A Saturation Method for the Modal Mu-Calculus with Backwards Modalities over Pushdown Systems

M. Hague and C.-H. L. Ong

Oxford University Computing Laboratory
Matthew.Hague@comlab.ox.ac.uk   Luke.Ong@comlab.ox.ac.uk

Abstract. We present an extension of an algorithm for computing *directly* the denotation of a modal μ-calculus formula χ over the configuration graph of a pushdown system to allow backwards modalities. Our method gives the first extension of the saturation technique to the full modal μ-calculus with backwards modalities.

1 Introduction

Recently we introduced a saturation method for directly computing the denotation of a modal μ-calculus formula over the configuration graph of a pushdown system [2]. Here we show how this algorithm can be extended to allow backwards modalities. This article is intended as a companion to our previous work, and as such, does not repeat many of the details.

2 Preliminaries

Since we extend our definition of modal μ-calculus, we give the full details here. The reader is directed to our previous work for the remaining preliminaries [2].

Given a set of propositions AP and a disjoint set of variables $\mathcal{Z}$, formulas of the modal μ-calculus are defined as follows (with $x \in AP$ and $Z \in \mathcal{Z}$):

$$\varphi \ := \ x \mid \neg x \mid Z \mid \varphi \ \wedge \ \varphi \mid \varphi \ \vee \ \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \mu Z.\varphi \mid \nu Z.\varphi \ .$$

Thus we assume that the formulas are in *positive form*, in the sense that negation is only applied to atomic propositions. Over a pushdown system, the semantics of a formula φ are given with respect to a *valuation* $V : \mathcal{Z} \rightarrow \mathcal{P}(\mathcal{C})$ which maps each free variable to its set of satisfying configurations and an environment $\rho : AP \rightarrow \mathcal{P}(\mathcal{C})$ mapping each atomic proposition to its set of satisfying configurations. We then have,

$$
\begin{aligned}
[\![x]\!]^{\mathbb{P}}_V &= \rho(x) \\
[\![\neg x]\!]^{\mathbb{P}}_V &= \mathcal{C} \setminus \rho(x) \\
[\![Z]\!]^{\mathbb{P}}_V &= V(Z) \\
[\![\varphi_1 \wedge \varphi_2]\!]^{\mathbb{P}}_V &= [\![\varphi_1]\!]^{\mathbb{P}}_V \cap [\![\varphi_2]\!]^{\mathbb{P}}_V \\
[\![\varphi_1 \vee \varphi_2]\!]^{\mathbb{P}}_V &= [\![\varphi_1]\!]^{\mathbb{P}}_V \cup [\![\varphi_2]\!]^{\mathbb{P}}_V \\
[\![\Box\varphi]\!]^{\mathbb{P}}_V &= \left\{ c \in \mathcal{C} \;\middle|\; \forall c'.c \hookrightarrow c' \Rightarrow c' \in [\![\varphi]\!]^{\mathbb{P}}_V \right\} \\
[\![\Diamond\varphi]\!]^{\mathbb{P}}_V &= \left\{ c \in \mathcal{C} \;\middle|\; \exists c'.c \hookrightarrow c' \wedge c' \in [\![\varphi]\!]^{\mathbb{P}}_V \right\} \\
[\![\overline{\Box}\varphi]\!]^{\mathbb{P}}_V &= \left\{ c \in \mathcal{C} \;\middle|\; \forall c'.c' \hookrightarrow c \Rightarrow c' \in [\![\varphi]\!]^{\mathbb{P}}_V \right\} \\
[\![\overline{\Diamond}\varphi]\!]^{\mathbb{P}}_V &= \left\{ c \in \mathcal{C} \;\middle|\; \exists c'.c' \hookrightarrow c \wedge c' \in [\![\varphi]\!]^{\mathbb{P}}_V \right\} \\
[\![\mu Z.\varphi]\!]^{\mathbb{P}}_V &= \bigcap \left\{ S \subseteq \mathcal{C} \;\middle|\; [\![\varphi]\!]^{\mathbb{P}}_{V[Z \mapsto S]} \subseteq S \right\} \\
[\![\nu Z.\varphi]\!]^{\mathbb{P}}_V &= \bigcup \left\{ S \subseteq \mathcal{C} \;\middle|\; S \subseteq [\![\varphi]\!]^{\mathbb{P}}_{V[Z \mapsto S]} \right\}
\end{aligned}
$$

where $V[Z \mapsto S]$ updates the valuation V to map the variable Z to the set S.

The operators $\Box\varphi$ and $\Diamond\varphi$ assert that φ holds after all possible transitions and after some transition respectively; $\overline{\Box}$ and $\overline{\Diamond}$ are their backwards time counterparts; and the μ and ν operators specify greatest and least fixed points. Another interpretation of these operators is given below. For a full discussion of the modal μ-calculus we refer the reader to a survey by Bradfield and Stirling [1].

3 The Algorithm

Without loss of generality, assume all pushdown commands are $p\,a \rightarrow p'\,\varepsilon$, $p\,a \rightarrow p'\,b$, or $p\,a \rightarrow p'\,bb'$.

The extensions to our earlier work [2] are given in Procedures 1 and 2. We refer the reader to the original article for a description of the notations used.

For a control state p and characters a, b, let $\overline{Pop}(p) = \{\ (p', a') \mid p'\,a' \rightarrow p\,\varepsilon\ \}$, and $\overline{Rew}(p, a) = \{\ (p', a') \mid p'\,a' \rightarrow p\,b\ \}$, $\overline{Push}(p, a, b) = \{\ (p', a') \mid p'\,a' \rightarrow p\,ab\ \}$, and together $Pre(p, a, b) = \overline{Pop}(p) \cup \overline{Rew}(p, a) \cup \overline{Push}(p, a, b)$.

4 Termination

The new procedures defined here add extra cases to the termination proof [2]. We show these cases here and refer the reader to the original article for an explanation of the notation and concepts.

Lemma 1 (Termination). *The algorithm satisfies the following properties.*

1. *Each subroutine introduces a fixed set of new states, independent of the automaton A given as input (but may depend on the other parameters). Transitions are only added to these new states.*
2. *For two input automata A_1 and A_2 (giving valuations of the same environments) such that $A_1 \preceq A_2$, then the returned automata A'_1 and A'_2, respectively, satisfy $A'_1 \preceq A'_2$.*
3. *The algorithm terminates.*

Procedure 1 $BackBox(A, \varphi_1, c, \mathbb{P})$

$((\mathcal{Q}_1, \Sigma, \Delta_1, \_, \mathcal{F}_1), I_1) = Dispatch(A, \varphi_1, c, \mathbb{P})$

$A' = (\mathcal{Q}_1 \cup I \cup \mathcal{Q}_{int}, \Sigma, \Delta_1 \cup \Delta', \_, \mathcal{F}_1)$

where $I = \left\{ \ (p, \overline{\Box}\varphi_1, c) \ \mid p \in \mathcal{P} \ \right\}$

and $\mathcal{Q}_{int} = \left\{ \ (p, \overline{\Box}\varphi_1, c, a) \ \mid p \in \mathcal{P} \land a \in \Sigma \ \right\}$

and $\Delta' =$

$$\left\{ ((p, \overline{\Box}\varphi_1, c), a, Q) \ \middle| \ \begin{array}{c} Q = \{(p, \overline{\Box}\varphi_1, c, a)\} \cup Q_{pop} \cup Q_{rew} \ \land \\ \overline{Pop}(p) = \{(p_1, a_1), \ldots, (p_n, a_n)\} \ \land \\ \bigwedge_{1 \leq j \leq n} \left(I_1(p_j) \xrightarrow[\Delta_1]{a_j} Q'_j \xrightarrow[\Delta_1]{a} Q^{pop}_j \right) \ \land \\ Q_{pop} = Q^{pop}_1 \cup \cdots \cup Q^{pop}_n \ \land \\ \overline{Rew}(p, a) = \{(p'_1, a'_1), \ldots, (p'_{n'}, a'_{n'})\} \ \land \\ \bigwedge_{1 \leq j \leq n'} \left(I_1(p'_j) \xrightarrow[\Delta_1]{a'_j} Q^{rew}_j \right) \ \land \\ Q_{rew} = Q^{rew}_1 \cup \cdots \cup Q^{rew}_n \end{array} \right\} \cup$$

$$\left\{ ((p, \overline{\Box}\varphi_1, c, a), b, Q) \ \middle| \ \begin{array}{c} Pre(p, a, b) = \{(p_1, a_1), \ldots, (p_n, a_n)\} \land \\ \bigwedge_{1 \leq j \leq n} \left(I_1(p_j) \xrightarrow[\Delta_1]{a_j} Q^{push}_j \right) \land \\ Q = Q^{push}_1 \cup \cdots \cup Q^{push}_n \end{array} \right\} \cup$$

$$\left\{ \ ((p, \overline{\Box}\varphi_1, c), a, \{q^*\}) \ \mid \forall b. Pre(p, a, b) = \emptyset \ \right\} \cup$$

$$\left\{ \ ((p, \overline{\Box}\varphi_1, c), \bot, \{q^\varepsilon_f\}) \ \mid \forall a. Pre(p, \bot, a) = \emptyset \ \right\} \cup$$

$$\left\{ \ ((p, \overline{\Box}\varphi_1, c, a), b, \{q^*\}) \ \mid \overline{Push}(p, a, b) = \emptyset \ \right\} \cup$$

$$\left\{ \ ((p, \overline{\Box}\varphi_1, c, a), \bot, \{q^\varepsilon_f\}) \ \mid \overline{Push}(p, a, \bot) = \emptyset \ \right\}$$

return (A', I)

Procedure 2 $BackDiamond(A, \varphi_1, c, \mathbb{P})$

$((\mathcal{Q}_1, \Sigma, \Delta_1, \_, \mathcal{F}_1), I_1) = Dispatch(A, \varphi_1, c, \mathbb{P})$

$A' = (\mathcal{Q}_1 \cup I \cup \mathcal{Q}_{int}, \Sigma, \Delta_1 \cup \Delta', \_, \mathcal{F}_1)$

where $I = \left\{ \ (p, \overline{\Diamond}\varphi_1, c) \ \mid p \in \mathcal{P} \ \right\}$

and $\mathcal{Q}_{int} = \left\{ \ (p, \overline{\Box}\varphi_1, c, a) \ \mid p \in \mathcal{P} \land a \in \Sigma \ \right\}$

$$\text{and } \Delta' = \begin{array}{c} \left\{ ((p, \overline{\Diamond}\varphi_1, c), a, Q) \ \middle| \ \begin{array}{c} (p', a') \in \overline{Pop}(p) \land \\ I_1(p') \xrightarrow[\Delta_1]{a'} Q' \xrightarrow[\Delta_1]{a} Q \end{array} \right\} \cup \\ \left\{ ((p, \overline{\Diamond}\varphi_1, c), a, Q) \ \middle| \ \begin{array}{c} (p', a') \in \overline{Rew}(p, a) \land \\ I_1(p') \xrightarrow[\Delta_1]{a'} Q \end{array} \right\} \cup \\ \{((p, \overline{\Diamond}\varphi_1, c), a, \{(p, \overline{\Diamond}\varphi_1, c, a)\})\} \cup \\ \left\{ ((p, \overline{\Diamond}\varphi_1, c, a), b, Q) \ \middle| \ \begin{array}{c} (p', a') \in \overline{Push}(p, a, b) \land \\ I_1(p') \xrightarrow[\Delta_1]{a'} Q \end{array} \right\} \end{array} .$$

return (A', I)

Proof. The first of these conditions is trivially satisfied by all constructions, hence we omit the proofs. Similarly, termination is trivial. The second and third conditions will be shown by mutual induction over the recursion (structure of the formula). The new cases follow.

Case $BackBox(A, \varphi_1, c, \mathbb{P})$ **and** $BackDiamond(A, \varphi_1, c, \mathbb{P})$:

It can be observed that all new transitions in A are derived from transitions $I(p') \xrightarrow[A]{a} Q$ (or are independent of A and A'). Since $A \preceq A'$ it follows that all transitions have a counterpart $I(p') \xrightarrow[A']{a} Q'$ with $Q' \ll Q$. Hence the property follows in a similar manner to the previous cases.

4.1 Complexity

The new procedures change the complexity of the algorithm slightly, although the algorithm remains in EXPTIME. In particular, the algorithm is now exponential in the number of control states, the size of the stack alphabet and the size of the formula. Let m be the nesting depth of the fixed points of the formula and n be the number of states in A_V. We introduce at most $k = \mathcal{O}\left(|\mathcal{P}| \cdot |\chi| \cdot m \cdot |\Sigma|\right)$ states to the automaton. Hence, there are at most $\mathcal{O}(n + k)$ states in the automaton during any stage of the algorithm. The fixed point computations iterate up to an $\mathcal{O}\left(2^{\mathcal{O}(n+k)}\right)$ number of times. Each iteration has a recursive call, which takes up to $\mathcal{O}\left(2^{\mathcal{O}(n+k)}\right)$ time. Hence the algorithm is $\mathcal{O}\left(2^{\mathcal{O}(n+k)}\right)$ overall.

5 Correctness

We extend the proofs of correctness. We refer the reader to our previous work for the full details [2].

Definition 1 (Correctness Conditions). *The correctness conditions are as follows. Let A be the input automaton, φ be the input formula*[1]*, c be the input level and A' be the result.*

1. *We only introduce level c states.*
2. *If A is V-sound, A' is V^c_φ-sound.*
3. *If A is V-complete, A' is V^c_φ-complete.*

The first condition is obvious. The remaining conditions are shown by induction and require the addition of proof cases for the new procedures.

Lemma 2 (Valuation Soundness). *The algorithm is V-sound.*

[1] For cases such as $And(A, \varphi_1, \varphi_2, c, \mathbb{P})$ we take, as appropriate $\varphi = \varphi_1 \wedge \varphi_2$.

Proof. **Case** $BackBox(A, \varphi_1, c, \mathbb{P})$:

We assume that A is valuation sound with respect to some valuation V. By induction the result A_1 of the recursive call is valuation sound with respect to $V^c_{\varphi_1}$. We show that A' is valuation sound with respect to $V^c_{\overline{\Box}\varphi_1}$.

We observe that no $(p', \overline{\Box}\varphi_1, c)$ are reachable from a state $(p, \overline{\Box}\varphi, c, a)$, hence we show soundness for the latter states first.

The first case is for some b with $\overline{Push}(p, a, b) = \emptyset$. In this case, the valuation of $(p, \overline{\Box}\varphi, c, a)$ contains all words of the form bw. Hence soundness is immediately satisfied.

Otherwise, $\overline{Push}(p, a, b) = \{(p_1, a_1), \ldots, (p_n, a_n)\}$ such that for all $1 \leq j \leq n$, $\langle p_j, a_i w\rangle \hookrightarrow \langle p, abw\rangle$. Take a new transition $((p, \overline{\Box}\varphi_1, c, a), b, Q)$ derived from the runs $I_1(p_j) \xrightarrow[A_1]{a_j} Q_j$ for all $1 \leq j \leq n$, with $Q = Q_1 \cup Q_n$. Suppose for some w, $w \in V^c_{\overline{\Box}\varphi_1}(q)$ for all $q \in Q$. By valuation soundness of A_1 we know $a_j w \in V^c_{\overline{\Box}\varphi_1}(I_1(p_j))$ and hence, since all transitions to $\langle p, abw\rangle$ are from configurations satisfying φ_1, we have $bw \in V^c_{\overline{\Box}\varphi_1}(p, \overline{\Box}\varphi_1, c, a)$ as required.

The remaining states are of the form $(p, \overline{\Box}\varphi_1, c)$. We first deal with the case when for all b we have $Pre(p, a, b) = \emptyset$. In this case, the valuation of $\overline{\Box}\varphi_1$ contains all words of the form aw for some w. Hence, all added transitions are trivially sound.

Otherwise, take a new transition $((p, \overline{\Box}\varphi_1, c), a, Q)$ derived from some b, the value of $\overline{Pop}(p) = \{(p_1, a_1), \ldots, (p_n, a_n)\}$ and for all $1 \leq j \leq n$, the runs $I_1(p_j) \xrightarrow[A_1]{w_j} Q'_j \xrightarrow[A_1]{b} Q^{pop}_j$, with $Q_{pop} = Q^{pop}_1 \cup Q^{pop}_n$, and the value of $\overline{Rew}(p, =)\,\{(p'_1, a'_1), \ldots, (p'_{n'}, a'_{n'})\}$ and for all $1 \leq j \leq n'$, the runs $I_1(p'_j) \xrightarrow[A_1]{a'_j} Q^{rew}_j$, with $Q_{rew} = Q^{rew}_1 \cup Q^{rew}_n$. Finally, $Q = \{(p, \overline{\Box}\varphi_1, c, a, b)\} \cup Q_{pop} \cup Q_{rew}$.

Suppose for some w, $w \in V^c_{\overline{\Box}\varphi_1}(q)$ for all $q \in Q_{pop}$. By valuation soundness of A_1 we know $a_j aw \in V^c_{\overline{\Box}\varphi_1}(I_1(p_j))$ and hence all pop transitions leading to $\langle p, aw\rangle$ are from configurations satisfying φ_1.

Now suppose for some aw, $aw \in V^c_{\overline{\Box}\varphi_1}(q)$ for all $q \in Q_{rew}$. By valuation soundness of A_1 we know $a_j w \in V^c_{\overline{\Box}\varphi_1}(I_1(p_j))$ and hence all rewrite transitions leading to $\langle p, aw\rangle$ are from configurations satisfying φ_1.

Finally, consider some bw in the valuation of $(p, \overline{\Box}\varphi_1, c, a)$. From the soundness of this state, shown above, we have that all push transitions leading to $\langle p, abw\rangle$ are from configurations satisfying φ_1.

Putting the three cases together, we have for all $abw \in V^c_{\overline{\Box}\varphi_1}(p, \overline{\Box}\varphi_1, c)$ as required.

The above cases do not cover the case $\bot \in V^c_{\overline{\Box}\varphi_1}(p, \overline{\Box}\varphi_1, c)$. However, since no push transition can reach this stack, we just require the first two cases and that $(p, \overline{\Box}\varphi_1, c, \bot) = q^{\varepsilon}_f$.

Case $BackDiamond(A, \varphi_1, c, \mathbb{P})$:

We assume that A is valuation sound with respect to some valuation V. By induction the result A_1 of the recursive call is valuation sound with respect to $V^c_{\varphi_1}$. We show that A' is valuation sound with respect to $V^c_{\overline{\Diamond}\varphi_1}$.

We begin with the states $(p, \overline{\Diamond}, c, a)$. Take a transition $((p, \overline{\Diamond}, c, a), b, Q)$. Then there is some $(p', a') \in \overline{Push}(p, a, b)$ such that $I_1(p') \xrightarrow{a'} QA_1$. From the soundness of A_1 we know for all w with $w \in V^c_{\overline{\Diamond}\varphi_1}(q)$ for all $q \in Q$ we have $a'w \in V^c\overline{\Diamond}\varphi_1(I_1(p'))$. Since $\langle p', a'w \rangle \hookrightarrow \langle p, abw \rangle$ we have $\langle p, abw \rangle$ satisfies φ_1 and hence $bw \in V^c_{\overline{\Diamond}\varphi_1}(p, \overline{\Diamond}, c, a)$ and the transition is sound.

For the remaining states, take a new transition $((p, \overline{\Diamond}\varphi_1, c), a, Q)$. There are three cases.

If the transition was derived from some $(p', a') \in \overline{Pop}(p)$ and the run $I_1(p') \xrightarrow[A_1]{a'a} Q$, then suppose for some w, $w \in V^c_{\overline{\Diamond}\varphi_1}(q)$ for all $q \in Q$. By valuation soundness of A_1 we know $a'aw \in V^c_{\overline{\Diamond}\varphi_1}(I_1(p'))$ and hence, since there is a transition $\langle p', a'aw \rangle$, a configuration satisfying φ_1, to $\langle p, aw \rangle$ we obtain $aw \in V^c_{\overline{\Diamond}\varphi_1}(p, \overline{\Diamond}\varphi_1, c)$ as required.

If the transition was derived from some $(p', a') \in \overline{Rew}(p, a)$ and the run $I_1(p') \xrightarrow[A_1]{a'} Q$, then suppose for some w, $w \in V^c_{\overline{\Diamond}\varphi_1}(q)$ for all $q \in Q$. By valuation soundness of A_1 we know $a'w \in V^c_{\overline{\Diamond}\varphi_1}(I_1(p'))$ and hence, since there is a transition $\langle p', a'w \rangle$, a configuration satisfying φ_1, to $\langle p, aw \rangle$ we obtain $aw \in V^c_{\overline{\Diamond}\varphi_1}(p, \overline{\Diamond}\varphi_1, c)$ as required.

Finally, if $Q = \{(p, \overline{\Diamond}, c, a)\}$ then soundness is immediate from the definition of $V^c_{\overline{\Diamond}\varphi_1}$.

Lemma 3 (Valuation Completeness). *The algorithm is V-complete.*

Proof. **Case** $BackBox(A, \varphi_1, c, \mathbb{P})$:

We are given that A is valuation complete with respect to some valuation V, and by induction we have completeness of the result A_1 of the recursive call with respect to $V^c_{\varphi_1}$. We show A' is complete with respect to $V^c_{\overline{\Box}\varphi_1}$.

As in the soundness proof, we begin with the states $(p, \overline{\Box}\varphi_1, c, a)$. In the case $\overline{Push}(p, a, b) = \emptyset$ for some b, we either have $b = \perp$ and the transition from $(p, \Box\varphi_1, c, a)$ to $\left\{q^\varepsilon_f\right\}$ witnesses completeness, or we have $a \neq \perp$ and the transition to $\{q^*\}$ witnesses completeness.

Otherwise $\overline{Push}(p, a, b) = \{(p_1, a_1), \ldots, (p_n, a_n)\}$. Take some bw such that $abw \in V^c_{\overline{\Box}\varphi_1}(p, \overline{\Box}\varphi_1, c, a)$. Then we have $a_j w \in V^c_{\overline{\Box}\varphi_1}(p_j, \varphi_1, c)$ for all $1 \leq j \leq n$. From completeness of A_1 we have a transition $I_1(p_j) \xrightarrow{a_j} Q_j$ with $w \in V^c_{\overline{\Box}\varphi_1}(q)$ for all $q \in Q_j$. Hence, we have a complete b-transition from $(p, \overline{\Box}\varphi_1, c, a)$ as required.

For the states of the form $(p, \overline{\Box}\varphi_1, c)$ we first deal with the case when for all b we have $Pre(p, a, b) = \emptyset$. In this case we immediately have transitions witnessing completeness.

Otherwise, take some $abw \in V^{c}_{\overline{\Box}\varphi_1}(p, \overline{\Box}\varphi_1, c)$. Then, for all $(p', a') \in \overline{Pop}(p)$, we have $a'abw \in V^{c}_{\overline{\Box}\varphi_1}(I_1(p'))$; and for all $(p', a') \in \overline{Rew}(p, a)$ we have $a'bw \in V^{c}_{\overline{\Box}\varphi_1}(I_1(p'))$; and for all $(p', a') \in \overline{Push}(p, a, b)$ we have $a'w \in V^{c}_{\overline{\Box}\varphi_1} I_1(p')$. From completeness of A_1 we have a complete run $I_1(p') \xrightarrow[A_1]{a'} Q' \xrightarrow[A_1]{a} Q$ for each $(p', a') \in \overline{Pop}(p)$ and a complete run $I_1(p') \xrightarrow[A_1]{a'} Q$ for each $(p', a') \in \overline{Rew}(p, a)$. Since we know $bw \in V^{c}_{\overline{\Box}\varphi_1}(p, \overline{\Box}\varphi_1, c, a)$ there must be some complete transition from $(p, \overline{\Box}\varphi_1, c)$ as required.

The only case not covered by the above is the case $\bot \in V^{c}_{\overline{\Box}\varphi_1}(p, \overline{\Box}, \varphi_1, c)$. In this case there are no push transitions reaching this configuration. That is $\overline{Push}(p, \bot, b) = \emptyset$ for all b. Note also that we equated all $(p, \overline{\Box}\varphi_1, c, \bot)$ with q^{ε}_f. Hence, from the pop and rewrite cases above, and that $(p, \overline{\Box}\varphi_1, c, \bot) = q^{\varepsilon}_f$ we have completeness as required.

Case $BackDiamond(A, \varphi_1, c, \mathbb{P})$:

We are given that A is valuation complete with respect to some valuation V, and by induction we have completeness of the result A_1 of the recursive call with respect to $V^{c}_{\varphi_1}$. We show A' is complete with respect to $V^{c}_{\overline{\Diamond}\varphi_1}$. There are three cases.

Assume some aw such that $aw \in V^{c}_{\overline{\Diamond}\varphi_1}(p, \overline{\Diamond}\varphi_1, c)$ by virtue of some $(p', a') \in \overline{Pop}(p)$ such that we have $\langle p', a'aw\rangle \in V^{c}_{\overline{\Diamond}\varphi_1}(I_1(p'))$. By completeness of A_1 we have a run $I_1(p') \xrightarrow[A_1]{a'a} Q$ such hat for all $q \in Q$, $w \in V^{c}_{\overline{\Diamond}\varphi_1}(q)$. Hence, the transition $((p, \overline{\Diamond}\varphi_1, c), a, Q)$ witnesses completeness.

Otherwise, take some aw such that $aw \in V^{c}_{\overline{\Diamond}\varphi_1}(p, \overline{\Diamond}\varphi_1, c)$ from some $(p', a') \in \overline{Rew}(p, a)$ such that we have $\langle p', a'w\rangle \in V^{c}_{\overline{\Diamond}\varphi_1}(I_1(p'))$. By completeness of A_1 we have a run $I_1(p') \xrightarrow[A_1]{a'} Q$ such that for all $q \in Q$, $w \in V^{c}_{\overline{\Diamond}\varphi_1}(q)$. Hence, the transition $((p, \overline{\Diamond}\varphi_1, c), a, Q)$ witnesses completeness.

Finally, take some abw such that $abw \in V^{c}_{\overline{\Diamond}\varphi_1}(p, \overline{\Diamond}\varphi_1, c)$ from some $(p', a') \in \overline{Push}(p, a, b)$ such that we have $\langle p', a'w\rangle \in V^{c}_{\overline{\Diamond}\varphi_1}(I_1(p'))$. By completeness of A_1 we have a run $I_1(p') \xrightarrow[A_1]{a'} Q$ such that for all $q \in Q$, $w \in V^{c}_{\overline{\Diamond}\varphi_1}(q)$. Hence, the transitions $((p, \overline{\Diamond}\varphi_1, c), a, \{(p, \overline{\Diamond}, c, a)\})$ and $((p, \overline{\Diamond}\varphi_1, c, a), a, Q)$ witness completeness.

6 Conclusion and Future Work

In previous work, we have introduced a saturation method for directly computing the denotation of a modal μ-calculus formula over the configuration graph of a pushdown system. Here, we have shown how to extend this work to allow backwards modalities.

References

1. J. C. Bradfield and C. P. Stirling. Modal logics and mu-calculi: An introduction. In *Handbook of Process Algebra*, pages 293–330, 2001.
2. M. Hague and C.-H. L. Ong. A saturation method for the modal mu-calculus over pushdown systems, 2010. To appear in Information and Computation.