The open time interval \((\delta_i, \delta_{i+1})\), while lies on the boundary \(bdry(\tilde{Q}_i, \tilde{Q}_{i+1})\) at time \(\delta_i\), i.e. for each \(1 \leq i < n\) we have that \(f(\delta_i) \in bdry(\tilde{Q}_i, \tilde{Q}_{i+1})\). Now, by construction of \(Init'\) and by recalling that \(f(0) \in \tilde{Q}_1\), then the state \(s_m = (l_1, u_e)\), where \(u = u_e |_X\), is an initial state of \(\mathcal{H}_m\) (i.e. \(s_m \in Init'\)). From the state \(s_m\), by construction, there exists the activity \(f_e\) that is \(f\) with the additional flow condition for the extra variable (clock) \(t \in X'\). By following \(f_e\), the system jumps among locations \(l_1, l_12, l_2, \ldots, l_n\), according to their invariants. This is possible because, by construction of \(\mathcal{H}_m\), locations of the form \(l_i\) are associated with the invariant \(\tilde{Q}_i\), while locations of the form \(l_i, i+1\) are associated with an invariant containing \(bdry(\tilde{Q}_i, \tilde{Q}_{i+1})\). Now, if \(f(\delta)\) also belongs to \([G]\), we conclude that from the state \(s_m\) and by following the activity \(f_e\), the system can reach the state \(s'_m = (l_n, v_e)\), where \(v_e\) is the same as \(v\) (except for the clock variable \(t\)). Hence, \(v = v_e |_X\), and we conclude that \(v \in Reach(\mathcal{H}_m) \downarrow \text{Loc''}_c \setminus \text{Loc}_c \setminus X\). Otherwise \(f(\delta) \notin G\) and then the system cannot remain in location \(l_n\), because its invariant \(\tilde{Q}_n\) is such that \(G \cap \tilde{Q}_n = \emptyset\). Hence, the system is constrained to jump to location \(l_n\). This jump is allowed because satisfies the invariant of location \(l_n\) (i.e. the topological closure of \(\tilde{Q}_n\) intersected with the condition \(t \leq \varepsilon\)). The automaton \(\mathcal{H}_m\) may jump to location \(l_n\) when the current valuation is \(f_e(\delta - \varepsilon)\). According to the invariant of \(\tilde{Q}_n\) on the clock \(t\), the valuation \(v_e = f_e(\delta)\) can be reached after time \(\varepsilon\). At that time the jump to location \(l_n\) is allowed, and when this happens the state \(s'_m = (l_n, v_e)\) is reached. Clearly, \(v = v_e |_X\) and we conclude that \(v \in Reach(\mathcal{H}_m) \downarrow \text{Loc''}_c \setminus \text{Loc}_c \setminus X\).

The second case is when \(loc = l\) and \(loc' = l'\). We follow a similar argumentation of the first case for the subcase with \(f(\delta) \in G\). Indeed, due to the must semantics, when the current valuation satisfies the guard \(G\), i.e. \(v = f(\delta) \in G\), the automaton \(\mathcal{H}_M\) must jump to the location \(l'\) by reaching the state \(s'_m = (l', v)\). On the other hand, when \(\mathcal{H}_m\) reaches the state \(s'_m = (l_n, v_e)\) it is enforced to immediately leave this location due to the location invariant \(t = 0\). Hence the transition to location \(l'\) must be taken, by reaching the state \(s'_m = (l', v_e)\). Therefore, \(v \in Reach(\mathcal{H}_m) \downarrow \text{Loc''}_c \setminus \text{Loc}_c \setminus X\) and this concludes the second case.

The last case is when \(loc = loc' = l'\). In this case, the
run that leads from $s_M$ to $s'_M$ consists of the timed transition $s_M = (l', v) \xrightarrow{\delta} s'_M = (l', v)$, for some admissible activity $f \in \text{Adm}(s_M)$ and time $\delta \geq 0$. By construction, the location $l'$ of $H_M$ is associated with same invariant and flow of location $l'$ of $H_M$ (except the extra conditions on the clock $t$ that do not affect the timed step), and then trivially the automaton $H_M$ may reach the state $s'_M = (v_e, l')$, where $v = v_e | x$. Hence, we can write $v \in \text{Reach}(H_M) \downarrow_{\loc \setminus \loc_e, X}$ by concluding the proof for the automata $H_M$ and $H_m$ that only consist of a single must transition.

The result can be easily extended to a general automaton $H_M$. Indeed it is enough to apply our technique (described in the main paper) to each source location $l$ of a must transition. If the location has several outgoing transitions, then the construction is applied by considering the guard $G$ as the union of the individual guards of the transitions. Finally, every may transition from a location $l$ to a location $l''$ is encoded by a may transition from the locations induced by $l$ to the location $l''$ (with the same flow and invariant as $l''$ of $H_M$).

Lemma 2. For a linear hybrid automaton (LHA) $H_M = (\loc, X, \text{Edg}, \text{Flow}, \text{Inv}, \text{Init})$ with must transitions featuring closed guards, there exists a hybrid automaton $H_m = (\loc', X', \text{Edg}', \text{Flow}', \text{Inv}', \text{Init}')$ with may transitions and a location set $\loc_e \subseteq \loc'$ such that

$$
\text{Reach}(H_m) \downarrow_{\loc \setminus \loc_e, X} \subseteq \text{CReach}(H_M).
$$

Proof. Similarly to Lemma 1, we first show the lemma for the automata $H_M$ and $H_m$ that only consist of a single must transition, and then we extend the result to general linear hybrid automata.

Let $v$ be a valuation such that $v \in \text{Reach}(H_m) \downarrow_{\loc \setminus \loc_e, X}$. By definition of projection, there exists a state $s_M = (v_e, \loc') \in \text{Reach}(H_m)$ such that $v = v_e | x$ and $\loc' \in \loc'$. By definition of reachable states, there exists an initial state $s_m = (u, \loc) \in \text{Init}'$ and a run from $s_m$ to $s'_M$. By definition of $\text{Init}'$, location $\loc$ could be location $l_1$, location $l_u$ or one of the locations of the form $l_i$, while by definition of projection, location $\loc'$ could be location $l'$, location $l_u$ or one of the locations of the form $l_i$ or $l_j$. By combining the conditions above, we can distinguish several cases.

Consider the case when both $\loc$ and $\loc'$ are in the form $l_i$ (for example $\loc = l_1$ and $\loc' = l_u$). By using a similar argumentation of the first case in the proof of Lemma 1, there exists an admissible activity $f_e \in \text{Adm}(s_m)$ and a sequence of times $0 = \delta_0 < \delta_1 < \ldots < \delta_n = \delta$ such that in the automaton $H_m$, it is possible, starting from $s_m$, to reach the state $s'_M$ by jumping among locations $l_1, l_{12}, l_2, \ldots, l_n$. During this run, the invariants $\hat{Q}_1, \text{bdry} \{\hat{Q}_1, \hat{Q}_2\}$, $\hat{Q}_2, \ldots, \hat{Q}_n \in [\hat{G}]$ are constantly satisfied. From $s_m = (l_1, u) \in \text{Init}'$ by construction of $H_m$ there exists an initial state $s_M = (l, u) \in \text{Init}$ such that $u = u_e | x$ and $u \in \hat{Q}_1$. Again by construction of $H_m$, it is easy to show that there exists an activity $f \in \text{Adm}(l, u)$, where $f$ is defined like $f_e$ except for the condition on the extra variable $t$, and a time $\delta \geq 0$, such that there exists a timed step $s_M \xrightarrow{f} s'_M$ and $s'_M = (l, f(\delta))$. Hence, $s'_M \in \text{Reach}(H_M)$ and clearly the valuation $v = f(\delta) \in \text{CReach}(H_M)$. The case with $\loc$ of the form $l_i$ and $\loc'$ of the form $l_j$ can be easily proven by following the same way of the previous case.

For the case when $\loc = l_1$ (just an example for a location of the form $l_j$) and $\loc' = l_u$, we can partially follow the procedure described for the first case. We need to consider that now $v_e = f(\delta) \in G$ because of the invariant of $l_u$, and that $\delta_n < \delta$ (otherwise, $f(\delta_n) \in G$). This means that in order to reach $s'_M = (l_u, v_e)$ from the initial state $s_m = (l_1, u)$ the system must first pass through locations $l_1, l_{12}, l_2, \ldots, l_n$ and make a jump from $l_n$ to $l_u$. When the valuation $v_e$ is reached in $l_u$ the system jumps to $l_u$ by reaching the state $s'_M = (l_u, v_e)$. To conclude this case, we need to analyze the jumps among locations $l_i$, $l_{ni}$ and $l_u$ in more detail. When the transition from $l_{ni}$ to $l_u$ is taken, the clock $t$ is reset and the invariant of $l_u$ imposes that the system must jump to $l_u$ after spending at most $\varepsilon$ time units in this location. This means that in location $l_u$ and by following the activity $f_e$ for a time $0 < \varepsilon' \leq \varepsilon$, the valuation $v_e$ will be reached (i.e. $f_e(\delta_n + \varepsilon') = f_e(\delta) = v_e$). Notice that, if the flow allows non-monotonic dynamics on the variables belonging to $X$, it could exists another time $\varepsilon' < \varepsilon' \leq \varepsilon$ such that $f_e(\delta_n + \varepsilon'') = f_e(\delta) = v_e$. Consider first the case when this does not happen. It is easy to show that there exists a time step $s_M = (l, u) \xrightarrow{\delta} (l, f(\delta_n + \varepsilon'))$. Recalling that $f(\delta_n + \varepsilon') = v \in G$, then the must semantics is such that it constraints a jump from $l$ to $l'$, by reaching the state $s'_M = (l_u, v)$, and we can write that $v \in \text{CReach}(H_M)$. Now consider the case when $H_m$ jumps to $l_u$ after the time $\varepsilon''$. This seems to be not allowed in the automaton $H_M$. Indeed because of the must semantics, the jump happens exactly when the system, by following $f$, reaches a valuation satisfying $G$ (i.e. at time $\varepsilon''$, and hence $\varepsilon''$ would not exists. But according to a fundamental property of LHA's (Alur, Henzinger, and Ho 1996), if the activity $f_e$ leads to the valuation $f_e(\delta_n + \varepsilon'')$, then there always exists a linear activity $f^*$ that does the same. As a consequence, even if $H_m$ jumps at time $\varepsilon''$ (and hence after having satisfied $G$ for some time by then), the automaton $H_M$ is also able to reach the corresponding valuation by following a straight-line, i.e. by touching $G$ only one time. Hence, we can write that $v \in \text{CReach}(H_M)$.

Note that the case when $\loc = l_1$ and $\loc' = l'$ can be handled similarly to the previous one. Indeed, once entered location $l_u$, the system must immediately jump to $l'$ (because of the invariant $t = 0$). The same thing happens in $H_M$ because of the must semantics.

The case when $\loc = l_1$ can be accompanied only with $\loc' = l'$ and can be easily derived from the case before. Finally, the case when $\loc = \loc' = l'$ is trivially valid by construction of $H_m$.

To extend the result to general automata, it is enough to follow the same procedure described for the extension of Lemma 1. \qed
Lemma 3. For an affine hybrid automaton $H_M = (\text{Loc}, X, \text{Edg}, \text{Flow}, \text{Inv}, \text{Init})$ with must transitions featuring closed guards, there exists a hybrid automaton $H_m = (\text{Loc'}, X', \text{Edg'}, \text{Flow'}, \text{Inv'}, \text{Init'})$ with may transitions and a location set $\text{Loc}_e \subseteq \text{Loc'}$ such that

$$\text{CReach}(H_M) \subseteq \text{Reach}(H_m) \downarrow \text{Loc'\textbackslash Loc}_e, X$$

and the approximation can be made arbitrarily precise.

Proof. Lemma 1 already states that $\text{CReach}(H_M) \subseteq \text{Reach}(H_m) \downarrow \text{Loc'\textbackslash Loc}_e, X$. Informally, to show that the approximation can be made arbitrarily precise, we need to identify those elements that belong to $\text{Reach}(H_m) \downarrow \text{Loc'\textbackslash Loc}_e, X$ but do not belong to $\text{CReach}(H_M)$ (i.e. the set $D = \text{Reach}(H_m) \downarrow \text{Loc'\textbackslash Loc}_e, X \setminus \text{CReach}(H_M)$). Then, we need to show that it is possible to systematically reduce the set $D$.

According to the proof of Lemma 2, the only valuations that could be in $D$ are those on the form $f(\delta_n + \varepsilon'')$. Indeed, because the considered automaton $H_M$ belongs to the class of affine automata, we cannot use the above mentioned property to replace an activity $f$ by a linear activity.

However, it is easy to argue that by choosing a smaller $\varepsilon$, we can arbitrarily reduce the cardinality of the set $D$. For example, consider the case when $H_m$ touches $G$ at the time moment $\varepsilon'$ and then at the time moment $\varepsilon''$. By setting $\varepsilon < \varepsilon''$, we prevent the system touching $G$ a second time and thus reduce the cardinality of the set $D$. $\square$

To prove Theorem 1, we apply Lemma 1, 2 and 3. To be more precise, we show the LHA case with Lemma 1 and Lemma 2. To prove the theorem for affine HA, we use Lemma 1 and 3.

Acknowledgments

This work was partly supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Center “Automatic Verification and Analysis of Complex Systems” (SFB/TR 14 AVACS, http://www.avacs.org/), by the European Research Council (ERC) under grant 267989 (QUAREM), by the Austrian Science Fund (FWF) under grants S11402-N23 (RiSE) and Z211-N23 (Wittgenstein Award), and by the Swiss National Science Foundation (SNSF) as part of the project “Automated Reformulation and Pruning in Factored State Spaces (ARAP)”.

References

