PDDL+ Planning with Hybrid Automata: Foundations of Translating Must Behavior (Technical Report)

Sergiy Bogomolov
IST Austria, Austria
University of Freiburg, Germany
sergiy.bogomolov@ist.ac.at

Daniele Magazzeni
King's College London
United Kingdom
daniele.magazzeni@kcl.ac.uk

Stefano Minopoli
UJF - Lab. VERIMAG
Grenoble - France
stefano.minopoli@imag.fr

Martin Wehrle
University of Basel
Switzerland
martin.wehrle@unibas.ch

This technical report contains the proof of Theorem 1 of the paper "PDDL+ Planning with Hybrid Automata: Foundations of Translating Must Behavior" (Bogomolov et al. 2015), using the same notation and terminology.

Lemma 1. *For a hybrid automaton (either LHA or AHA) $\mathcal{H}_M = (Loc, X, Edg, Flow, Inv, Init)$ with must transitions featuring closed guards, there exists a hybrid automaton $\mathcal{H}_m = (Loc', X', Edg', Flow', Inv', Init')$ with may transitions and a location set $Loc_\varepsilon \subset Loc'$ such that*

$$CReach(\mathcal{H}_M) \subseteq Reach(\mathcal{H}_m) \downdownarrows_{Loc' \setminus Loc_\varepsilon, X} .$$

Proof. We first show that the lemma is valid when the automata $\mathcal{H}_M$ and $\mathcal{H}_m$, resp., are those shown in (Bogomolov et al. 2015) that only consist of a single transition. In this case, the set Loc_ε consists of the locations $\breve{l}_i$. Moreover, the considered valuations are those reachable in Loc' but not in Loc_ε. Then, the result is extended for a general must automaton $\mathcal{H}_M$.

Let v be a reachable valuation in the automaton $\mathcal{H}_M$, i. e. $v \in CReach(\mathcal{H}_M)$. Hence by definition there exists a state $s'_M = \langle loc', v\rangle \in Reach(\mathcal{H}_M)$, where $loc' \in \{l, l'\}$. By definition of reachable sets there exists an initial state $s_M = \langle loc, u\rangle \in Init$, where $loc \in \{l, l'\}$, such that there exists a run leading from s_M to s'_M. Depending on loc and loc', we distinguish three cases.

The first case is when $loc = loc' = l$. Clearly since source and target locations coincide, the run could be only a single timed transition. Hence there exists an admissible activity $f \in Adm(s_M)$ and a time $\delta \geq 0$, such that $s_M \xrightarrow{\delta, f} s'_M$. Due to the must semantics, for all $0 \leq \delta' < \delta$ it holds that $f(\delta') \in \overline{G}$, and $f(\delta)$ belongs either to $\overline{G}$ or to G. Notice that, from $f(\delta') \in \overline{G}$ and Lemma 1 of (Benerecetti, Faella, and Minopoli 2013) there exists a sequence of convex components $\widehat{Q}_1, \ldots, \widehat{Q}_n \in [\![\overline{G}]\!]$, and times $0 = \delta_0 < \delta_1 < \ldots < \delta_n = \delta$ such that, for any $0 \leq i < n$ and $\delta' \in (\delta_i, \delta_{i+1})$ we have that $f(\delta') \in \widehat{Q}_i$. This means that the system, by following the activity f, remains always inside the single convex component $\widehat{Q}_i \in [\![\overline{G}]\!]$ along all the open time interval (δ_i, δ_{i+1}), while lies on the boundary $bndry(\widehat{Q}_i, \widehat{Q}_{i+1})$ at time δ_i, i. e. for each $1 \leq i < n$ we have that $f(\delta_i) \in bndry(\widehat{Q}_i, \widehat{Q}_{i+1})$. Now, by construction of $Init'$ and by recalling that $f(0) \in \widehat{Q}_1$, then the state $s_m = \langle l_1, u_e\rangle$, where $u = u_e \downharpoonleft_X$, 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 $\widehat{Q}_i$, while locations of the form $l_{i,i+1}$ are associated with an invariant containing $bndry(\widehat{Q}_i, \widehat{Q}_{i+1})$. Now, if $f(\delta)$ also belongs to $[\![\overline{G}]\!]$, we conclude that from the state s_m and by following the activity f_e, the system can reach the state $s'_m = \langle l_n, v_e\rangle$, where v_e is the same as v (except for the clock variable t). Hence, $v = v_e \downharpoonleft_X$, and we conclude that $v \in Reach(\mathcal{H}_m) \downdownarrows_{Loc' \setminus Loc_\varepsilon, X}$. Otherwise $f(\delta) \in G$ and then the system cannot remain in location l_n because its invariant $\widehat{Q}_n$ is such that $G \cap \widehat{Q}_n = \emptyset$. Hence, the system is constrained to jump to location $\breve{l}_n$. This jump is allowed because satisfies the invariant of location $\breve{l}_n$ (i. e. the topological closure of $\widehat{Q}_n$ intersected with the condition $t \leq \varepsilon$). The automaton $\mathcal{H}_m$ may jump to location $\breve{l}_n$ when the current valuation is $f_e(\delta - \varepsilon)$. According to the invariant of $\breve{l}_n$ on the clock t, the valuation $v_e = f_e(\delta)$ can be reached after time ε. At that time the jump to location l_u is allowed, and when this happens the state $s'_m = \langle l_u, v_e\rangle$ is reached. Clearly, $v = v_e \downharpoonleft_X$ and we conclude that $v \in Reach(\mathcal{H}_m) \downdownarrows_{Loc' \setminus Loc_\varepsilon, X}$.

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 $s'_M = \langle l', v\rangle$. On the other hand, when $\mathcal{H}_m$ reaches the state $s'_m = \langle l_u, v_e\rangle$ 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 $s'_m = \langle l', v_e\rangle$. Therefore, $v \in Reach(\mathcal{H}_m) \downdownarrows_{Loc' \setminus Loc_\varepsilon, X}$ 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 $s_M = \langle l', u\rangle \xrightarrow{\delta, f} s'_M = \langle l', v\rangle$, for some admissible activity $f \in \mathit{Adm}(s_M)$ and time $\delta \geq 0$. By construction, the location l' of $\mathcal{H}_m$ is associated with same invariant and flow of location l' of $\mathcal{H}_M$ (except the extra conditions on the clock t that do not affect the timed step), and then trivially the automaton $\mathcal{H}_m$ may reach the state $s'_m = \langle v_e, l'\rangle$, where $v = v_e \downharpoonright_X$. Hence, we can write $v \in \mathit{Reach}(\mathcal{H}_m) \downdownarrows_{\mathit{Loc}' \setminus \mathit{Loc}_\varepsilon, X}$, by concluding the proof for the automata $\mathcal{H}_M$ and $\mathcal{H}_m$ that only consist of a single must transition.

The result can be easily extended to a general automaton $\mathcal{H}_M$. 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 $\mathcal{H}_M$). □

Lemma 2. *For a linear hybrid automaton (LHA) $\mathcal{H}_M = (\mathit{Loc}, X, \mathit{Edg}, \mathit{Flow}, \mathit{Inv}, \mathit{Init})$ with must transitions featuring closed guards, there exists a hybrid automaton $\mathcal{H}_m = (\mathit{Loc}', X', \mathit{Edg}', \mathit{Flow}', \mathit{Inv}', \mathit{Init}')$ with may transitions and a location set $\mathit{Loc}_\varepsilon \subset \mathit{Loc}'$ such that*

$$\mathit{Reach}(\mathcal{H}_m) \downdownarrows_{\mathit{Loc}' \setminus \mathit{Loc}_\varepsilon, X} \subseteq \mathit{CReach}(\mathcal{H}_M).$$

Proof. Similarly to Lemma 1, we first show the lemma for the automata $\mathcal{H}_M$ and $\mathcal{H}_m$ 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 $v \in \mathit{Reach}(\mathcal{H}_m) \downdownarrows_{\mathit{Loc}' \setminus \mathit{Loc}_\varepsilon, X}$. By definition of projection, there exists a state $s'_m = \langle v_e, \mathit{loc}'\rangle \in \mathit{Reach}(\mathcal{H}_m)$ such that $v = v_e \downharpoonright_X$ and $\mathit{loc}' \in \mathit{Loc}'$. By definition of reachable sets, there exists an initial state $s_m = \langle \mathit{loc}, u_e\rangle \in \mathit{Init}'$ and a run from s_m to s'_m. By definition of Init', location loc could be location l', 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_{ij}. 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 $\mathit{loc} = l_1$ and $\mathit{loc}' = l_n$). By using a similar argumentation of the first case in the proof of Lemma 1, there exists an admissible activity $f_e \in \mathit{Adm}(s_m)$ and a sequence of times $0 = \delta_0 < \delta_1 < \ldots < \delta_n = \delta$ such that in the automaton $\mathcal{H}_m$ it is possible, starting from s_m, to reach the state s'_m by jumping among locations l_1, l_{12}, l_2, …, l_n. During this run, the invariants $\widehat{Q}_1$, $\mathit{bndry}(\widehat{Q}_1, \widehat{Q}_2)$, $\widehat{Q}_2$, …, $\widehat{Q}_n \in [\![\overline{G}]\!]$ are constantly satisfied. From $s_m = \langle l_1, u_e\rangle \in \mathit{Init}'$ by construction of $\mathcal{H}_m$ there exists an initial state $s_M = \langle l, u\rangle \in \mathit{Init}$ such that $u = u_e \downharpoonright_X$ and $u \in \widehat{Q}_1$. Again by construction of $\mathcal{H}_m$, it is easy to show that there exists an activity $f \in \mathit{Adm}(\langle l, u\rangle)$, 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 $s_M \xrightarrow{f, \delta} s'_M$ and $s'_M = \langle l, f(\delta)\rangle$. Hence, $s'_M \in \mathit{Reach}(\mathcal{H}_M)$ and clearly the valuation $v = f(\delta) \in \mathit{CReach}(\mathcal{H}_M)$. The case with loc of the form l_i and loc' of the form l_{ij} can be easily proven by following the same way of the previous case.

For the case when $\mathit{loc} = l_1$ (just an example for a location of the form l_i) and $\mathit{loc}' = l_u$, 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 $s'_m = \langle l_u, v_e\rangle$ from the initial state $s_m = \langle l_1, u_e\rangle$ the system must first pass through locations l_1, l_{12}, l_2, …, l_n and make a jump from l_n to $\breve{l}_n$. When the valuation v_e is reached in $\breve{l}_n$ the system jumps to l_u by reaching the state $s'_m = \langle l_u, f_e(\delta)\rangle$. To conclude this case, we need to analyze the jumps among locations l_n, $\breve{l}_n$ and l_u in more detail. When the transition from l_n to $\breve{l}_n$ is taken, the clock t is reset and the invariant of $\breve{l}_n$ imposes that the system must jump to l_u after spending at most ε time units in this location. This means that in location $\breve{l}_n$ 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 $s_M = \langle l, u\rangle \xrightarrow{\delta_n, f} \langle l, f(\delta_n + \varepsilon')\rangle$. 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 $s'_M = \langle l_u, v\rangle$, and we can write that $v \in \mathit{CReach}(\mathcal{H}_M)$. Now consider the case when $\mathcal{H}_m$ jumps to l_u after the time ε''. This seems to be not allowed in the automaton $\mathcal{H}_M$. 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 ε'), and hence ε'' 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 $\mathcal{H}_m$ jumps at time ε'' (and hence after having satisfied G for some time by then), the automaton $\mathcal{H}_M$ 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 $v \in \mathit{CReach}(\mathcal{H}_M)$.

Note that the case when $\mathit{loc} = l_1$ and $\mathit{loc}' = l'$ 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 $\mathcal{H}_M$ because of the must semantics.

The case when $\mathit{loc} = l_u$ can be accompanied only with $\mathit{loc}' = l'$ and can be easily derived from the case before. Finally, the case when $\mathit{loc} = \mathit{loc}' = l'$ is trivially valid by construction of $\mathcal{H}_m$.

To extend the result to general automata, it is enough to follow the same procedure described for the extension of Lemma 1. □

Lemma 3. *For an affine hybrid automaton* $\mathcal{H}_M = (Loc, X, Edg, Flow, Inv, Init)$ *with must transitions featuring closed guards, there exists a hybrid automaton* $\mathcal{H}_m = (Loc', X', Edg', Flow', Inv', Init')$ *with may transitions and a location set* $Loc_\varepsilon \subset Loc'$ *such that*

$$CReach(\mathcal{H}_M) \subseteq Reach(\mathcal{H}_m) \downdownarrows_{Loc' \setminus Loc_\varepsilon, X}$$

and the approximation can be made arbitrarily precise.

Proof. Lemma 1 already states that $CReach(\mathcal{H}_M) \subseteq Reach(\mathcal{H}_m) \downdownarrows_{Loc' \setminus Loc_\varepsilon, X}$. Informally, to show that the approximation can be made arbitrarily precise, we need to identify those elements that belong to $Reach(\mathcal{H}_m) \downdownarrows_{Loc' \setminus Loc_\varepsilon, X}$ but do not belong to $CReach(\mathcal{H}_M)$ (i.e. the set $D = Reach(\mathcal{H}_m) \downdownarrows_{Loc' \setminus Loc_\varepsilon, X} \setminus CReach(\mathcal{H}_M)$). 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 $\mathcal{H}_M$ 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 ε, we can arbitrarily reduce the cardinality of the set D. For example, consider the case when $\mathcal{H}_m$ touches G at the time moment ε' and then at the time moment ε''. By setting $\varepsilon < \varepsilon''$, we prevent the system touching G a second time and thus reduce the cardinality of the set D. □

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

Alur, R.; Henzinger, T.; and Ho, P.-H. 1996. Automatic symbolic verification of embedded systems. *Software Engineering, IEEE Transactions on* 22(3):181–201.

Benerecetti, M.; Faella, M.; and Minopoli, S. 2013. Automatic synthesis of switching controllers for linear hybrid systems: Safety control. *Theor. Comput. Sci.* 493:116–138.

Bogomolov, S.; Magazzeni, D.; Minopoli, S.; and Wehrle, M. 2015. PDDL+ planning with hybrid automata: Foundations of translating must behavior. In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling (ICAPS 2015)*, 42–46.