Intersection of Regular Signal-Event (Timed) Languages

Béatrice Bérard[1], Paul Gastin[2], and Antoine Petit[2]

[1] LAMSADE, Université Paris Dauphine & CNRS,
Place du Maréchal de Lattre de Tassigny, F-75775 Paris Cedex 16, France
[2] LSV, ENS de Cachan & CNRS,
61 av. du Président Wilson, F-94235 Cachan Cedex, France

Abstract. We propose in this paper a construction for a "well known" result: regular signal-event languages are closed by intersection. In fact, while this result is indeed trivial for languages defined by Alur and Dill's timed automata (the proof is an immediate extension of the one in the untimed case), it turns out that the construction is much more tricky when considering the most involved model of signal-event automata. While several constructions have been proposed in particular cases, it is the first time, up to our knowledge, that a construction working on finite and infinite signal-event words and taking into account signal stuttering, unobservability of zero-duration τ-signals and Zeno runs is proposed.

1 Introduction

In order to accept timed words, the model of timed automata was first proposed in [1,2]. It has been widely studied for the last fifteen years and successfully applied to industrial cases. For this model, an observation, called a time-event word, may be viewed as an alternating sequence of waiting times and instantaneous actions. A timed automaton is a finite automaton extended with variables called clocks, designed to recognize time-event words: time elapses while the control stays in a given node and an event is observed when a discrete transition occurs.

Another model was introduced by [3], and further studied in [10,4,11] with the aim of describing hardware systems. In this case, an observation is a signal word, i.e. a sequence of factors a^d, where a is a signal and d is its duration. The original model of timed automata was then modified to fit this setting: a signal is emitted while the automaton stays in some state and no event is produced when a discrete transition is fired. In this framework, when a transition occurs between two states with the same signal a, we obtain a^{d_1} followed by a^{d_2}, which are merged in $a^{d_1+d_2}$. This phenomenon is called stuttering.

It was noticed in [4] that both approaches are complementary and can be combined in an algebraic formalism to obtain the so-called signal-event monoid. Timed automata can be easily adapted to take both signals and events into account, thus yielding signal-event automata: states emit signals and transitions produce events.

We consider in this paper both finite and infinite behaviors of signal-event automata and we also include unobservable events (ε-transitions) and hidden signals (τ-labeled states). These features can be very useful and even necessary, for instance for handling abstractions [6]. They also allow us to get as special cases the initial models of timed automata and signal automata.

E. Asarin and P. Bouyer (Eds.): FORMATS 2006, LNCS 4202, pp. 52–66, 2006.

We study in this paper a construction for the intersection of languages accepted by signal-event automata. Surprisingly, it turns out that this closure property is rather difficult to obtain. Usually, the construction for intersection relies on a basic synchronized product. In [1], which deals with infinite time-event words only (no signal is involved), a Büchi-like product is performed. The situation is more complex for signal words due to stuttering of signals and unobservability of zero-duration signals. In [3], a construction is given for the intersection of signal automata, but neither signal stuttering nor unobservability of zero-duration signal is taken into account, and only finite runs are considered. Note that the full version [4] of [3] deals with the intersection of usual timed automata only. In [10], in order to obtain a determinization result, a construction is proposed to remove stuttering and zero-duration signals on signal automata using a single clock but intersection is not considered directly. In [11], stuttering is handled but intersection is done for signal automata acting on finite sequences only and without zero-duration signals. To cope with stuttering, intermediate states and ε-transitions are added to the automaton, thus introducing all possible ways of splitting some signal a^d into a finite concatenation $a^{d_1} \dots a^{d_n}$. When dealing with ω-sequences, this approach would produce additional Zeno runs leading to another difficulty arising with the possibility to accept a finite signal-event word of finite duration with either a finite run or an infinite Zeno run.

We provide a general construction for the intersection of signal-event timed automata working on finite and infinite signal-event words. We solve the main difficulties of signal stuttering, unobservability of zero-duration τ-signals and Zeno runs. Note that, although Zeno behaviours have been studied (see for instance [12,9]), it has been sometimes argued that excluding Zeno runs directly from the semantics of timed automata is a realistic assumption, since they do not appear in "real" systems. However, the semantics of timed automata used by model-checking tools like UPPAAL do include Zeno runs while performing forward reachability analysis (this can be easily checked with an example). Hence, we think that the general theory should include Zeno runs as well.

We first give in Section 2 precise definitions of finite and infinite signal-event languages, with the corresponding notion of signal-event automata. Section 3 establishes a normal form for signal-event automata so that no infinite run accepts a finite word with finite duration and no finite run accepts a word with infinite duration. This normal form is useful for the general construction of intersection of signal-event automata dealing both with signal stuttering and with finite and infinite sequences proposed in Section 4.

For lack of space, this paper does not contain the proofs of the correctness of the different automata constructions we propose. These proofs are available in the technical report [7].

2 Signal-Event Words and Signal-Event Automata

Let Z be any set. We write Z^* (respectively Z^ω) the set of finite (respectively infinite) sequences of elements in Z, with ε for the empty sequence, and $Z^\infty = Z^* \cup Z^\omega$ the set of all sequences of elements in Z. The set Z^∞ is equipped with the usual partial concatenation defined from $Z^* \times Z^\infty$ to Z^∞.