Flexibility Analysis of Resource Timed Automata

Jonas Hansen[1] and Kim Guldstrand Larsen[2]

[1] Aalborg University, Aalborg, Denmark
jonash@cs.aau.dk
[2] Aalborg University, Aalborg, Denmark.
kgl@cs.aau.dk

We study methods for analyzing flexible resource systems. In such, real valued resource variables (eg. batteries, water tanks, hydraulic pumps, etc.) are continuously affected by system behavior while system executions must conform to some bounds on said variables. Flexibility refers to how the variables can change during some time interval. Flexibility analysis refers to establishing guarantees on whether a system in some state can realize some flexibility in its variables. A real world example is the Smart Grid model, where prosumers i.e. components both producing and consuming power are present. Taking advantage of prosumers in scheduling and aggregation activities is an ongoing research endeavor. There are many reliable methods for capturing the flexibility of prosumer components. Flexibility Models (FM) as defined in [4] provides a lossy but easily scalable model. Dependency-based Flexoffer (DFO) as defined by Šikšnys. et. al. in [6] is a low complexity generalized model, allowing for efficient computation of outer and inner approximation of exact prosumer flexibility.
We tackle the problem of developing an automata based formalism suitable for modeling systems constituting multiple resource variables. Extending the work on bounded infinite behavior in Energy Timed Automata ETA done by Bacci et. al. in [3] we study the multi variable setting of finite behavior in RTAs.
We motivate this by providing methods for verifying flexibility properties, denoted as *Flexoffers*. We formally define a *Flexoffer* of a resource variable η as a pair (I_T, I_F), where I_T defines a time interval and I_F defines bounds on η. We say that a *Flexoffer* is *feasible* if a state satisfying I_F is reachable in some time $t \in I_T$. The formalism, denoted Resource Timed Automata (RTA), is a variation of Weighted Timed Automata (WTA) [2] which describes Timed Automata (TA) [1] extended with real valued variables, constrained by bounds.
RTAs extends ETAs[3] in the sense that they generalize the concept of a single energy variable into a set of resource variables. RTAs conforming to certain branching (single infinite path guaranty) and clock restrictions (reset guarantees) extends the notion of Segmented ETAs as defined in [3].
We show how Flexoffer feasibility can be determined by translating Segmented RTAs into first-order linear arithmetic expressions and solving these using an appropriate quantifier elimination method such as Fourier Motzkin Elimination. As segmented RTAs are guaranteed to model systems of linear inequalities, we insure that Flexibility exerts certain convexness properties. We use the tool Mathematica[5] to model Segmented RTAs as systems of linear inequalities, and the library Mjollnir to perform efficient linear quantifier elimination.
In section 1 we define RTAs and provide suitable definitions for resource constrained runs. In section 2 we describe the intuition behind flexibility analysis and provide definitions for *Flexoffer*. Additionally we present the intuition and results behind *Flexoffer* feasibility in single variable RTAs.

1 Resource Timed Automata

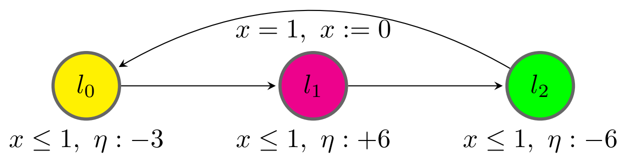


Figure 1: RTA A defined over clock x and resource η. Update rates are defined for η in each location.

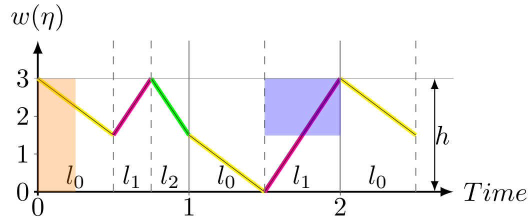
We start by introducing preliminaries. Let C denote a set of clocks. We define the set $\mathcal{B}(C)$ of clock constraints over C, by the abstract syntax: $g\ :=\ x \sim c \mid g \wedge g$, where $x, y \in C$, $c \in \mathbb{Q}_{\geq 0}$ and $\sim \in \{\leq, \geq\}$.
A clock stores the amount of time elapsed since last reset, captured by a valuation $v : C \to \mathbb{R}_{\geq 0}$. Let $d \in \mathbb{R}_{\geq 0}$, $x \in C$ then $v + d$ is defined as: $(v + d)(x) = v(x) + d$.
Resetting clocks $r \subseteq C$, captured by $v[r]$ is defined as: $v[r](x) = \begin{cases} 0; & x \in r \\ v(x); & otherwise \end{cases}$
Let $g, g_1, g_2 \in \mathcal{B}(C)$ be clock constraints and $x \in C$ denote a clock. We define the evaluation of a clock constraint $v \models g$ inductively on the structure of g as follows: $v \models x \sim c \Leftrightarrow v(x) \sim c$ and $v \models g_1 \wedge g_2 \Leftrightarrow v \models g_1 \wedge v \models g_2$, where $c \in \mathbb{Q}_{\geq 0}$ and $\sim \in \{\leq, \geq\}$.
We denote initial clock valuation as $v_0(x) = 0$.

Definition 1 (RTA)**.** *An RTA over a finite set of clocks C, a finite set of resource variables $\mathcal{E}$, and a finite set of actions Act is defined as a tuple: (L, L_0, E, r, u, I), where:*
- *L is a finite set of locations*
- *$L_0 \subseteq L$ is a set of initial location s.t. $L_0 \neq \emptyset$*
- *$E \subseteq L \times \mathcal{B}(C) \times Act \times 2^C \times L$ is a finite set of edges*
- *$u : E \to (\mathcal{E} \to \mathbb{Q})$ assigns discrete updates of resource variables to edges*
- *$r : L \to (\mathcal{E} \to \mathbb{Q})$ assigns update rates of resource variables to locations*
- *$I : L \to \mathcal{B}(C)$ assigns invariants to locations*

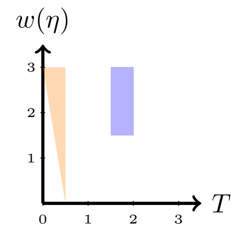
If $a \in Act$ and $(l, g, a, r, l') \in E$ we say that $(l \xrightarrow{g,a,r} l')$.
A set of resource variables $\mathcal{E} = \{\eta_1, \eta_2, \cdots, \eta_n\}$ stores the accumulated resource values, which are captured by a valuation $w : \mathcal{E} \to \mathbb{R}$. For $f : \mathcal{E} \to \mathbb{Q}$, $\eta \in \mathcal{E}$, and $d \in \mathbb{R}_{\geq 0}$, we define the two following operations: $(w + f)(\eta) = w(\eta) + f(\eta)$ and $(f \cdot d)(\eta) = f(\eta) \cdot d$.
Figure 1 depicts a small example RTA.
States/configurations of RTAs are defined as triples of locations, clock valuations and resource valuations: $(l, v, w) \in L \times (C \to \mathbb{R}_{\geq 0}) \times (\mathcal{E} \to \mathbb{R})$. A legal state must satisfy the invariant of its location, i.e. $v \models I(l)$. Semantically, RTAs defines a Resource Timed Labeled Transition System (RTLTS) $(S, \longrightarrow, \omega)$ of states S, transition relation $\longrightarrow$ over labels Lab and resource valuation ω.
$\omega : S \to (\mathcal{E} \to \mathbb{R})$ defines a mapping from states to resource valuations. eg. for state $s = (l_1, x = 0.5, \eta = 1.5)$ in RTA A of figure 1 we have $\omega(s)(\eta) = w(\eta) = 1.5$. The transition relation defines discrete transitions over Act and delay transitions over $d \in \mathbb{R}_{\geq 0}$, e.g. in A we have:

$(l_2, x = 1, \eta = 3) \rightarrow (l_0, x = 0, \eta = 3) \xrightarrow{0.5} (l_0, x = 0.5, \eta = 1.5) \rightarrow (l_1, x = 0.5, \eta = 1.5) \xrightarrow{0.25} (l_1, x = 1, \eta = 3)$

If $l_0 \in L_0$, $v_0 \models I(l_0)$ and $w : \mathcal{E} \rightarrow \mathbb{R}$, then (l_0, v_0, w) denotes an initial state of T.
Constraints on resource variables are defined as closed intervals over $\mathbb{Q}_{\geq 0}$. A state s satisfies a constraint h if the valuation in s is within h eg. for $\omega(s)(\eta) = 1.5$ and $h = [1, 2]$ we have $s \models h \Leftrightarrow 1.5 \in [1, 2]$. We define runs of RTAs as runs of the generated RTLTS in which all



(a) A run from $(l_0, x = 0, w(\eta) = 3)$ to $(l_0, x = 0.5, w(\eta) = 1.5)$ in RTA A constrained in h. The blue rectangle defines the Flexoffer $([1.5, 2], [1.5, 3])$ and the orange rectangle defines the Flexoffer $([0, 0.5], [0, 3])$.



(b) All pairs of $T \in I_T$ and $w(\eta) \in I_F$ satisfiable by a reachable state from initial state $(l_0, x = 0, \omega(\eta) = 3)$ in A.

Figure 2

states satisfy a given resource constraint.

Definition 2 (Runs of RTLTS)**.** *Let $T = (S, \omega, \longrightarrow)$ be an RTLTS over $\mathcal{E}$ s.t. $\alpha \in Lab$ and let $h \in \mathcal{H}(\mathcal{E})$ be a resource constraint.*
We formally define a (infinite) run of T as: $\mathcal{T} = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_{j-1}} s_j \cdots$ where, $\forall i. s_i \xrightarrow{\alpha_i} s_{i+1} \in \longrightarrow$, and $\forall i. \omega(s_i) \models h$

Additionally, $\mathcal{R}(T)$ denotes all runs of T. If $s \in S$ is a state of T then $\mathcal{R}(T)(s)$ denotes all runs of T starting in s. $\mathcal{T}[d] \in S$ denotes the state immediately following a total delay $d \in \mathbb{R}_{\geq 0}$ in $\mathcal{T}$ from the initial state of $\mathcal{T}$. Figure 2a depicts an arbitrary run of RTA A from figure 1.
RTAs where only a single variable constitute the set of resources is semantically equivalent to ETAs i.e. ETAs are special cases of RTAs.

2 Flexibility Analysis

Definition 3 (Flexoffer)**.** *Let $I_T = [T_l, T_u]$ be a time interval s.t. $T_l, T_u \in \mathbb{R}_{\geq 0}$ and let $I_F = [c_l, c_u]$ be a closed interval over $\mathbb{Q}_{\geq 0}$. We define a Flexoffer as the tuple: (I_T, I_F).*

Flexibility analysis refers to establishing guarantees on whether a Flexoffer is satisfied by a run given some state of an RTA.

Definition 4 (Flexoffer satisfiability)**.** *Let A be an RTA defined over $\eta \in \mathcal{E}$ and constrained in h. We denote the RTLTS generated by A as $T = (S, \longrightarrow, \omega)$ and $s \in S$ some state in T. Let (I_T, I_F) be a Flexoffer where $I_F \subseteq h$. Let $t \in I_T$ and $c \in I_F$. We say that (t, c) is realizable*

from s if there exists $\mathcal{T} \in \mathcal{R}(T)(s)$ s.t. $\omega(\mathcal{T}[t])(\eta) = c$.
We define two types of logical propositions on the reachability of (I_T, I_F); the existential property $Eflex(I_T, I_F)$ and Universal property $Aflex(I_T, I_F)$. We define satisfiability of s as:

$$s \models Eflex^{\eta}(I_T, I_F) \textit{ iff } \exists t \in I_T. \exists c \in I_F.\ (t, c) \textit{ is realizable from } s$$

$$s \models Aflex^{\eta}(I_T, I_F) \textit{ iff } \forall t \in I_T. \forall c \in I_F.\ (t, c) \textit{ is realizible from } s$$

Let A depicted in figure 1 be constrained in $h = [0, 3]$. All realizable pairs in I_T and I_F of A from initial state $(l_0, x = 0, \omega(\eta) = 3)$ within the Flexoffer $([1.5, 2], [1.5, 3])$ and $([0, 0.5], [0, 3])$ (visualized in figure 2a) can be depicted as convex polytopes as shown in figure 2b. Feasibility can be determined through this method eg. the orange polytope specifies that the existential property is satisfied, however the universal property is not satisfied for Flexoffer $([0, 0.5], [0, 3])$.

References

[1] Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, April 1994.

[2] Rajeev Alur, Salvatore La Torre, and George J. Pappas. Optimal paths in weighted timed automata. In Maria Domenica Di Benedetto and Alberto Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, pages 49–62, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[3] Giovanni Bacci, Patricia Bouyer, Uli Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Pierre-Alain Reynier. Optimal and robust controller synthesis. In Klaus Havelund, Jan Peleska, Bill Roscoe, and Erik de Vink, editors, *Formal Methods*, pages 203–221, Cham, 2018. Springer International Publishing.

[4] Matthias Boehm, Lars Dannecker, Andreas Doms, Erik Dovgan, Bogdan Filipič, Ulrike Fischer, Wolfgang Lehner, Torben Bach Pedersen, Yoann Pitarch, Laurynas Siksnys, and Tea Tušar. Data management in the mirabel smart grid system. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops, Berlin, Germany, March 30, 2012.*, pages 95–102, United States, 2012. Association for Computing Machinery. null ; Conference date: 30-03-2012 Through 30-03-2012.

[5] Wolfram Research, Inc. Mathematica, Version 12.3.1. Champaign, IL, 2021.

[6] Laurynas Šikšnys and Torben Bach Pedersen. Dependency-based flexoffers: Scalable management of flexible loads with dependencies. In *Proceedings of the Seventh International Conference on Future Energy Systems*, e-Energy '16, New York, NY, USA, 2016. Association for Computing Machinery.