Real time automata and the Kleene algebra of sets of real numbers

Cătălin Dima,
University of Bucharest.

Contents:

- Signals, real-time automata and their regular expressions;
- The problem of complementing real-time automata;
- The Kleene algebra of sets of real numbers, normal forms, matrices;
- Complementation of real-time automata.
- Relationship to Simple Duration Calculus.
- Expressivity of real-time automata.
- Event-clock temporal logic with "periodic" constraints.
- Synchronizations of real-time automata.
- (Ideas for) Kleene theorems for automata with timing constraints.

Signal over Σ: function $\sigma : [0, e) \longrightarrow \Sigma$, having finitely many discontinuities, all of them being left discontinuities.

$Sig(\Sigma)$ = the set of signals over Σ, subsets of it = **real-time languages**.

Concatenation: For $\sigma_1, \sigma_2 \in Sig(\Sigma)$, $dom(\sigma_i) = [0, e_i)$ $(i = 1, 2)$

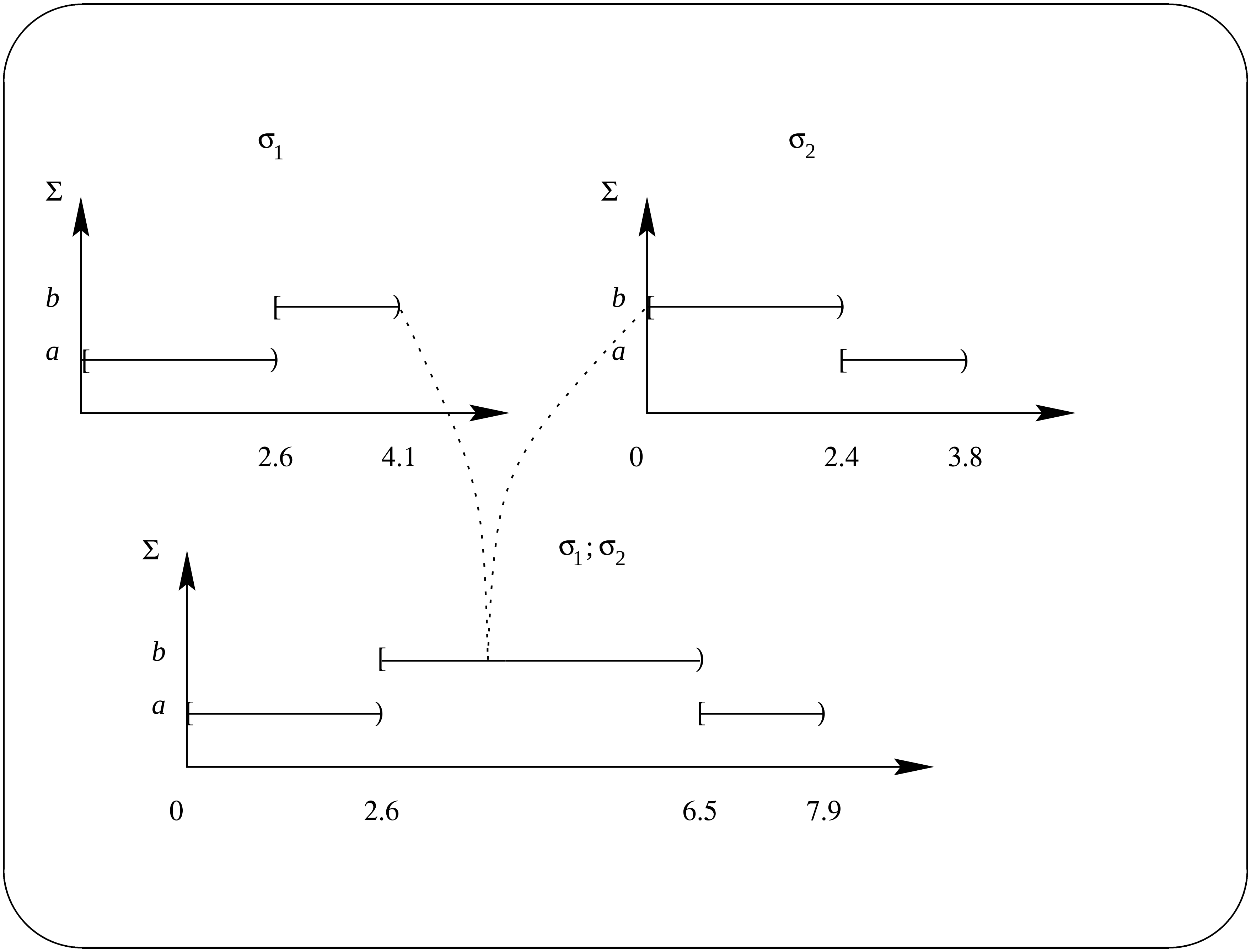
$$(\sigma_1; \sigma_2)(t) = \begin{cases} \sigma_1(t), & t \in [0, e_1), \\ \sigma_2(t - e_1), & t \in [e_1, e_1 + e_2) \end{cases}$$

$\sigma_\epsilon : [0, 0) \longrightarrow \Sigma$ is the unit of concatenation.

Concatenation and star on RT languages:

$L_1; L_2 = \{\sigma_1; \sigma_2 \mid \sigma_i \in L_i\}$

$L^* = \bigcup_{n \in \mathbb{N}} L^n$, where $L^0 = \{\sigma_\epsilon\}$, $L^{n+1} = L^n; L$.


σ1
σ2
Σ
b
a
2.6
4.1
0
2.4
3.8
σ1; σ2
0
2.6
6.5
7.9

Real-time automaton (RTA): $\mathcal{A} = (Q, \Sigma, \lambda, \iota, \delta, Q_0, F)$ where
Q = the (finite) set of *states*, $\delta \subseteq Q \times Q$ = the *transition relation*,
$Q_0 \subseteq Q$ = the set of *initial* states, $F \subseteq Q$ = the set of *final* states,
$\lambda : Q \longrightarrow \Sigma$ = the *state labeling function*,
$\iota : Q \longrightarrow Int_{\mathbb{Q}}^{\geq 0}$ = the *interval* (or *time*) *labeling function.*

$Int_{\mathbb{Q}}^{\geq 0}$ = the set of intervals with positive rational bounds.

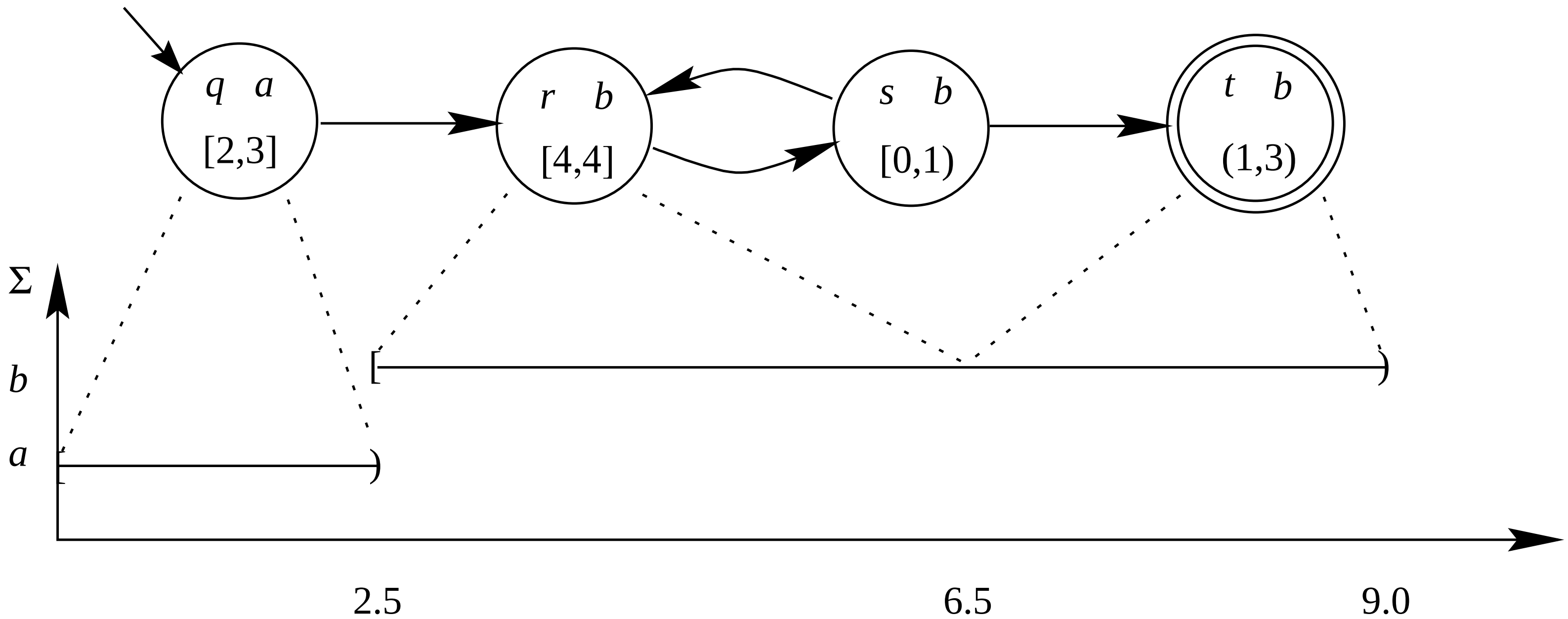
Run: sequence of states $(q_i)_{i\in[n]}$ with $(q_{i-1}, q_i) \in \delta, \forall i \in [n]$.
Accepting run: starts in Q_0 and ends in F.

A run is **associated to** a signal σ with $dom(\sigma) = [0, e)$ iff there exist $0 = e_1 \leq \ldots \leq e_{n+1} = e$ s.t. $e_{i+1} - e_i \in \iota(q_i)$ and $\sigma(t) = \lambda(q_i)$ for all $t \in [e_i, e_{i+1})$ and all $i \in [n]$.

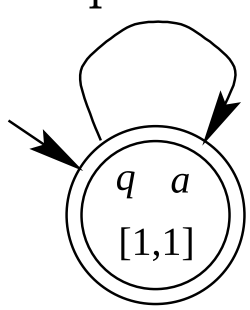
A signal is **accepted** by $\mathcal{A}$ iff it is associated to an accepting run.

The **language** of $\mathcal{A} = L(\mathcal{A})$ = the set of signals accepted by it.

Example 1:



Example 2:



$$L(\mathcal{A}) = \{\sigma : [0,n) \longrightarrow \{a\} \mid n \in \mathbb{N}, n \geq 1\}$$

ETRE(Σ) = **elementary timed regular expressions** over Σ:

$$E ::= 0 \mid [a]_I \mid E + E \mid E; E \mid E^*$$

where the *atoms* $[a]_I$ are all expressions with $a \in \Sigma$ and $I \in Int_{\mathbb{Q}}^{\geq 0}$.

Semantics of ETRE:

$$|[a]_I| = \{\sigma \in Sig(\Sigma) \mid dom(\sigma) = [0, e), e \in I \text{ and } \forall t \in [0, e) \;\; \sigma(t) = a\}$$

$$\begin{array}{rclcrcl} |E+F| & = & |E| \cup |F| & \quad & |E;F| & = & |E|;|F| \\ |E^*| & = & |E|^* & & |0| & = & \emptyset \end{array}$$

Remark: $|[a]_{[0,0]}| = \{\sigma_\epsilon\}$.

A Kleene theorem: ETRE and RTA are equally expressive.

Example: $L(\mathcal{A}) = ([a]_{[1,1]})^*$

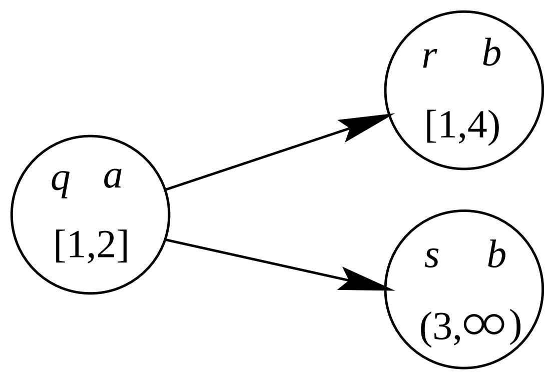
An RTA $\mathcal{A}$ is called

- **language deterministic** iff each signal in $L(\mathcal{A})$ is associated to a unique run that starts in an initial state (but might not end in a final one!).
- **stuttering-free** iff it does not have time labels which contain 0 and transitions (q, r) with $\lambda(q) = \lambda(r)$.
- **state-deterministic** iff whenever $r \neq s$ and either $r, s \in Q_0$ or $(q, r), (q, s) \in \delta$ then $\lambda(r) \neq \lambda(s)$ or $\iota(r) \cap \iota(s) = \emptyset$.
- **deterministic** iff it is both state-deterministic and stuttering-free.

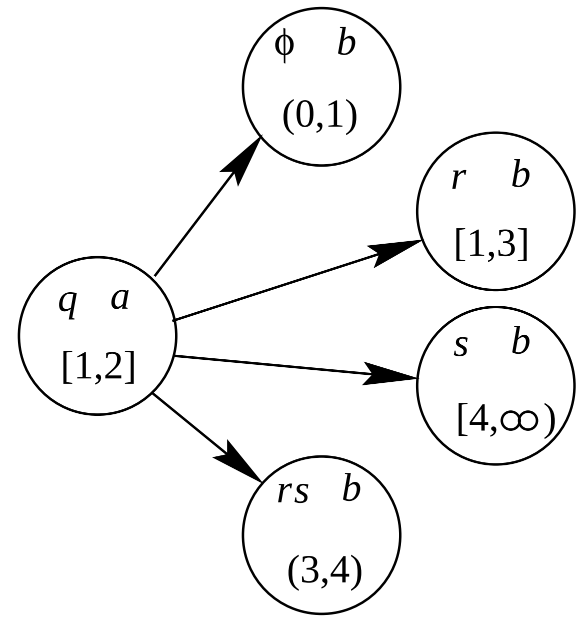
Proposition: Each stuttering-free RTA can be transformed into a deterministic RTA.

The subset construction (with some adaptation) works for stuttering free RTA.

Intuitively



original

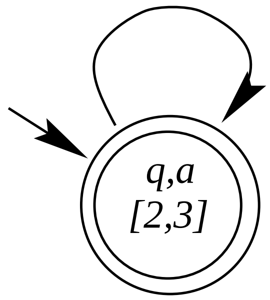


determinized

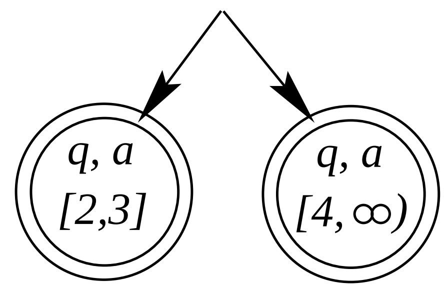
If they are n "target" states in the original RTA (i.e. $(q, q_i) \in \delta$ for $i \in [n]$), then in the determinized they would be 2^n target states $S \subseteq \{q_1, \ldots q_n\}$ with $\iota(S) = \bigcap_{r \in S} \iota(r) \cap \bigcap_{r \notin S} \overline{\iota(r)}$

where $\overline{\iota(r)} = (0, \infty) \setminus \iota(r)$.

Is stuttering freeness achievable?



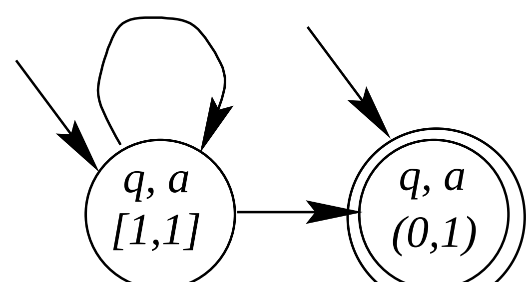
can be transformed into



Intuitively, $[2,3]^* = \{0\} \cup [2,3] \cup [4,6] \cup [6,9] \cup \ldots = \{0\} \cup [2,3] \cup [4,\infty)$

But

- $L(\mathcal{A}) = \{\sigma : [0,n) \longrightarrow \{a\} \mid n \in \mathbb{N}, n \geq 1\}$ cannot be accepted by a stuttering free RTA, ($\{1\}^*$ is not a finite union of intervals) **though** $\mathcal{A}$ is a language deterministic RTA.
- The complement of $L(\mathcal{A})$ is accepted by



Need a broader notion of determinism.

A Kleene algebra of sets of reals: $\mathcal{P}(\mathbb{R}_{\geq 0}) = (\mathcal{P}(\mathbb{R}_{\geq 0}), \cup, +, \cdot^*, \emptyset, \mathbf{0})$

- $\cup$ is union with unit $\emptyset$;
- summation is defined pointwise $X + Y = \{x + y \mid x \in X, y \in Y\}$ and has unit $\mathbf{0} = \{0\}$;
- repetition is defined by $X^* = \bigcup_{n \in \mathbb{N}} nX$, where the *multiples* of X are $0X = \mathbf{0}$ and $(n+1)X = nX + X$.

Interested just in the subalgebra generated by intervals which have *rational bounds* $\mathcal{K}(Int_{\mathbb{Q}}^{\geq 0})$.

Scope: modelling stuttering removal: $+$ is used when "fusing" two states and * when removing circuits (those that can be removed).

This means we build regular expressions over the intervals and then try to transform the results into a simpler form - a **normal form**.

Some stars might not disappear as e.g. $\{1\}^*$.

$X \in \mathcal{K}(Int_{\mathbb{Q}}^{\geq 0})$ can be written in **normal form** (NF) iff

$$X = X_1 \cup (X_2 + \{k\}^*)$$

where k is some positive rational and X_1, X_2 are *finite unions of rational intervals* for which there exists some $N \in \mathbb{N}$ such that $X_1 \subseteq [0, Nk)$ and $X_2 \subseteq [Nk, (N+1)k)$. N = the **bound**.

- $((2,3) \cup (4,6)) \cup (([6,7) \cup (8,9)) + \{3\}^*)$ is written in NF with bound 2: $X_1 = (2,3) \cup (4,6) \subseteq [0,6)$, $X_2 = [6,7) \cup (8,9) \subseteq [6,9)$.
- $((2,3) \cup (4,7)) \cup ([5,7] + \{3\}^*)$ is not in NF: there is no k such that $[5,7] \subseteq [3k, 3(k+1))$.

But both expand to the same set:

$$(2,3) \cup (4,7] \cup \bigcup_{n \geq 2} [3n-1, 3n+1]$$

Theorem: NFs are closed under $\cup$, $+$, * and *complementation.*

Proof: Based upon properties of interger division and upon archimedianity of $\mathbb{R}$.

Closure under star:

- If one of X_1, X_2 contains a nonsingular interval then X^* is a finite union of intervals and hence in NF with $k = 0$.

  (Eventually the multiples of the nonsingular interval with overlap, giving an infinite interval).

  E.g. $[2, 3]^* = \{0\} \cup [2, 3] \cup [4, \infty)$.

- Otherwise X^* is a ultimately periodic set whose ratio is the gcd of the set $X_1 \cup X_2 \cup \{k\}$.

Why proving closure under complementation too? Because in determinization we have to compute complements of interval labels.

Matrices of NFs:

$$
\begin{aligned}
(A \cup B)_{ij} &= A_{ij} \cup B_{ij} \\
(A + B)_{ij} &= \bigcup_{k=1}^{n} (A_{ik} + B_{kj}) \\
A^* &= \bigcup_{n \in \mathbb{N}} nA
\end{aligned}
$$

Given a matrix A of NFs, define:

$$
\begin{aligned}
A(0) &= A \\
A(k)_{ij} &= A(k-1)_{ij} \cup (A(k-1)_{ik} + (A(k-1)_{kk})^* + A(k-1)_{kj})
\end{aligned}
$$

Then $A^* = A(n)$.

Proposition: Matrices of NFs are closed under $\cup$, $+$ and * (i.e. the components of the result can be put into NF).

The idea of removing stuttering steps: associate to some RTA $\mathcal{A}$ and each symbol a a matrix A_a of *a-transitions*.

- The size of the matrix equals to $card(\delta) + 2$.
- Transitions are numbered: $\delta = \{t_1, \ldots, t_p\}$, $t_i = (in_i, out_i)$.
- If $out_i = in_j$ and $\lambda(out_i) = a$ then $(A_a)_{ij} = \iota(out_i) = \iota(in_j)$,
- Otherwise $(A_a)_{ij} = \emptyset$.
- Two more rows and columns are added:

$$\begin{array}{lll}(A_a)_{p+1,j} = X & \text{iff} & in_j \in Q_0, \lambda(in_j) = a \text{ and } \iota(in_j) = X \\ (A_a)_{i,p+2} = X & \text{iff} & out_i \in F, \lambda(out_i) = a \text{ and } \iota(out_i) = X\end{array}$$

Then for any $i, j \in [p+2]$, $(A_a^*)_{ij}$ is the union of the durations of all runs which start in out_i, end in in_j and have only a-transitions.

A_a^* will be used to label the states of an RTA in which there are no more stuttering steps involving symbol a.

An **augmented real-time automaton** (ARTA for short) is a tuple $\mathcal{A} = (Q, \Sigma, \delta, \lambda, \iota, Q_0, F)$ where Q, Q_0, F, δ and λ are the same as in RTA while $\iota : Q \longrightarrow \mathcal{K}(Int_{\mathbb{Q}}^{\geq 0})$ (actually ι gives a NF).

Kleene Theorem for ARTA: ARTA and ETRE (and therefore RTA) have the same expressive power.

Based on the observation that

$$|[a]_{(a,b)\cup((c,d)+\{k\}^*)}| = |[a]_{(a,b)} + [a]_{(c,d)}; ([a]_{\{k\}})^*|$$

The time labels in deterministic ARTA define some stuttering pattern for RTA which is harmless to complementation of RTA since it assures language determinism.

Theorem: ARTA have the same expressive power as stuttering-free ARTA.

To remove stuttering steps from ARTA we recursively remove stuttering steps involving each symbol in Σ.

For each a we construct A_a as shown. The non-a-states will be kept unchanged, the a-states will be replaced by new ones. An a-state is added for each nonempty entry in A_a^*.

- $Q' = \{q \in Q \mid \lambda(q) \neq a\} \cup \{(i,j) \mid (A_a^*)_{ij} \neq \emptyset\}$
- $(q,r) \in \delta'$ iff $(q,r) \in \delta$.
- $(q,(i,j)) \in \delta'$ iff $q = in_i$ and $((i,j),r) \in \delta'$ iff $out_j = r$.
- $\lambda'((i,j)) = a$, $\iota'((i,j)) = (A_a^*)_{ij}$, and $\lambda'(q) = \lambda(q)$, $\iota'(q) = \iota(q)$.
- $q \in Q_0'$ iff $q \in Q_0$ and $q \in F'$ iff $q \in F$.
- $(i,j) \in Q_0'$ iff $i = p+1$ and $(i,j) \in F$ iff $j = p+2$.

The resulting ARTA does not have stuttering steps involving a-states and no stuttering steps involving other symbols are added.

Determinization:

- for each $U \subseteq Q$, denote $Tl(U) = \{\iota(q) \mid q \in U\}$.
- put $\mathcal{T} = \{(S, S', a) \mid S' \subseteq S \text{ and } \lambda(S) = a\}$.
- define $\bar{\iota}((S, S', a)) = \mathbb{R}_{>0} \cap (\bigcap Tl(S')) \cap \neg (\bigcup Tl(S \setminus S'))$.
- put $(S, S', a) \simeq (S, S'', a)$ iff $\bar{\iota}((S, S', a)) = \bar{\iota}((S, S'', a))$.

Then $\mathcal{T}/\simeq$ is the set of states of the deterministic ARTA.

$$\delta_d = \{([S, S', a], [U, U', b]) \mid U = \{q \in Q \mid \exists r \in S' \text{ s.t. } (r, q) \in \delta, \lambda(q) = b\} \\ \text{and } U' \subseteq U\}$$

States $[\emptyset, \emptyset, a]$ play the role of the trap states in dfa.

After getting a deterministic ARTA, simply complement its set of final states.

Theorem: The class of ARTA (and hence of RTA) is closed under complementation.

Simple Duration Calculus: two types of formulas: *state formulas and interval formulas.*

$$\begin{aligned} \phi &::= p \mid \phi \vee \phi \mid \neg\phi \\ \Phi &::= \lceil \phi \rceil \wedge a \prec \ell \prec b \mid \Phi \vee \Phi \mid \Phi ; \Phi \mid \neg\Phi \mid \Phi^* \end{aligned}$$

where $p \in Var$, $\prec$ is $<, \leq, =$ and $a, b \in \mathbb{Q}_{\geq 0} \cup \{\infty\}$, $a \neq \infty$.

Semantics: signals over $Var \longrightarrow \{0,1\}$, i.e. functions $\sigma : [0,e) \longrightarrow [Var \longrightarrow \{0,1\}]$.

State formulas are interpreted at each time point: for each $t \in [0,e)$ we define

- $\sigma(t) \models p$ iff $\sigma(t)(p) = 1$.
- $\sigma(t) \models \phi_1 \vee \phi_2$ iff $\sigma(t) \models \phi_1$ or $\sigma(t) \models \phi_2$.
- $\sigma(t) \models \neg\phi$ iff it does not happen that $\sigma(t) \models \phi$.

Interval formulas are interpreted over the whole signal:

- $\sigma \models \lceil \phi \rceil \wedge a < \ell \leq b$ iff $e \in (a, b]$ and for all $t \in [0, e)$ $\sigma(t) \models \phi$.
- $\sigma \models \Phi_1 ; \Phi_2$ iff there exist two signals $\sigma_1, \sigma_2 \in Sig(Var \longrightarrow \{0, 1\})$ such that $\sigma = \sigma_1 ; \sigma_2$ and $\sigma_1 \models \Phi_1$, $\sigma_2 \models \Phi_2$.
- $\sigma \models \Phi^*$ iff there exist some integer $n \geq 0$ and n signals σ_i such that $\sigma = \sigma_1 ; \sigma_2 ; \ldots ; \sigma_n$ and for all $i \in \{1, \ldots, n\}$, $\sigma_i \models \Phi$.

This includes the case $n = 0$ when Φ can be e.g. $\lceil p \rceil \wedge 0 = l$.

In general, Simple Duration Calculus is undecidable (Zhou, Hansen and Sestoft, STACS'93 paper), hence it does not correspond to the elementary timed regular expressions (or real-time automata).

The fragment which corresponds to ETRE and RTA:

- State formulas that are used are conjunctions of literals (i.e. conjunctions of variables or negations of variables).
- All state formulas within an interval formula *use the same set of literals.*

E.g. $(\lceil p \wedge q \rceil \wedge 2 \leq \ell < 3)\,;(\lceil p \wedge \neg q \rceil \wedge 1 < \ell < 3)$ is ok.

$(\lceil p \rceil \wedge 2 \leq \ell < 3)\,;(\lceil p \wedge \neg q \rceil \wedge 1 < \ell < 3)$ is not ok since the first state formula does not use q or $\neg q$ while the second does.

The correspondence to ETRE: define $Var(\Phi)$ as the set of variables which appear in Φ, say $Var(\Phi) = \{p_1, \ldots p_n\}$. Also denote $p_i^0 = \neg p_i$ and $p_i^1 = p_i$. Then put

$$\Sigma = \{\bigwedge_{i=1}^{n} p_i^{\alpha_i} \mid \alpha_i = 0, 1\} = [Var(\Phi) \longrightarrow \{0, 1\}]$$

Then ETREs over Σ are just encodings of Simple DC formulas and the encoding is bijective.

Star Lemma: If a language L is accepted by an RTA then there exists $N \in \mathbb{N}$ such that each signal σ having at least N discontinuities can be chopped into three signals $\sigma = \sigma_1; \sigma_2; \sigma_3$, such that σ_2 has at least one discontinuity and for any $n \in \mathbb{N}$ we have $\sigma_1; \sigma_2^n; \sigma_3 \in L$.

Timed regular expressions over Σ ($TRE(\Sigma)$)

$$E ::= 0 \mid [A]_I \mid E + E \mid E; E \mid E^*$$

where in atoms $[A]_I$ we have $A \subseteq \Sigma$.

$$|[A]_I| = \{\sigma \mid l(\sigma) \in I \text{ and } \sigma(t) \in A \ \ \forall t \in [0, l(\sigma))\}$$

With the Star Lemma we may prove that $L = |[a, b]_{[1,2]}|$ cannot be accepted by any RTA.

RTA $\simeq$ systems with timing constraints having a single component.

Timed regular expressions are the regular expressions which correspond to Simple Duration Calculus.

Event-clock temporal logic:

$$\phi = p \mid \phi \wedge \phi \mid \neg\phi \mid \phi\,\mathcal{U}\phi \mid \phi\,\mathcal{S}\phi \mid \rhd_X \phi \mid \lhd_X \phi$$

where $X \in \mathcal{K}(Int_{\mathbb{Q}}^{\geq 0})$ and $p \in Var$, the set of variables.

Models: $\sigma : \mathbb{R}_{\geq 0} \longrightarrow [Var \longrightarrow \{0, 1\}]$ for which in each finite interval (a, b) there are only finitely many discontinuities.

- $\sigma(t) \models p$ iff $\sigma(t)(p) = 1$.
- $\sigma(t) \models \phi\,\mathcal{U}\psi$ iff there exists $t' > t$ such that $\sigma(t') \models \psi$ and for all $t'' \in (t, t')$ we have $\sigma(t'') \models \phi$.
- $\sigma(t) \models \phi\,\mathcal{S}\psi$ iff there exists $t' < t$ such that $\sigma(t') \models \psi$ and for all $t'' \in (t'', t)$ we have $\sigma(t'') \models \phi$.
- $\sigma(t) \models \rhd_X \phi$ iff there exists $t' > t$ such that $\sigma(t) \models X$ and for all $t'' \in (t, t')$ we have $\sigma(t'') \not\models \phi$.
- $\sigma(t) \models \lhd_X \phi$ iff there exists $t' < t$ such that $\sigma(t) \models X$ and for all $t'' \in (t', t)$ we have $\sigma(t'') \not\models \phi$.

E.g. $\rhd_{\{1\}^*} p$ says that the first time at which p holds is at an integer distance "from now".

$$\rhd_{\{1\}^*} p = \bigvee_{n \in \mathbb{N}} \rhd_{\{n\}} p$$

This cannot be expressed without "periodic" labels.

The satisfiability problem: does some given formula have at least one model?

The satisfiability problem is decidable, the existing decision procedure (i.e. without "periodic" constraints) being easily adapted to handle "peroidic" constraints.

Given n-sets of symbols $\Sigma_1, \ldots, \Sigma_n$ (not necessarily pairwise disjoint) an **n-signal** is a signal over $\Sigma_1 \times \ldots \times \Sigma_n$.

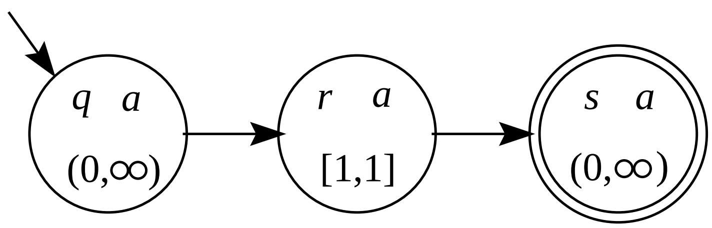
Given n RTA $\mathcal{A}_i = (Q_i, \Sigma_i, \lambda_i, \iota_i, \delta_i, S_i, F_i)$, a **synchronization pattern** on $(\mathcal{A}_i)_{i\in[n]}$ is a set of tuples $\delta \subseteq \left(\prod_{i=1}^n Q^i\right)^2$ such that:

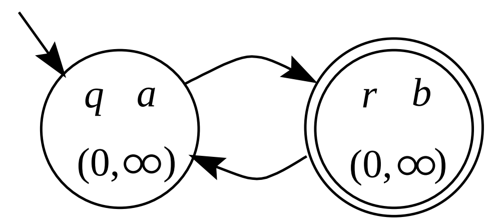
$$((q_1, \ldots, q_n), (r_1, \ldots, r_n)) \in \delta \Rightarrow (q_i, r_i) \in \delta_i \cup Diag(Q_i)$$

Intuitively each RTA works on its own tape but transitions are taken in synchronization with (some of) the others RTA.

Another view is:

An **n-distributed real-time automaton** (n-DRTA) is a tuple $\mathcal{A} = (Q, \Sigma_1, \ldots, \Sigma_n, \delta, \lambda_1, \ldots, \lambda_n, \iota_1, \ldots, \iota_n, S, F)$ where $\delta \subseteq Q \times \mathcal{P}([n]) \times Q$. Moreover tuples $(q, K, r) \in \delta$ have the property that if $\lambda_i(q) \neq \lambda_i(r)$ then $i \in K$. K = the set of "clocks" associated to each component to be reset on the transition.





A synchronization pattern:

$$Diag(Q_1) \times \delta_2 \cup \{((q,q),(r,r)),((r,q),(s,r))\}$$

The accepted language:

$$L \;=\; \{\sigma : [0,e) \longrightarrow \{(a,a),(a,b)\} \mid \exists t \in [0,e) \text{ such that } \sigma(t) = (a,b)$$
$$\sigma(t-0) = (a,a), \sigma(t+1) = (a,b), \sigma(t+1-0) = (a,a)\}$$

It is known (Alur and Dill, *TCS* paper) that the complement of L cannot be accepted by any timed automaton.

DRTA have a decidable emptiness problem but are not closed under complementation (which in fact is undecidable).

Stuttering-free DRTA: for each transition $(q, K, r) \in \delta$, K is the set of indices i such that $\lambda_i(q) \neq \lambda_i(r)$.

Stuttering-free synchronization patterns: each RTA is stuttering-free.

Stuttering-free DRTA are determinizable (and hence have a decidable emptiness problem).

DRTA are less expressive than timed automata and stuttering-free DRTA are less expressive than event-clock automata **but**:

- The emptiness problem for timed automata is reducible to the emptiness problem for DRTA.
- The emptiness problem for event-clock automata is reducible to the emptiness problem for stuttering-free DRTA.

Distributed timed regular expressions over n sets of symbols $\Sigma_1, \ldots, \Sigma_n$ ($DTRE(\Sigma_1, \ldots, \Sigma_n)$):

$$E ::= [a,k]_X \mid E + E \mid E\&E \mid E;E \mid E^*$$

where $k \in \{1, \ldots, n\}$, $a \in \Sigma_k$ and $X \in KInt$.

$$\begin{aligned} |[a,k]_X| \quad = \quad & \{\sigma : [0,e) \longrightarrow \Sigma_1 \times \ldots \times \Sigma_n \mid e \in X \text{ and} \\ & \text{for all } t \in [0,e)\ \pi_k^n(\sigma(t)) = a\} \end{aligned}$$

where π_k^n is the k-th projection of a tuple of n components.

Half of a Kleene theorem: $DRTE \subseteq DRTA$.

Stuttering-free DTRA: by projecting the expression onto each component we get a ETRE which is associated to a RTA (by the Kleene theorem for RTA) in which the only stuttering steps are between states whose time label is $(0, \infty)$.

That is, the associated RTA is obtained by applying only the Kleene theorem and not by further removing the stuttering steps.

The idea comes from considering a fragment of Simple Duration Calculus in which state formulas are only literals and the same stuttering freenes condition holds too.

Such formulas are used in giving "logical semantics" to some concurrent programming language with timing constraints. A formula like

$$\neg\left[\mathsf{true}; (\lceil at_\lambda \rceil \wedge 1 \leq \ell \leq 2 \wedge \lceil x = v \rceil) ; \left(\lceil \neg at_\lambda \rceil \wedge \lceil x \neq v' \rceil\right) ; \mathsf{true}\right]$$

says that being at the label λ in the program, having $x = v$ and the statement being executed after at least 1 and at most 2 units, the effect is that the variable x takes value v'.

A Kleene theorem for a class of automata with timing constraints can be obtained as follows:

- Transport all information into states such that transitions be simply pairs of states.
- Define regular expressions whose atoms' labels are exactly the labels that appear in states.
- Assure that the semantics of regular expressions can be given in a compositional manner, i.e. define concatenation on *languages*.

For timed automata (only last goal):

A **clock valuation** is a function $\xi : [0, e) \longrightarrow \mathbb{R}_{\geq 0}$ where $e \in \mathbb{R}_{\geq 0}$, whose graph has finitely many discontinuities, all of them being right discontinuities, and with the property that in between two consecutive dicontinuities it is linearly increasing from an initial zero value.

An n-**clocked signal** is a function $\sigma : [0, e) \longrightarrow \Sigma \times (\mathbb{R}^n_{>0})$ whose first projection is a signal and whose $k+1$-th projection is a clock valuation ($k \in [n]$).

A *partial* concatenation on clock signals: $\sigma_1; \sigma_2$ is undefined when $\pi_i(\sigma_2(t)) \neq 0$ and $\lim_{t \to 0} \sigma_1(end(\sigma_1)) \neq \pi_i(\sigma_2(t))$; otherwise it is:

$$\sigma(t) = \begin{cases} \sigma_1(t), & t \in dom(\sigma_1) \\ \sigma_2(t - end(\sigma_1)), & t \in [end(\sigma_1), end(\sigma_1) + end(\sigma_2)) \end{cases}$$

Convention $\sigma_\epsilon : [0, 0) \longrightarrow \Sigma \times \mathbb{R}^n$ is the unit for concatenation.

A **total** concatenation on sets of n-clocked signals:

$$L_1; L_2 = \{\sigma_1; \sigma_2 \mid \sigma_1; \sigma_2 \text{ is defined}, \sigma_i \in L_i, i = 1, 2\}$$

Star is defined as a least fixpoint: $L^* = \bigcup_{n \in \mathbb{N}} L^n$ where $L^0 = \{\sigma_\epsilon\}$, $L^{n+1} = L^n; L$.

Conclusions:

- A Kleene algebra of sets of real numbers with a normal form theorem.
- A class of finite automata with timing constraints that is closed under complementation and has some other features from finite untimed automata.

Applications:

- Removing stuttering steps from timed automata.
- Enhancing event-clock temporal logic with "periodic" constraints.
- Decidable subsets of Simple Duration Calculus.

Problems:

- What happens if the generating intervals have any bounds?
- Axiomatization for the Kleene algebra.

The subalgebra generated by intervals with integer bounds has *two* generators: $\{1\}$ and $(0,1)$. None of them can be generated by the other.

However they are not independent:

$$\begin{aligned} \neg\{1\}^* &= (0,1) + \{1\}^* \\ (0,1)^* &= (\{0\} \cup (0,1)) + \{1\}^* \end{aligned}$$