemma 9 *Two LVTTS T_1, T_2 are timed bisimilar if and only if there is an LVTTS T_3 and a span of open maps $T_1 \leftarrow T_3 \to T_2$.*

If T_1 has finite K_1-region quotient and T_2 has finite K_2-region quotient, then T_3 has $\max(K_1, K_2)$-region quotient, and the open maps pass to open maps of the delay $\max(K_1, K_2)$-quotients.

Theorem 10 *If A and B are timed automata which are timed bisimilar, then the diagram below defines mappings $A \leftarrow C \to B$. These are morphisms of timed automata in the sense of [4].*



Whether the concept of open map suggested by the above theorem agrees with the one from [4] remains to be seen.

Proof Idea: $[\![A]\!]$ is K_1-region stable for some K_1, and $[\![B]\!]$ is K_2-region stable for some K_2. By Lemma 9, R is K-region stable for $K = \max(K_1, K_2)$, and we have open maps of delay K-quotients. An application of Theorem 6 gives the timed automaton C together with the mapping φ.

4 Conclusion

We have arrived at a notion of morphism and open map for timed automata which appears to resemble the one of [4]. However our path to arrive at this notion is very different, which is why the resemblance is a Good Thing.

The usual way to introduce open maps starts by defining a notion of *path category* and then lets those morphisms be open which have the right-lifting property with respect to this path category. Here we have taken a different approach, by working our way up from the category in which the *semantics* of timed automata is living.

The way this is accomplished is by refining the semantics by introducing some "book-keeping" mappings. This approach is categorical in nature, in that the semantics now is a *morphism* instead of an object.

We hope that our approach can be applied to other formalisms, where there is no immediate notion of path category, yet notions of bisimilarity and open maps are available in the semantics. We notice that it appears difficult to recover a path category once one knows what open maps should look like; however it is shown in [3] that the role of the path category can be taken over by its *colimit closure*, which is the same as the category of left-lifting maps with respect to the open maps, hence can be obtained from these.

References

[1] Rajeev Alur and David Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[2] André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164–185, 1996.

[3] Alexander Kurz and Jiří Rosický. Weak factorizations, fractions and homotopies. *Applied Categorical Structures*, 13(2):141–160, 2005.

[4] Mogens Nielsen and Thomas Hune. Bisimulation and open maps for timed transition systems. *Fundam.Inform.*, 38(1-2):61–77, 1999.