Efficient CTL Model-Checking for Pushdown Systems*

Fu Song and Tayssir Touili

LIAFA, CNRS and Univ. Paris Diderot, France.
E-mail: {song,touili}@liafa.jussieu.fr

Abstract. Pushdown systems (PDS) are well adapted to model sequential programs with (possibly recursive) procedure calls. Therefore, it is important to have efficient model checking algorithms for PDSs. We consider in this paper CTL model checking for PDSs. We consider the "standard" CTL model checking problem where whether a configuration of a PDS satisfies an atomic proposition or not depends only on the control state of the configuration. We consider also CTL model checking with regular valuations, where the set of configurations in which an atomic proposition holds is a regular language. We reduce these problems to the emptiness problem in Alternating Büchi Pushdown Systems, and we give an algorithm to solve this emptiness problem. Our algorithms are more efficient than the other existing algorithms for CTL model checking for PDSs in the literature. We implemented our techniques in a tool, and we applied it to different case studies. Our results are encouraging. In particular, we were able to find bugs in linux source code.

1 Introduction

PushDown Systems (PDS for short) are an adequate formalism to model sequential, possibly recursive, programs [EK99,ES01]. It is then important to have verification algorithms for pushdown systems. This problem has been intensively studied by the verification community. Several model-checking algorithms have been proposed for both linear-time logics [BEM97,ES01,EHRS00,FWW97,KPV10], and branching-time logics [BEM97,Boz07,BS97,Wal96,KV00,PV04,FWW97,KPV10].

In this paper, we study the CTL model-checking problem for PDSs. First, we consider the "standard" model-checking problem as was considered in the literature. In this setting, whether a configuration satisfies an atomic proposition or not depends only on the control state of the configuration, not on its stack content. This problem is known to be EXPTIME-complete [Wal00]. CTL corresponds to a fragment of the alternation-free μ-calculus and of CTL*. Existing algorithms for model-checking these logics for PDSs could then be applied for CTL model-checking. However, these algorithms either allow only to decide whether a given configuration satisfies the formula i.e., they cannot compute all the set of PDS configurations where the formula holds [BS95,BS97,Wal96,KV00], or have a high complexity [PV04,Boz07,BEM97,EKS03,EKS01,FWW97,KPV10]. Moreover, these algorithms have not been implemented due to their high complexity. Thus, there does not exist a tool for CTL model-checking of PDSs.

* Work partially funded by ANR grant ANR-08-SEGI-006.

In this work, we propose a new efficient algorithm for CTL-model checking for PDSs. Our algorithm allows to compute the set of PDS configurations that satisfy a given CTL formula. Our procedure is more efficient than the existing model-checking algorithms for μ-calculus and CTL* that are able to compute the set of configurations where a given property holds [PV04,Boz07,BEM97,EKS03,EKS01,FWW97,KPV10]. Our technique reduces CTL model-checking to the problem of computing the set of configurations from which an Alternating Büchi Pushdown System (ABPDS for short) has an accepting run. We show that this set can be effectively represented using an alternating finite automaton.

Then, we consider CTL model checking with regular valuations. In this setting, the set of configurations where an atomic proposition holds is given by a finite state automaton. Indeed, since a configuration of a PDS has a control state and a stack content, it is natural that the validity of an atomic proposition in a configuration depends on both the control state *and the stack*. For example, in one of the case studies we considered, we needed to check that whenever a function *call_hpsb_send_phy_config* is invoked, there is a path where *call_hpsb_send_packet* is called before *call_hpsb_send_phy_config* returns. We need propositions about the stack to express this property. "Standard" CTL is not sufficient. We provide an efficient algorithm that solves CTL model checking with regular valuations for PDSs. Our procedure reduces the model-checking problem to the problem of computing the set of configurations from which an ABPDS has an accepting run.

We implemented our techniques in a tool for CTL model-checking for pushdown systems. Our tool deals with both "standard" model-checking, and model-checking with regular valuations. As far as we know, this is the *first* tool for CTL model-checking for PDSs. Indeed, existing model-checking tools for PDSs like Moped [Sch02] consider only reachability and LTL model-checking, they don't consider CTL. We run several experiments on our tool. We obtained encouraging results. In particular, we were able to find bugs in source files of the linux system, in a watchdog driver of linux, and in an IEEE 1394 driver of linux. We needed regular valuations to express the properties of some of these examples.

Outline. The rest of the paper is structured as follows. Section 2 gives the basic definitions used in the paper. In section 3, we present an algorithm for computing an alternating automaton recognizing all the configurations from which an ABPDS has an accepting run. Sections 4 and 5 describe the reductions from "standard" CTL model-checking for PDSs and CTL model-checking for PDSs with regular valuations, to the emptiness problem in ABPDS. The experiments are provided in Section 6. Section 7 describes the related work.

2 Preliminaries

2.1 The temporal logic CTL

We consider the standard branching-time temporal logic CTL. For technical reasons, we use the operator $\tilde{U}$ as a dual of the until operator for which the stop condition is not required to occur; and we suppose w.l.o.g. that formulas are given in positive normal form, i.e., negations are applied only to atomic propositions. Indeed, each CTL formula can be written in positive normal form by pushing the negations inside.

Definition 1. *Let* AP $= \{a, b, c, ...\}$ *be a finite set of atomic propositions. The set of CTL formulas is given by (where* $a \in$ AP*):*

$$\varphi ::= a \mid \neg a \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid AX\varphi \mid EX\varphi \mid A[\varphi U \varphi] \mid E[\varphi U \varphi] \mid A[\varphi \tilde{U} \varphi] \mid E[\varphi \tilde{U} \varphi].$$

The closure $cl(\varphi)$ of a CTL formula φ is the set of all the subformulas of φ, including φ. Let $AP^+(\varphi) = \{a \in AP \mid a \in cl(\varphi)\}$ and $AP^-(\varphi) = \{a \in AP \mid \neg a \in cl(\varphi)\}$. The size $|\varphi|$ of φ is the number of elements in $cl(\varphi)$. Let $T = (S, \longrightarrow, c_0)$ be a transition system where S is a set of states, $\longrightarrow \subseteq S \times S$ is a set of transitions, and c_0 is the initial state. Let $s, s' \in S$. s' is a successor of s iff $s \longrightarrow s'$. A path is a sequence of states $s_0, s_1, \ldots$ such that for every $i \geq 0$, $s_i \longrightarrow s_{i+1}$. Let $\lambda : AP \to 2^S$ be a labelling function that assigns to each atomic proposition a set of states in S. The validity of a formula φ in a state s w.r.t. the labelling function λ, denoted $s \models_\lambda \varphi$, is defined inductively in **Figure** 1. $T \models_\lambda \varphi$ iff $c_0 \models_\lambda \varphi$. Note that a path π satisfies $\psi_1 \tilde{U} \psi_2$ iff either ψ_2 holds everywhere in π, or the first occurrence in the path where ψ_2 does not hold must be proceeded by a position where ψ_1 holds.

$s \models_\lambda a$	$\Longleftrightarrow$	$s \in \lambda(a)$.
$s \models_\lambda \neg a$	$\Longleftrightarrow$	$s \notin \lambda(a)$.
$s \models_\lambda \psi_1 \wedge \psi_2$	$\Longleftrightarrow$	$s \models_\lambda \psi_1$ and $s \models_\lambda \psi_2$.
$s \models_\lambda \psi_1 \vee \psi_2$	$\Longleftrightarrow$	$s \models_\lambda \psi_1$ or $s \models_\lambda \psi_2$.
$s \models_\lambda AX\,\psi$	$\Longleftrightarrow$	$s' \models_\lambda \psi$ for every successor s' of s.
$s \models_\lambda EX\,\psi$	$\Longleftrightarrow$	There exists a successor s' of s s.t. $s' \models_\lambda \psi$.
$s \models_\lambda A[\psi_1 U \psi_2]$	$\Longleftrightarrow$	For every path of $T, \pi = s_0, s_1, ...,$ with $s_0 = s, \exists i \geq 0$ s.t. $s_i \models_\lambda \psi_2$ and $\forall 0 \leq j < i, s_j \models_\lambda \psi_1$.
$s \models_\lambda E[\psi_1 U \psi_2]$	$\Longleftrightarrow$	There exists a path of $T, \pi = s_0, s_1, ...,$ with $s_0 = s,$ s.t. $\exists i \geq 0, s_i \models_\lambda \psi_2$ and $\forall 0 \leq j < i, s_j \models_\lambda \psi_1$.
$s \models_\lambda A[\psi_1 \tilde{U} \psi_2]$	$\Longleftrightarrow$	For every path of $T, \pi = s_0, s_1, ...,$ with $s_0 = s, \forall i \geq 0$ s.t. $s_i \not\models_\lambda \psi_2$, $\exists 0 \leq j < i,$ s.t. $s_j \models_\lambda \psi_1$.
$s \models_\lambda E[\psi_1 \tilde{U} \psi_2]$	$\Longleftrightarrow$	There exists a path of $T, \pi = s_0, s_1, ...,$ with $s_0 = s,$ s.t. $\forall i \geq 0$ s.t. $s_i \not\models_\lambda \psi_2$, $\exists 0 \leq j < i$ s.t. $s_j \models_\lambda \psi_1$.

Fig. 1. Semantics of CTL

2.2 PushDown Systems

Definition 2. *A PushDown System* (PDS for short) *is a tuple* $\mathcal{P} = (P, \Gamma, \Delta, \sharp)$*, where* P *is a finite set of control locations,* Γ *is the stack alphabet,* $\Delta \subseteq (P \times \Gamma) \times (P \times \Gamma^*)$ *is a finite set of transition rules and* $\sharp \in \Gamma$ *is a bottom stack symbol.*

A configuration of $\mathcal{P}$ is an element $\langle p, \omega \rangle$ of $P \times \Gamma^*$. We write $\langle p, \gamma \rangle \hookrightarrow \langle q, \omega \rangle$ instead of $((p, \gamma), (q, \omega)) \in \Delta$. For technical reasons, we consider the bottom stack symbol $\sharp$, and we assume w.l.o.g. that it is never popped from the stack, i.e., there is no transition rule of the form $\langle p, \sharp\rangle \hookrightarrow \langle q, \omega\rangle \in \Delta$. The successor relation $\rightsquigarrow_{\mathcal{P}} \subseteq (P \times \Gamma^*) \times (P \times \Gamma^*)$ is defined as follows: if $\langle p, \gamma\rangle \hookrightarrow \langle q, \omega\rangle$, then $\langle p, \gamma\omega'\rangle \rightsquigarrow_{\mathcal{P}} \langle q, \omega\omega'\rangle$ for every $\omega' \in \Gamma^*$.

Let c be a given initial configuration of $\mathcal{P}$. Starting from c, $\mathcal{P}$ induces the transition system $T^c_{\mathcal{P}} = (P \times \Gamma^*, \rightsquigarrow_{\mathcal{P}}, c)$. Let AP be a set of atomic propositions, φ be a CTL formula on AP, and $\lambda : AP \to 2^{P\times\Gamma^*}$ be a labelling function. We say that $(\mathcal{P}, c) \models_\lambda \varphi$ iff $T^c_{\mathcal{P}} \models_\lambda \varphi$.

2.3 Alternating Büchi PushDown Systems

Definition 3. *An Alternating Büchi PushDown System* (ABPDS for short) *is a tuple $\mathcal{BP} = (P, \Gamma, \Delta, F)$, where P is a finite set of control locations, Γ is the stack alphabet, $F \subseteq P$ is a finite set of accepting control locations and Δ is a function that assigns to each element of $P \times \Gamma$ a* positive boolean formula *over $P \times \Gamma^*$.*

A configuration of an ABPDS is a pair $\langle p, \omega\rangle$, where $p \in P$ is a control location and $\omega \in \Gamma^*$ is the stack content. We assume w.l.o.g. that the boolean formulas are in disjunctive normal form. This allows to consider Δ as a subset of $(P \times \Gamma) \times 2^{P\times\Gamma^*}$. Thus, rules of Δ of the form[1] $\langle p, \gamma\rangle \hookrightarrow \bigvee_{j=1}^{n} \bigwedge_{i=1}^{m_j} \langle p_i^j, \omega_i^j\rangle$ can be denoted by the union of n rules of the form $\langle p, \gamma\rangle \hookrightarrow \{\langle p_1^j, \omega_1^j\rangle, ..., \langle p_{m_j}^j, \omega_{m_j}^j\rangle\}$, where $1 \le j \le n$. Let $t = \langle p, \gamma\rangle \hookrightarrow \{\langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle\}$ be a rule of Δ. For every $\omega \in \Gamma^*$, the configuration $\langle p, \gamma\omega\rangle$ (resp. $\{\langle p_1, \omega_1\omega\rangle, ..., \langle p_n, \omega_n\omega\rangle\}$) is an immediate predecessor (resp. successor) of $\{\langle p_1, \omega_1\omega\rangle, ..., \langle p_n, \omega_n\omega\rangle\}$ (resp. $\langle p, \gamma\omega\rangle$).

A run ρ of $\mathcal{BP}$ from an initial configuration $\langle p_0, \omega_0\rangle$ is a tree in which the root is labeled by $\langle p_0, \omega_0\rangle$, and the other nodes are labeled by elements of $P \times \Gamma^*$. If a node of ρ is labeled by $\langle p, \omega\rangle$ and has n children labeled by $\langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle$, respectively, then necessarily, $\langle p, \omega\rangle$ has $\{\langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle\}$ as an immediate successor in $\mathcal{BP}$. A path $c_0c_1...$ of a run ρ is an *infinite* sequence of configurations such that c_0 is the root of ρ and for every $i \ge 0$, c_{i+1} is one of the children of the node c_i in ρ. The path is accepting from the initial configuration c_0 if and only if it visits infinitely often configurations with control locations in F. A run ρ is accepting if and only if all its paths are accepting. Note that an accepting run has only *infinite* paths; it does not involve finite paths. A configuration c is accepted (or recognized) by $\mathcal{BP}$ iff $\mathcal{BP}$ has an accepting run starting from c. The language of $\mathcal{BP}$, $\mathcal{L}(\mathcal{BP})$ is the set of configurations accepted by $\mathcal{BP}$.

The reachability relation $\Longrightarrow_{\mathcal{BP}} \subseteq (P\times\Gamma^*)\times 2^{P\times\Gamma^*}$ is the reflexive and transitive closure of the immediate successor relation. Formally $\Longrightarrow_{\mathcal{BP}}$ is defined as follows: (1) $c \Longrightarrow_{\mathcal{BP}} \{c\}$ for every $c \in P \times \Gamma^*$, (2) $c \Longrightarrow_{\mathcal{BP}} C$ if C is an immediate successor of c, (3) if $c \Longrightarrow_{\mathcal{BP}} \{c_1, ..., c_n\}$ and $c_i \Longrightarrow_{\mathcal{BP}} C_i$ for every $1 \le i \le n$, then $c \Longrightarrow_{\mathcal{BP}} \bigcup_{i=1}^{n} C_i$.

The functions $Pre_{\mathcal{BP}}$, $Pre^*_{\mathcal{BP}}$ and $Pre^+_{\mathcal{BP}}$: $2^{P\times\Gamma^*} \longrightarrow 2^{P\times\Gamma^*}$ are defined as follows: $Pre_{\mathcal{BP}}(C) = \{c \in P\times\Gamma^* \mid \exists C' \subseteq C$ s.t. C' is an immediate successor of $c\}$, (2) $Pre^*_{\mathcal{BP}}(C) = \{c \in P \times \Gamma^* | \exists C' \subseteq C\ s.t.\ c \Longrightarrow_{\mathcal{BP}} C'\}$, (3) $Pre^+_{\mathcal{BP}}(C) = Pre_{\mathcal{BP}} \circ Pre^*_{\mathcal{BP}}(C)$.

To represent (infinite) sets of configurations of ABPDSs, we use Alternating Multi-Automata:

Definition 4. *[BEM97] Let $\mathcal{BP} = (P, \Gamma, \Delta, F)$ be an ABPDS. An Alternating Multi-Automaton* (AMA for short) *is a tuple $\mathcal{A} = (Q, \Gamma, \delta, I, Q_f)$, where Q is a finite set of*

[1] This rule represents $\Delta(p, \gamma) = \bigvee_{j=1}^{n} \bigwedge_{i=1}^{m_j} (p_i^j, \omega_i^j)$.

states that contains P, Γ is the input alphabet, $\delta \subseteq (Q \times \Gamma) \times 2^Q$ is a finite set of transition rules, $I \subseteq P$ is a finite set of initial states, $Q_f \subseteq Q$ is a finite set of final states.

A Multi-Automaton (MA for short) *is an AMA such that $\delta \subseteq (Q \times \Gamma) \times Q$.*

We define the reflexive and transitive transition relation $\longrightarrow_\delta \subseteq (Q \times \Gamma^*) \times 2^Q$ as follows: (1) $q \xrightarrow{\epsilon}_\delta \{q\}$ for every $q \in Q$, where ϵ is the empty word, (2) $q \xrightarrow{\gamma}_\delta Q'$, if $q \xrightarrow{\gamma} Q' \in \delta$, (3) if $q \xrightarrow{\omega}_\delta \{q_1, ..., q_n\}$ and $q_i \xrightarrow{\gamma}_\delta Q_i$ for every $1 \leq i \leq n$, then $q \xrightarrow{\omega\gamma}_\delta \bigcup_{i=1}^n Q_i$. The automaton $\mathcal{A}$ recognizes a configuration $\langle p, \omega \rangle$ iff there exists $Q' \subseteq Q_f$ such that $p \xrightarrow{\omega}_\delta Q'$ and $p \in I$. The language of $\mathcal{A}$, $L(\mathcal{A})$, is the set of configurations recognized by $\mathcal{A}$. A set of configurations is regular if it can be recognized by an AMA. It is easy to show that AMAs are closed under boolean operations and that they are equivalent to MAs. Given an AMA, one can compute an equivalent MA by performing a kind of powerset construction as done for the determinisation procedure. Similarly, MAs can also be used to recognize (infinite) regular sets of configurations for PDSs.

Proposition 1. *Let $\mathcal{A} = (Q, \Gamma, \delta, I, Q_f)$ be an AMA. Deciding whether a configuration $\langle p, \omega \rangle$ is accepted by $\mathcal{A}$ can be done in $O(|Q| \cdot |\delta| \cdot |\omega|)$ time.*

3 Computing the language of an ABPDS

Our goal in this section is to compute the set of accepting configurations of an Alternating Büchi PushDown System $\mathcal{BP} = (P, \Gamma, \Delta, F)$. We show that it is regular and that it can effectively be represented by an AMA. Determining whether $\mathcal{BP}$ has an accepting run is a non-trivial problem because a run of $\mathcal{BP}$ is an *infinite* tree with an infinite number of paths labelled by PDS configurations, which are control states and stack contents. All the paths of an accepting run are infinite and should all go through final control locations infinitely often. The difficulty comes from the fact that we cannot reason about the different paths of an ABPDS independently, we need to reason about *runs labeled with PDS configurations*. We proceed as follows: First, we characterize the set of configurations from which $\mathcal{BP}$ has an accepting run. Then, based on this characterization, we compute an AMA representing this set.

3.1 Characterizing $\mathcal{L}(\mathcal{BP})$

We give in this section a characterization of $\mathcal{L}(\mathcal{BP})$, i.e., the set of configurations from which $\mathcal{BP}$ has an accepting run. Let $(X_i)_{i \geq 0}$ be the sequence defined as follows: $X_0 = P \times \Gamma^*$ and $X_{i+1} = Pre^+(X_i \cap F \times \Gamma^*)$ for every $i \geq 0$. Let $Y_{\mathcal{BP}} = \bigcap_{i \geq 0} X_i$. We show that $\mathcal{L}(\mathcal{BP}) = Y_{\mathcal{BP}}$:

Theorem 1. *$\mathcal{BP}$ has an accepting run from a configuration $\langle p, \omega \rangle$ iff $\langle p, \omega \rangle \in Y_{\mathcal{BP}}$.*

To prove this result, we first show that:

Lemma 1. *$\mathcal{BP}$ has a run ρ from a configuration $\langle p, \omega \rangle$ such that each path of ρ visits configurations with control locations in F at least k times iff $\langle p, \omega \rangle \in X_k$.*

Intuitively, let c be a configuration in X_1. Since $X_1 = Pre^+(X_0 \cap F \times \Gamma^*)$, c has a successor C that is a subset of $F \times \Gamma^*$. Thus, $\mathcal{BP}$ has a run starting from c whose paths visit configurations with control locations in F at least once. Since $X_2 = Pre^+(X_1 \cap F \times \Gamma^*)$, it follows that from every configuration in X_2, $\mathcal{BP}$ has a run whose paths visit configurations in $X_1 \cap F \times \Gamma^*$ at least once, and thus, they visit configurations with control locations in F at least twice. We get by induction that for every $k \geq 1$, from every configuration c in X_k, $\mathcal{BP}$ has a run whose paths visit configurations with control locations in F at least k times. Since $Y_{\mathcal{BP}}$ is the set of configurations from which $\mathcal{BP}$ has a run that visits control locations in F infinitely often, Theorem 1 follows.

3.2 Computing $\mathcal{L}(\mathcal{BP})$

Our goal is to compute $Y_{\mathcal{BP}} = \bigcap_{i \geq 0} X_i$, where $X_0 = P \times \Gamma^*$ and for every $i \geq 0$, $X_{i+1} = Pre^+(X_i \cap F \times \Gamma^*)$. We provide a saturation procedure that computes the set $Y_{\mathcal{BP}}$. Our procedure is inspired from the algorithm given in [Cac02a] to compute the winning region of a Büchi game on a pushdown graph.

We show that $Y_{\mathcal{BP}}$ can be represented by an AMA $\mathcal{A} = (Q, \Gamma, \delta, I, Q_f)$ whose set of states Q is a subset of $P \times \mathbb{N} \cup \{q_f\}$, where q_f is a special state denoting the final state ($Q_f = \{q_f\}$). From now on, for every $p \in P$ and $i \in \mathbb{N}$, we write p^i to denote (p, i).

Intuitively, to compute $Y_{\mathcal{BP}}$, we will compute iteratively the different X_i's by applying the saturation procedure of [BEM97]. The iterative procedure computes different automata. The automaton computed during the iteration i uses states of the form p^i having i as index. To force termination, we use an acceleration criterion. For this, we need to define two projection functions π^{-1} and π^i defined as follows: For every $S \subseteq P \times \mathbb{N} \cup \{q_f\}$,

$$\pi^{-1}(S) = \begin{cases} \{q^i \mid q^{i+1} \in S\} \cup \{q_f\} & \text{if } q_f \in S \text{ or } \exists q^1 \in S, \\ \{q^i \mid q^{i+1} \in S\} & \text{else.} \end{cases}$$

$$\pi^i(S) = \{q^i \mid \exists 1 \leq j \leq i \ s.t.\ q^j \in S\} \cup \{q_f \mid q_f \in S\}.$$

The AMA $\mathcal{A}$ is computed iteratively using **Algorithm 1**:

Algorithm 1: Computation of $Y_{\mathcal{BP}}$

Input: An ABPDS $\mathcal{BP} = (P, \Gamma, \Delta, F)$.

Output: An AMA $\mathcal{A} = (Q, \Gamma, \delta, I, Q_f)$ that recognizes $Y_{\mathcal{BP}}$.

1.**Initially:** Let $i = 0, \delta = \{(q_f, \gamma, \{q_f\})$ for every $\gamma \in \Gamma\}$, and for every control state $p \in P$, $p^0 = q_f$.
2. **Repeat** (we call this loop $loop_1$)
3. $i := i + 1$;
4. Add in δ a new transition rule $p^i \xrightarrow{\epsilon} p^{i-1}$, for every $p \in F$;
5. **Repeat** (we call this loop $loop_2$)
6. For every $\langle p, \gamma \rangle \hookrightarrow \{\langle p_1, \omega_1 \rangle, ..., \langle p_n, \omega_n \rangle\}$ in Δ
7. and every case where $p_k^i \xrightarrow{\omega_k}_\delta Q_k$, for every $1 \leq k \leq n$;
8. Add a new rule $p^i \xrightarrow{\gamma} \bigcup_{k=1}^n Q_k$ in δ;
9. **Until** No new transition rule can be added.
10. Remove from δ the transition rules $p^i \xrightarrow{\epsilon} p^{i-1}$, for every $p \in F$;
11. Replace in δ every transition rule $p^i \xrightarrow{\gamma} R$ by $p^i \xrightarrow{\gamma} \pi^i(R)$, for every $p \in P$, $\gamma \in \Gamma$, $R \subseteq Q$;
12. **Until** $i > 1$ and for every $p \in P$, $\gamma \in \Gamma$, $R \subseteq P \times \{i\} \cup \{q_f\}$; $p^i \xrightarrow{\gamma} R \in \delta \Longleftrightarrow p^{i-1} \xrightarrow{\gamma} \pi^{-1}(R) \in \delta$

Let us explain the intuition behind the different lines of this algorithm. Let A_i be the automaton obtained at step i (a step starts at Line 3). For every $p \in P$, the state p^i is meant to represent state p at step i, i.e., A_i recognizes a configuration $\langle p, \omega \rangle$ iff $p^i \xrightarrow{\omega}_\delta q_f$. Let A_0 be the automaton obtained after the initialization step (Line 1). It is clear that A_0 recognizes $X_0 = P \times \Gamma^*$. Suppose now that the algorithm is at the beginning of the i-th iteration ($loop_1$). Line 4 adds the ϵ-transition $p^i \xrightarrow{\epsilon} p^{i-1}$ for every control state $p \in F$. After this step, we obtain $L(A_{i-1}) \cap F \times \Gamma^*$. $loop_2$ at lines $5-9$ is the saturation procedure of [BEM97]. It computes the Pre^* of $L(A_{i-1}) \cap F \times \Gamma^*$. Line 10 removes the ϵ-transition added by Line 4. After this step, the automaton recognizes $Pre^+(L(A_{i-1}) \cap F \times \Gamma^*)$, i.e., X_i. Let us call **Algorithm B** the above algorithm without Line 11. It follows from the explanation above that if **Algorithm B** terminates, it will produce $Y_{\mathcal{BP}}$. However, this procedure will never terminate if the sequence (X_i) is strictly decreasing. Consider for example the ABPDS $\mathcal{BP} = (\{q\}, \{\gamma\}, \Delta, \{q\})$, where $\Delta = \{\langle q, \gamma \rangle \hookrightarrow \langle q, \epsilon \rangle\}$. Then, for every $i \geq 0$, $X_i = \{\langle q, \gamma^i \omega \rangle \mid \omega \in \gamma^*\}$. It is clear that **Algorithm B** will never terminate on this example.

The substitution at Line 11 is the acceleration used to force the termination of the algorithm, tested at Line 12. We can show that thanks to Line 11 and to the test of Line 12, our algorithm always terminates and produces $Y_{\mathcal{BP}}$:

Theorem 2. ***Algorithm*** *1 always terminates and produces $Y_{\mathcal{BP}}$.*

Proof (Sketch): Termination. Let us first prove the termination of our procedure. Note that due to the substitution of Line 11, at the end of step i, states with index $j < i$ are not useful and can be removed. We can then suppose that at the end of step i, the automaton A_i uses only states of index i (in addition to state q_f). Thus, the termination tested at Line 12 holds when at step i, the transitions of A_i are "the same" than those of A_{i-1}.

We can show that at each step i, $loop_2$ (corresponding to the saturation procedure) adds less transitions than at step $i-1$, meaning that A_i has less transitions than A_{i-1}. Intuitively, this is due to the fact that at step i, we obtain after the saturation procedure $Pre^+(L(A_{i-1}) \cap F \times \Gamma^*)$. Since Pre^+ is monotonic, and since we start at step 0 with an automaton A_0 that recognizes all the configurations $P \times \Gamma^*$, we get that for $i > 0$, $L(A_i) \subseteq L(A_{i-1})$. More precisely, we can show by induction on i that:

Proposition 2. *In **Algorithm** 1, for every $\gamma \in \Gamma$, $p \in P, S \subseteq Q$; at each step $i \geq 2$, if $p^i \xrightarrow{\gamma} S \in \delta$, then $p^{i-1} \xrightarrow{\gamma} \pi^{-1}(\pi^i(S)) \in \delta$.*

Thus, the substitution of Line 11 guarantees that at each step, the number of transitions of the automaton A_i is less than the number of transitions of A_{i-1}. Since the number of transitions that can be added at each step is finite, and since the termination criterion of Line 12 holds if the transitions of A_i are "the same" than those of A_{i-1}, the termination of our algorithm is guaranteed.

Correctness. Let us now prove that our algorithm is correct, i.e., it produces $Y_{\mathcal{BP}}$. As mentionned previously, without Line 11, the algorithm above would have computed the different X_i's. Since $Y_{\mathcal{BP}} = \bigcap_{i \geq 0} X_i$, we need to show that Line 11 does not introduce new configurations that are not in $Y_{\mathcal{BP}}$, nor remove ones that should be in $Y_{\mathcal{BP}}$.

Suppose we are at step i, and let $p \in P$, $\gamma \in \Gamma$, and $R \subseteq Q$ be such that Line 11 adds the transition $p^i \xrightarrow{\gamma} \pi^i(R)$ and removes the transition $p^i \xrightarrow{\gamma} R$. This substitution adds a new transition iff R contains at least one state of the form q^{i-1} (otherwise, $\pi^i(R) = R$ and Line 11 does not introduce any change for this transition). Let then $S \subseteq Q$ be such $R = S \cup \{q^{i-1}\}$. Let us first show that this substitution does not introduce new configurations. Let $u \in \Gamma^*$ such that $p^i \xrightarrow{\gamma}_{\delta} \pi^i(R) \xrightarrow{u}_{\delta} q_f$ is a new accepting run of the automaton. Then, due to Proposition 2, we can show that there exists already (before the substitution) a run $p^i \xrightarrow{\gamma}_{\delta} R \xrightarrow{u}_{\delta} q_f$ in the automaton that accepts the configuration $\langle p, \gamma u \rangle$.

Let us now show that the substitution above does not remove configurations that are in $Y_{\mathcal{BP}}$. Let $\langle p, \omega \rangle$ be a configuration removed by the substitution above, i.e., $\langle p, \omega \rangle$ is no more recognized by A_i due to the fact that $p^i \xrightarrow{\gamma} R$ is removed. We show that $\langle p, \omega \rangle$ cannot be in $Y_{\mathcal{BP}}$. Let $v \in \Gamma^*$ such that $\omega = \gamma v$ and $\rho = p^i \xrightarrow{\gamma}_{\delta} q^{i-1} \cup S \xrightarrow{v}_{\delta} \{q_f\}$ is a run accepting $\langle p, \omega \rangle$ whereas there is no run of the form $q^i \xrightarrow{v}_{\delta} \{q_f\}$. Suppose for simplicity that ρ is the only run recognizing $\langle p, \omega \rangle$, the same reasoning can also be applied if this is not the case. Since $p^i \xrightarrow{\gamma} q^{i-1} \cup S$, we can show that there exist states $q_1, \ldots, q_n$, and words $\omega_1, \ldots, \omega_n$ such that $\langle p, \gamma \rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon \rangle, \langle q_1, \omega_1 \rangle, \cdots \langle q_n, \omega_n \rangle\}$. Then, due to the fact that $\langle p, \omega \rangle$ is removed from the automaton and that ρ is the only path accepting $\langle p, \omega \rangle$, we can show that all the possible runs from the configuration $\langle p, \omega \rangle$ go through the configuration $\langle q, v \rangle$. Since $\langle q, v \rangle \notin Y_{\mathcal{BP}}$ (because there is no run of the form $q^i \xrightarrow{v}_{\delta} \{q_f\}$), $\mathcal{BP}$ has no accepting run from the configuration $\langle q, v \rangle$. It follows that $\mathcal{BP}$ cannot have an accepting run from $\langle p, \omega \rangle$.

□

Complexity: Given an AMA A with n states, [SSE06] provides a procedure that can implement the saturation procedure $loop_2$ to compute the Pre^* of A in time $O(n \cdot |\Delta| \cdot 2^{2n})$. Since at each step i, **Algorithm 1** needs to consider only states of the form p^i and p^{i-1} (in addition to q_f), the number of states at each step i should be $2|P| + 1$. Thus, $loop_2$ can be done in $O(|P| \cdot |\Delta| \cdot 2^{4|P|})$. Furthermore, Line 11 and the termination condition are done in time $O(|\Gamma| \cdot |P| \cdot 2^{2|P|})$ and $O(|\Gamma| \cdot |P| \cdot 2^{|P|})$, respectively. We know that the number of transition rules of A_i is less than those of A_{i-1}. Since the number of transition rules of the AMA is at most $|\Gamma| \cdot |P| \cdot 2^{|P|+1}$, $loop_1$ can be done at most $|\Gamma| \cdot |P| \cdot 2^{|P|+1}$ times. Putting all these estimations together, the algorithm runs in $O(|P|^2 \cdot |\Delta| \cdot |\Gamma| \cdot 2^{5|P|})$ time.

Thus, since $\mathcal{L}(\mathcal{BP}) = Y_{\mathcal{BP}}$, we get :

Theorem 3. *Given an ABPDS $\mathcal{BP} = (P, \Gamma, \Delta, F)$, we can effectively compute an AMA $\mathcal{A}$ with $O(|P|)$ states and $O(|P| \cdot |\Gamma| \cdot 2^{|P|})$ transition rules that recognizes $\mathcal{L}(\mathcal{BP})$. This AMA can be computed in time $O(|P|^2 \cdot |\Delta| \cdot |\Gamma| \cdot 2^{5|P|})$.*

Example: Let us illustrate our algorithm by an example. Consider an ABPDS $\mathcal{BP} = (\{q\}, \{\gamma\}, \Delta, \{q\})$, where $\Delta = \{\langle q, \gamma\rangle \hookrightarrow \langle q, \epsilon\rangle\}$. The automaton produced by **Algorithm** 1 is shown in Figure 2. The dashed lines denote the transitions removed by Lines 10 and 11. In the first iteration, $t_1 = q^1 \xrightarrow{\epsilon} q_f$ is added by Line 4, the saturation procedure (lines 5 – 9) adds two transitions $q^1 \xrightarrow{\gamma} q_f$ and $q^1 \xrightarrow{\gamma} q^1$. Then the transition t_1 is removed by Line 10. In the second iteration, $t_2 = q^2 \xrightarrow{\epsilon} q^1$ is added by Line 4. The saturation procedure adds the transitions $t_3 = q^2 \xrightarrow{\gamma} q^1$ and $q^2 \xrightarrow{\gamma} q^2$. Finally, t_2 is removed by Line 10 and t_3 is replaced by $q^2 \xrightarrow{\gamma} q^2$ (this transition already exists in the automaton). Now the termination condition is satisfied and the algorithm terminates. In this case, $\mathcal{BP}$ has no accepting run.

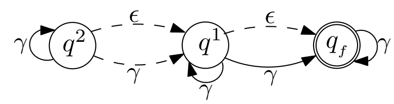

Fig. 2: The result automaton.

Efficient implementation of Algorithm 1. We show that we can improve the complexity of **Algorithm** 1 as follows:

Improvement 1. For every $q \in Q$ and $\gamma \in \Gamma$, if $t_1 = q \xrightarrow{\gamma} Q_1$ and $t_2 = q \xrightarrow{\gamma} Q_2$ are two transitions in δ such that $Q_1 \subseteq Q_2$, then remove t_2. This means that if $\mathcal{A}$ contains two transitions $t_1 = p \xrightarrow{\gamma} \{q_1, q_2, q_3\}$ and $t_2 = p \xrightarrow{\gamma} \{q_1, q_2\}$, then we can remove t_1 without changing the language of $\mathcal{A}$. Indeed, if a path $q \xrightarrow{\omega}_\delta q_f$ uses the transition rule t_1, then there must be necessarily a path $q \xrightarrow{\omega}_\delta q_f$ that uses the transition rule t_2 instead of t_1.

Improvement 2. Each transition $q^i \xrightarrow{\gamma} R$ added by the saturation procedure will be substituted by $q^i \xrightarrow{\gamma} \pi^i(R)$ in Line 11. Transitions of the form $q^i \xrightarrow{\gamma} \{q^i_1, q^{i-1}_1\} \cup R$ and $q^i \xrightarrow{\gamma} \{q^{i-1}_1\} \cup R$ have the same substitution $q^i \xrightarrow{\gamma} \{q^i_1\} \cup \pi^i(R)$. We show that each transition $q^i \xrightarrow{\gamma} \{q^i_1, q^{i-1}_1\} \cup R$ can be replaced by $q^i \xrightarrow{\gamma} \{q^{i-1}_1\} \cup R$ in the saturation procedure (i.e., during $loop_2$). Moreover, we show that if both $t_1 = q^i \xrightarrow{\gamma} \{q^{i-1}_1, ..., q^{i-1}_n\} \cup R$ and $t_2 = q^i \xrightarrow{\gamma} \{q^i_1, ..., q^i_n\} \cup R$ exist during $loop_2$, then t_2 can be removed. This is due to the fact that they both have the same substitution rule.

4 CTL Model-Checking for PushDown Systems

We consider in this section "standard" CTL model checking for pushdown systems as considered in the literature, i.e., the case where whether an atomic proposition holds for a given configuration c or not depends only on the control state of c, not on its stack. Let $\mathcal{P} = (P, \Gamma, \Delta, \sharp)$ be a pushdown system, c_0 its initial configuration, AP a set of atomic propositions, φ a CTL formula, $f : AP \to 2^P$ a function that associates atomic propositions to sets of control states, and $\lambda_f : AP \to 2^{P\times\Gamma^*}$ a labelling function such that for every $a \in AP$, $\lambda_f(a) = \{\langle p, \omega\rangle \mid p \in f(a), \omega \in \Gamma^*\}$. We provide in this section an algorithm to determine whether $(\mathcal{P}, c_0) \models_{\lambda_f} \varphi$. We proceed as follows: Roughly speaking, we compute an Alternating Büchi PushDown System $\mathcal{BP}$ that recognizes the set of configurations c such that $(\mathcal{P}, c) \models_{\lambda_f} \varphi$. Then $(\mathcal{P}, c_0) \models_{\lambda_f} \varphi$ holds iff $c_0 \in \mathcal{L}(\mathcal{BP})$. This can be effectively checked due to Theorem 3 and Proposition 1.

Let $\mathcal{BP}_\varphi = (P', \Gamma, \Delta', F)$ be the ABPDS defined as follows: $P' = P\times cl(\varphi)$; $F = \{[p, a] \mid a \in cl(\varphi) \cap AP \text{ and } p \in f(a)\} \cup \{[p, \neg a] \mid \neg a \in cl(\varphi), a \in AP \text{ and } p \notin f(a)\} \cup P \times cl_{\tilde{U}}(\varphi)$, where $cl_{\tilde{U}}(\varphi)$ is the set of formulas of $cl(\varphi)$ of the form $E[\varphi_1\tilde{U}\varphi_2]$ or $A[\varphi_1\tilde{U}\varphi_2]$; and Δ' is the smallest set of transition rules such that for every control location $p \in P$, every subformula $\psi \in cl(\varphi)$, and every $\gamma \in \Gamma$, we have:

1. if $\psi = a$, $a \in AP$ and $p \in f(a)$; $\langle[p,\psi],\gamma\rangle \hookrightarrow \langle[p,\psi],\gamma\rangle \in \Delta'$,
2. if $\psi = \neg a$, $a \in AP$ and $p \notin f(a)$; $\langle[p,\psi],\gamma\rangle \hookrightarrow \langle[p,\psi],\gamma\rangle \in \Delta'$,
3. if $\psi = \psi_1 \wedge \psi_2$; $\langle[p,\psi],\gamma\rangle \hookrightarrow \langle[p,\psi_1],\gamma\rangle \wedge \langle[p,\psi_2],\gamma\rangle \in \Delta'$,
4. if $\psi = \psi_1 \vee \psi_2$; $\langle[p,\psi],\gamma\rangle \hookrightarrow \langle[p,\psi_1],\gamma\rangle \vee \langle[p,\psi_2],\gamma\rangle \in \Delta'$,
5. if $\psi = EX\psi_1$; $\langle[p,\psi],\gamma\rangle \hookrightarrow \bigvee_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta}\langle[p',\psi_1],\omega\rangle \in \Delta'$,
6. if $\psi = AX\psi_1$; $\langle[p,\psi],\gamma\rangle \hookrightarrow \bigwedge_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta}\langle[p',\psi_1],\omega\rangle \in \Delta'$,
7. if $\psi = E[\psi_1U\psi_2]$; $\langle[p,\psi],\gamma\rangle \hookrightarrow \langle[p,\psi_2],\gamma\rangle \vee \bigvee_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta}(\langle[p,\psi_1],\gamma\rangle \wedge \langle[p',\psi],\omega\rangle) \in \Delta'$,
8. if $\psi = A[\psi_1U\psi_2]$; $\langle[p,\psi],\gamma\rangle \hookrightarrow \langle[p,\psi_2],\gamma\rangle \vee \bigwedge_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta}(\langle[p,\psi_1],\gamma\rangle \wedge \langle[p',\psi],\omega\rangle) \in \Delta'$,
9. if $\psi = E[\psi_1\tilde{U}\psi_2]$; $\langle[p,\psi],\gamma\rangle \hookrightarrow \langle[p,\psi_2],\gamma\rangle \wedge (\langle[p,\psi_1],\gamma\rangle \vee \bigvee_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta}\langle[p',\psi],\omega\rangle) \in \Delta'$,
10. if $\psi = A[\psi_1\tilde{U}\psi_2]$; $\langle[p,\psi],\gamma\rangle \hookrightarrow \langle[p,\psi_2],\gamma\rangle \wedge (\langle[p,\psi_1],\gamma\rangle \vee \bigwedge_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta}\langle[p',\psi],\omega\rangle) \in \Delta'$.

The ABPDS $\mathcal{BP}_\varphi$ above can be seen as the "product" of $\mathcal{P}$ with the formula φ. Intuitively, $\mathcal{BP}_\varphi$ has an accepting run from $\langle[p,\psi],\omega\rangle$ if and only if the configuration $\langle p,\omega\rangle$ satisfies ψ. Let us explain the intuition behind the different items defining Δ'.

Let $\psi = a \in AP$. If $p \in f(a)$ then for every $\omega \in \Gamma^*$, $\langle p,\omega\rangle$ satisfies ψ. Thus, $\mathcal{BP}_\varphi$ should accept $\langle[p,a],\omega\rangle$, i.e., have an accepting run from $\langle[p,a],\omega\rangle$. This is ensured by Item 1 that adds a loop in $\langle[p,a],\omega\rangle$, and the fact that $[p,a] \in F$.

Let $\psi = \neg a$, where $a \in AP$. If $p \notin f(a)$ then for every $\omega \in \Gamma^*$, $\langle p,\omega\rangle$ satisfies ψ. Thus, $\mathcal{BP}_\varphi$ should accept $\langle[p,\neg a],\omega\rangle$, i.e., have an accepting run from $\langle[p,\neg a],\omega\rangle$. This is ensured by Item 2 and the fact that $[p,\neg a] \in F$.

Item 3 expresses that if $\psi = \psi_1 \wedge \psi_2$, then for every $\omega \in \Gamma^*$, $\mathcal{BP}_\varphi$ has an accepting run from $\langle[p,\psi_1 \wedge \psi_2],\omega\rangle$ iff $\mathcal{BP}_\varphi$ has an accepting run from $\langle[p,\psi_1],\omega\rangle$ and $\langle[p,\psi_2],\omega\rangle$; meaning that $\langle p,\omega\rangle$ satisfies ψ iff $\langle p,\omega\rangle$ satisfies ψ_1 and ψ_2. Item 4 is similar to Item 3.

Item 5 means that if $\psi = EX\psi_1$, then for every $\omega \in \Gamma^*$, $\langle p,\omega\rangle$ satisfies ψ iff there exists an immediate sucessor $\langle p',\omega'\rangle$ of $\langle p,\omega\rangle$ such that $\langle p',\omega'\rangle$ satisfies ψ_1. Thus, $\mathcal{BP}_\varphi$ should have an accepting run from $\langle[p,\psi],\omega\rangle$ iff it has an accepting run from $\langle[p',\psi_1],\omega'\rangle$. Similarly, item 6 states that if $\psi = AX\psi_1$, then for every $\omega \in \Gamma^*$, $\langle p,\omega\rangle$ satisfies ψ iff $\langle p',\omega'\rangle$ satisfies ψ_1 for every immediate sucessor $\langle p',\omega'\rangle$ of $\langle p,\omega\rangle$.

Item 7 expresses that if $\psi = E[\psi_1U\psi_2]$, then for every $\omega \in \Gamma^*$, $\langle p,\omega\rangle$ satisfies ψ iff either it satisfies ψ_2, or it satisfies ψ_1 and there exists an immediate sucessor $\langle p',\omega'\rangle$ of $\langle p,\omega\rangle$ such that $\langle p',\omega'\rangle$ satisfies ψ. Item 8 is similar to Item 7.

Item 9 expresses that if $\psi = E[\psi_1\tilde{U}\psi_2]$, then for every $\omega \in \Gamma^*$, $\langle p,\omega\rangle$ satisfies ψ iff it satisfies ψ_2, and either it satisfies also ψ_1, or it has a successor that satisfies ψ. This guarantees that ψ_2 holds either always, or until both ψ_1 and ψ_2 hold. The fact that the state $[p, \psi]$ is in F ensures that paths where ψ_2 always hold are accepting. The intuition behind Item 10 is analogous.

Formally, we can show that:

Theorem 4. *Let $\mathcal{P} = (P, \Gamma, \Delta, \sharp)$ be a PDS, $f : AP \longrightarrow 2^P$ a labelling function, φ a CTL formula, and $\langle p, \omega \rangle$ a configuration of $\mathcal{P}$. Let $\mathcal{BP}_\varphi$ be the ABPDS computed above. Then, $(\mathcal{P}, \langle p, \omega \rangle) \models_{\lambda_f} \varphi$ iff $\mathcal{BP}_\varphi$ has an accepting run from the configuration $\langle [p, \varphi], \omega \rangle$.*

It follows from Theorems 3 and 4 that:

Corollary 1. *Given a PDS $\mathcal{P} = (P, \Gamma, \Delta, \sharp)$, a labeling function $f : P \longrightarrow 2^{AP}$, and a CTL formula φ, we can construct an AMA $\mathcal{A}$ in time $O(|P|^2 \cdot |\varphi|^3 \cdot (|P| \cdot |\Gamma| + |\Delta|) \cdot |\Gamma| \cdot 2^{5|P||\varphi|})$ such that for every configuration $\langle p, \omega \rangle$ of $\mathcal{P}$, $(\mathcal{P}, \langle p, \omega \rangle) \models_{\lambda_f} \varphi$ iff the AMA $\mathcal{A}$ recognizes the configuration $\langle [p, \varphi], \omega \rangle$.*

The complexity follows from the complexity of **Algorithm 1** and the fact that $\mathcal{BP}_\varphi$ has $O(|P||\varphi|)$ states and $O((|P||\Gamma| + |\Delta|)|\varphi|)$ transitions.

5 CTL Model-Checking for PushDown Systems with regular valuations

So far, we considered the "standard" model-checking problem for CTL, where the validity of an atomic proposition in a configuration c depends only on the control state of c, not on the stack. In this section, we go further and consider an extension where the set of configurations in which an atomic proposition holds is a regular set of configurations.

Let $\mathcal{P} = (P, \Gamma, \Delta, \sharp)$ be a pushdown system, c_0 its initial configuration, AP a set of atomic propositions, φ a CTL formula, and $\lambda : AP \to 2^{P \times \Gamma^*}$ a labelling function such that for every $a \in AP$, $\lambda(a)$ is a regular set of configurations. We say that λ is a regular labelling. We give in this section an algorithm that checks whether $(\mathcal{P}, c_0) \models_\lambda \varphi$. We proceed as previously: Roughly speaking, we compute an ABPDS $\mathcal{BP}'_\varphi$ such that $\mathcal{BP}'_\varphi$ recognizes a configuration c iff $(\mathcal{P}, c) \models_\lambda \varphi$. Then $(\mathcal{P}, c_0)$ satisfies φ iff c_0 is accepted by $\mathcal{BP}'_\varphi$. As previously, this can be checked using Theorem 3 and Proposition 1.

For every $a \in AP$, since $\lambda(a)$ is a regular set of configurations, let $M_a = (Q_a, \Gamma, \delta_a, I_a, F_a)$ be a multi-automaton such that $L(M_a) = \lambda(a)$, and $M_{\neg a} = (Q_{\neg a}, \Gamma, \delta_{\neg a}, I_{\neg a}, F_{\neg a})$ such that $L(M_{\neg a}) = P \times \Gamma^* \setminus \lambda(a)$ be a multi-automaton that recognizes the complement of $\lambda(a)$, i.e., the set of configurations where a does not hold. Since for every $a \in AP$ and every control state $p \in P$, p is an initial state of Q_a and $Q_{\neg a}$; to distinguish between all these initial states, for every $a \in AP$, we will denote in the following the initial state corresponding to p in Q_a (resp. in $Q_{\neg a}$) by p_a (resp. $p_{\neg a}$).

Let $\mathcal{BP}'_\varphi = (P'', \Gamma, \Delta'', F')$ be the ABPDS defined as follows[2]: $P'' = P \times cl(\varphi) \cup \bigcup_{a \in AP^+(\varphi)} Q_a \cup \bigcup_{a \in AP^-(\varphi)} Q_{\neg a}$; $F' = P \times cl_{\tilde{U}}(\varphi) \cup \bigcup_{a \in AP^+(\varphi)} F_a \cup \bigcup_{a \in AP^-(\varphi)} F_{\neg a}$; and Δ'' is the smallest set of transition rules such that for every control location $p \in P$, every subformula $\psi \in cl(\varphi)$, and every $\gamma \in \Gamma$, we have:

[2] $AP^+(\varphi)$ and $AP^-(\varphi)$ are as defined in Section 2.1.

1. if $\psi = a$, $a \in AP$; $\langle [p,\psi],\gamma\rangle \hookrightarrow \langle p_a,\gamma\rangle \in \Delta''$,
2. if $\psi = \neg a$, $a \in AP$; $\langle [p,\psi],\gamma\rangle \hookrightarrow \langle p_{\neg a},\gamma\rangle \in \Delta''$,
3. if $\psi = \psi_1 \wedge \psi_2$; $\langle [p,\psi],\gamma\rangle \hookrightarrow \langle [p,\psi_1],\gamma\rangle \wedge \langle [p,\psi_2],\gamma\rangle \in \Delta''$,
4. if $\psi = \psi_1 \vee \psi_2$; $\langle [p,\psi],\gamma\rangle \hookrightarrow \langle [p,\psi_1],\gamma\rangle \vee \langle [p,\psi_2],\gamma\rangle \in \Delta''$,
5. if $\psi = EX\psi_1$; $\langle [p,\psi],\gamma\rangle \hookrightarrow \bigvee_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta} \langle [p',\psi_1],\omega\rangle \in \Delta''$,
6. if $\psi = AX\psi_1$; $\langle [p,\psi],\gamma\rangle \hookrightarrow \bigwedge_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta} \langle [p',\psi_1],\omega\rangle \in \Delta''$,
7. if $\psi = E[\psi_1 U \psi_2]$; $\langle [p,\psi],\gamma\rangle \hookrightarrow \langle [p,\psi_2],\gamma\rangle \vee \bigvee_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta} (\langle [p,\psi_1],\gamma\rangle \wedge \langle [p',\psi],\omega\rangle) \in \Delta''$,
8. if $\psi = A[\psi_1 U \psi_2]$; $\langle [p,\psi],\gamma\rangle \hookrightarrow \langle [p,\psi_2],\gamma\rangle \vee \bigwedge_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta} (\langle [p,\psi_1],\gamma\rangle \wedge \langle [p',\psi],\omega\rangle) \in \Delta''$,
9. if $\psi = E[\psi_1 \tilde{U} \psi_2]$; $\langle [p,\psi],\gamma\rangle \hookrightarrow \langle [p,\psi_2],\gamma\rangle \wedge (\langle [p,\psi_1],\gamma\rangle \vee \bigvee_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta} \langle [p',\psi],\omega\rangle) \in \Delta''$,
10. if $\psi = A[\psi_1 \tilde{U} \psi_2]$; $\langle [p,\psi],\gamma\rangle \hookrightarrow \langle [p,\psi_2],\gamma\rangle \wedge (\langle [p,\psi_1],\gamma\rangle \vee \bigwedge_{\langle p,\gamma\rangle\hookrightarrow\langle p',\omega\rangle\in\Delta} \langle [p',\psi],\omega\rangle) \in \Delta''$.

Moreover:

11. for every transition $q_1 \xrightarrow{\gamma} q_2$ in $(\bigcup_{a\in AP^+(\varphi)} \delta_a) \cup (\bigcup_{a\in AP^-(\varphi)} \delta_{\neg a})$; $\langle q_1,\gamma\rangle \hookrightarrow \langle q_2,\epsilon\rangle \in \Delta''$,
12. for every $q \in (\bigcup_{a\in AP^+(\varphi)} F_a) \cup (\bigcup_{a\in AP^-(\varphi)} F_{\neg a})$; $\langle q,\sharp\rangle \hookrightarrow \langle q,\sharp\rangle \in \Delta''$.

The ABPDS $\mathcal{BP}'_\varphi$ has an accepting run from $\langle [p,\psi],\omega\rangle$ if and only if the configuration $\langle p,\omega\rangle$ satisfies ψ according to the regular labellings M_a's. Let us explain the intuition behind the rules above. Let $p \in P$, $\psi = a \in AP$, and $\omega \in \Gamma^*$. The ABPDS $\mathcal{BP}'_\varphi$ should accept $\langle [p,a],\omega\rangle$, iff $\langle p,\omega\rangle \in L(M_a)$. To check this, $\mathcal{BP}'_\varphi$ goes to state p_a, the initial state corresponding to p in M_a (Item 1); and then, from this state, it checks whether ω is accepted by M_a. This is ensured by Items 11 and 12. Item 11 allows $\mathcal{BP}'_\varphi$ to mimic a run of M_a on ω: if $\mathcal{BP}'_\varphi$ is in state q_1 with γ on top of its stack, and if $q_1 \xrightarrow{\gamma} q_2$ is a rule in δ_a, then $\mathcal{BP}'_\varphi$ moves to state q_2 while popping γ from the stack. Popping γ allows to check the rest of the word. The configuration is accepted if the run (with label ω) in M_a reaches a final state, i.e., if $\mathcal{BP}'_\varphi$ reaches a state $q \in F_a$ with an empty stack, i.e., a stack containing only the bottom stack symbol $\sharp$. Thus, F_a is in F''. Since all the accepting runs of $\mathcal{BP}'_\varphi$ are infinite, we add a loop on every configuration in control state $q \in F_a$ and having $\sharp$ as content of the stack (Item 12).

The intuition behind Item 2 is similar. This item applies for ψ of the from $\neg a$. Items 3–10 are similar to Items 3–10 in the construction underlying Theorem 4. We get:

Theorem 5. *$(\mathcal{P},\langle p,\omega\rangle) \models_\lambda \varphi$ iff $\mathcal{BP}'_\varphi$ has an accepting run from the configuration $\langle [p,\varphi],\omega\rangle$.*

From this theorem and Theorem 3, it follows that:

Corollary 2. *Given a PDS $\mathcal{P} = (P,\Gamma,\Delta,\sharp)$, a regular labelling function λ, and a CTL formula φ, we can construct an AMA $\mathcal{A}$ such that for every configuration $\langle p,\omega\rangle$ of $\mathcal{P}$, $(\mathcal{P},\langle p,\omega\rangle) \models_\lambda \varphi$ iff the AMA $\mathcal{A}$ recognizes the configuration $\langle [p,\varphi],\omega\rangle$. This AMA can be computed in time $O(|P|^3 \cdot |\Gamma|^2 \cdot |\varphi|^3 \cdot k^2 \cdot |\Delta| \cdot d \cdot 2^{5(|P||\varphi|+k)})$, where $k = \sum_{a\in \mathrm{AP}^+(\varphi)} |Q_a| + \sum_{a\in \mathrm{AP}^-(\varphi)} |Q_{\neg a}|$ and $d = \sum_{a\in \mathrm{AP}^+(\varphi)} |\delta_a| + \sum_{a\in \mathrm{AP}^-(\varphi)} |\delta_{\neg a}|$.*

The complexity follows from the complexity of **Algorithm 1** and the fact that $\mathcal{BP}'_\varphi$ has $O(|P||\varphi| + k)$ states and $O((|P||\Gamma| + |\Delta|)|\varphi| + d)$ transitions.

Remark 1. Note that to improve the complexity, we represent the regular valuations M_a's using AMAs instead of MAs. This prevents the exponential blow-up when complementing these automata to compute $M_{\neg a}$.

6 Experiments

We implemented all the algorithms presented in the previous sections in a tool. As far as we know, this is the first tool for CTL model-checking for PDSs. We applied our tool to the verification of sequential programs. Indeed, PDSs are well adapted to model sequential (possibly recursive) programs [EK99,ES01]. We carried out several experiments. We obtained interesting results. In particular, we were able to find bugs in linux drivers. Our results are reported in Figure 3. **Column** *formula size* gives the size of the formula. **Column** *time(s)* and *mem(kb)* give the time (in seconds) and memory (in kb). **Column** *Recu.* gives the number of iterations of $loop_1$. The last **Column** *result* gives the result whether the formula is satisfied or not (Y is satisfied, otherwise N). The first eleven lines of the table describe experiments done to evaluate **Algorithm 1.** that computes the set of configurations from which an ABPDS has an accepting run. The second part of the table describes experiments for "standard" CTL model-checking in which most of the specifications cannot be expressed in LTL. The last part considers CTL model-checking with regular valuations.

Plotter controls a plotter that creates random bar graphs [Sch02]. We checked three CTL properties for this example (**Plotter1**, **Plotter2** and **Plotter3**). **ATM** is an automatic teller machine controller. We checked that if the pincode is correct, then the ATM will provide money (**ATM1**), and otherwise, it will set an alarm (**ATM2**). **ATM3** checks that the ATM gives the money only if the pincode is correct, and if it is accessed from the main session. Regular valuations are needed to express this property. **Lock** is a lock-unlock program. We checked different properties that ensure that the program is correct. **Lock-err** is a buggy version of the program. **M-WO** is a Micro-Wave Oven controller. We checked that the oven will stop once it is hot, and that it cannot continue heating forever. **File** is a file management program. **W.G.C.** checks to solve the Wolf, Goat and Cabbage problem. **btrfsfile.c** models the source file *file.c* from the linux btrfs file system. We found a lock error in this file. **Bluetooth** is a simplified model of a Bluetooth driver [QW04]. We also found an error in this system. **w83627ehf**, **w83697ehf** and **advantech** are watchdog linux drivers. **at91rm9200** and **at32ap700x** are Real Time Clock drivers for linux. **pcf857x** corresponds also to a driver. **IEEE1394** is the IEEE 1394 driver in Linux. As described in Figure 3, we found errors in some of these drivers. We needed regular valuations to express the properties of the IEEE 1394 driver. For example, we needed to check that whenever a function *call_hpsb_send_phy_config* is invoked, there is a path where *call_hpsb_send_packet* is called before *call_hpsb_send_phy_config* returns. We need propositions about the stack to express this property. "Standard" CTL is not sufficient. **RSM** are examples written by us to check the efficiency of the regular valuations part of our tool.

7 Related Work

Alternating Büchi Pushdown Systems can be seen as non-deterministic Büchi Pushdown Systems over trees. Emptiness of non-deterministic Büchi Pushdown Systems over trees is solved in triple exponential time by Harel and Raz [HR94]. Our algorithm is less complex. [Boz07] considers the emptiness problem in Alternating Parity Pushdown Automata. The

Examples		$\lvert P\rvert+\lvert\Gamma\rvert$ $+\lvert\Delta\rvert$	Formula size	Recu	Time(s)	Mem(kb)	Result
Algorithm 1	1	3+3+4	-	3	0	22.34	Y
	2	17+5+24	-	4	0	33.23	N
	3	73+5+73	-	4	0.02	128.28	Y
	4	75+6+75	-	5	0.02	81.36	N
	5	3+4+4	-	4	0	22.36	N
	6	3+4+5	-	3	0	21.54	Y
	7	3+4+4	-	3	0	20.11	Y
	8	3+4+4	-	4	0	27.40	Y
	9	74+6+76	-	5	0.02	87.54	Y
	10	17+5+24	-	3	0	28.46	Y
	11	18+5+28	-	3	0	26.15	Y
Standard	Plotter.1	1+19+24	2	3	0.02	41.56	Y
	Plotter.2	1+19+24	2	3	0	43.52	N
	Plotter.3	1+19+24	14	9	0.03	241.32	Y
	ATM.1	2+18+45	8	6	0.03	169.64	Y
	ATM.2	2+18+45	10	6	0.03	192.53	Y
	Lock.1	6+37+82	7	11	0.11	387.15	Y
	Lock.2	6+37+82	7	11	0.11	379.46	N
	Lock-err	6+37+82	3	9	0.00	186.52	N
	M-WO.1	1+7+12	6	2	0	40.20	Y
	M-WO.2	1+7+12	6	7	0	37.28	N
	File.1	1+5+9	2	3	0	34.77	Y
	File.2	1+5+9	2	4	0.02	32.51	N
	W.G.C.	16+1+40	23	2	0.05	202.01	Y
	btrfs/file.c	2+14+20	3	10	0	64.32	N
	btrfs/file.c-fixed	2+15+22	3	9	0.02	82.52	Y
	bluetooth	32+12+294	5	8	0.12	821.03	N
	w83627ehf	1+20+20	5	9	0.02	132.76	N
	w83627ehf-fixed	1+21+22	5	4	0.03	121.69	Y
	w83697ehf	1+56+57	6	11	0.35	394.61	Y
	advantech	2+16+31	7	6	0.05	120.41	Y
	at91rm9200	4+15+64	7	5	0.06	234.42	N
	at91rm9200-fixed	4+16+67	7	6	0.12	255.62	Y
	at32ap700x	4+25+105	7	8	0.15	356.04	N
	at32ap700x-fixed	4+25+109	7	9	0.22	334.42	Y
	pcf857x	1+98+106	10	18	0.23	541.35	Y
Regular Valuation	ATM.3	2+18+45	8	6	0.20	352.47	Y
	File.3	1+5+9	5	5	0	33.21	Y
	RSM1	1+8+11	25	4	0.06	438.23	Y
	RSM2	1+8+12	30	4	0.48	1231.45	Y
	RSM3	1+11+17	45	4	12.11	6206.73	Y
	RSM4	1+11+18	45	4	0.72	1269.26	Y
	RSM5	1+11+16	35	4	12.14	6212.2	Y
	ieee1394_core_1	1+104+108	12	14	0.20	413.69	Y
	ieee1394_core_2	1+104+108	13	14	0.19	422.17	Y
	ieee1394_core_3	1+104+108	14	17	0.19	438.42	N
	ieee1394_core_4	1+104+109	14	14	0.19	414.27	Y

Fig. 3. The performance of our tool.

emptiness problem of nondeterministic parity pushdown tree automata is investigated in [KPV02,BMP05,BMP10]. ABPDSs can be seen as a subclass of these Automata. For ABPDSs, our algorithm is more general than the ones in these works since it allows to characterize and compute the set of configurations from which the ABPDS has an accepting run, whereas the other algorithms allow only to check emptiness

Model-checking pushdown systems against branching time temporal logics has already been intensively investigated in the literature. Several algorithms have been proposed. Walukiewicz [Wal00] showed that CTL model checking is EXPTIME-complete for PDSs. The complexity of our algorithm matches this bound. CTL corresponds to a fragment of the alternation-free μ-calculus and of CTL*. Model checking full μ-calculus for PDSs has been considered in [BS95,BS97,Wal96,KV00]. These algorithms allow only to determine whether a given configuration satisfies the property. They cannot compute the set of all the configurations where the formula holds. As far as CTL is concerned, our algorithm is more general since it allows to compute a finite automaton that characterizes the set of all such configurations. Moreover, the complexity of our algorithm is comparable to the ones of [BS95,BS97,Wal96,KV00] when applied to CTL, it is even better in some cases.

[PV04,KPV10] considers the global model-checking μ-calculus problem for PDSs, i.e., they compute the set of configurations that satisfy the formula. They reduce this problem to the membership problem in two-way alternating parity tree automata. [KPV10] considers also μ-calculus model-checking with regular valuations. These algorithms are more complex, technically more complicated and less intuitive than our procedure. Indeed, the complexity of [PV04,KPV10] is $(|\varphi| \cdot |P| \cdot |\Delta| \cdot |\Gamma|)^{O(|P|\cdot|\Delta|\cdot|\varphi|)^2}$, whereas our complexity is $O(|P|^2 \cdot |\varphi|^3 \cdot (|P| \cdot |\Gamma| + |\Delta|) \cdot |\Gamma| \cdot 2^{5|P||\varphi|})$.

In [BEM97], Bouajjani et al. consider alternating pushdown systems (without the Büchi accepting condition). They provide an algorithm to compute a finite automaton representing the Pre^* of a regular set of configurations for these systems. We use this procedure in $loop_2$ of **Algorithm 1**. [SSE06] showed how to efficiently implement this procedure. We used the ideas in [SSE06] while implementing **Algorithm 1**. In their paper, Bouajjani et al. applied their Pre^* algorithm to compute the set of PDS configurations that satisfy a given alternation-free μ-calculus formula. Their procedure is more complex than ours. It is exponential in $|P| \cdot |\varphi|^2$ whereas our algorithm is exponential only in $|P| \cdot |\varphi|$, where $|P|$ is the number of states of the PDS and $|\varphi|$ is the size of the formula.

It is well known that the model-checking problem for μ-calculus is polynomially reducible to the problem of solving parity games. Parity games for pushdown systems are considered in [Cac02b,Ser03] and are solved in time exponential in $(|P||\varphi|)^2$. As far as CTL model-checking is concerned, our method is simpler, less complex, and more intuitive than these algorithms.

Model checking CTL* for PDS is 2EXPTIME-complete (in the size of the formula) [Boz07]. Algorithms for model-checking CTL* specifications for PDSs have been proposed in [FWW97,EKS03,EKS01,Boz07]. [FWW97] considers also CTL* model checking with regular valuations. When applied to CTL formulas, these algorithms are more complex than our techniques. They are double exponential in the size of the formula and exponential in the size of the system; whereas our procedure is only exponential for both sizes (the formula and the system).

LTL model-checking with regular valuations was considered in [EKS03,EKS01]. Their algorithm is based on a reduction to the "standard" LTL model-checking problem for PDSs. The reduction is done by performing a kind of product of the PDS with the different regular automata representing the different constraints on the stack. Compared to these algorithms, our techniques for CTL model-checking with regular valuations are direct, in the sense that they do not necessitate to make the product of the PDS with the different automata of the regular constraints.

References

[BEM97] A. Bouajjani, J. Esparza, and O. Maler. Reachability Analysis of Pushdown Automata: Application to Model Checking. In *CONCUR'97*. LNCS 1243, 1997.

[BMP05] Laura Bozzelli, Aniello Murano, and Adriano Peron. Pushdown module checking. In *LPAR*, pages 504–518, 2005.

[BMP10] Laura Bozzelli, Aniello Murano, and Adriano Peron. Pushdown module checking. *Formal Methods in System Design*, 36(1):65–95, 2010.

[Boz07] Laura Bozzelli. Complexity results on branching-time pushdown model checking. *Theor. Comput. Sci.*, 379(1-2):286–297, 2007.

[BS95] Olaf Burkart and Bernhard Steffen. Composition, decomposition and model checking of pushdown processes. *Nord. J. Comput.*, 2(2):89–125, 1995.

[BS97] Olaf Burkart and Bernhard Steffen. Model checking the full modal mu-calculus for infinite sequential processes. In *ICALP*, pages 419–429, 1997.

[Cac02a] Thierry Cachat. Symbolic strategy synthesis for games on pushdown graphs. In *ICALP*, pages 704–715, 2002.

[Cac02b] Thierry Cachat. Uniform solution of parity games on prefix-recognizable graphs. *Electr. Notes Theor. Comput. Sci.*, 68(6), 2002.

[EHRS00] J. Esparza, D. Hansel, P. Rossmanith, and S. Schwoon. Efficient algorithm for model checking pushdown systems. In *CAV'00*, volume 1885 of *LNCS*, 2000.

[EK99] J. Esparza and J. Knoop. An automata-theoretic approach to interprocedural data-flow analysis. In *FOSSACS'99*. LNCS 1578, 1999.

[EKS01] Javier Esparza, Antonín Kucera, and Stefan Schwoon. Model-checking ltl with regular valuations for pushdown systems. In *TACS*, pages 316–339, 2001.

[EKS03] Javier Esparza, Antonín Kucera, and Stefan Schwoon. Model checking ltl with regular valuations for pushdown systems. *Inf. Comput.*, 186(2):355–376, 2003.

[ES01] J. Esparza and S. Schwoon. A bdd-based model checker for recursive programs. In *In Proc. of CAV'01*, 2001.

[FWW97] A. Finkel, B. Willems, and P. Wolper. A Direct Symbolic Approach to Model Checking Pushdown Systems. In *Infinity'97, ENTCS 9*. Elsevier Sci. Pub., 1997.

[HR94] David Harel and Danny Raz. Deciding emptiness for stack automata on infinite trees. *Inf. Comput.*, 113(2):278–299, 1994.

[KPV02] Orna Kupferman, Nir Piterman, and Moshe Y. Vardi. Pushdown specifications. In *LPAR*, pages 262–277, 2002.

[KPV10] Orna Kupferman, Nir Piterman, and Moshe Y. Vardi. An automata-theoretic approach to infinite-state systems. In *Essays in Memory of Amir Pnueli*, pages 202–259, 2010.

[KV00] Orna Kupferman and Moshe Y. Vardi. An automata-theoretic approach to reasoning about infinite-state systems. In *CAV*, pages 36–52, 2000.

[PV04] Nir Piterman and Moshe Y. Vardi. Global model-checking of infinite-state systems. In *CAV*, pages 387–400, 2004.

[QW04] S. Qadeer and D. Wu. Kiss: Keep it simple and sequential. In *PLDI 04: Programming Language Design and Implementation*, pages 14–24, 2004.

[Sch02] Stefan Schwoon. *Model-Checking Pushdown Systems*. PhD thesis, Technische Universität München, 2002.

[Ser03] Olivier Serre. Note on winning positions on pushdown games with [omega]-regular conditions. *Inf. Process. Lett.*, 85(6):285–291, 2003.

[SSE06] Dejvuth Suwimonteerabuth, Stefan Schwoon, and Javier Esparza. Efficient algorithms for alternating pushdown systems with an application to the computation of certificate chains. In *ATVA*, pages 141–153, 2006.

[Wal96] Igor Walukiewicz. Pushdown processes: Games and model checking. In *CAV*, pages 62–74, 1996.

[Wal00] Igor Walukiewicz. Model checking ctl properties of pushdown systems. In *FSTTCS*, pages 127–138, 2000.

A Appendix

A.1 Proof of Lemma 1

Lemma 1. *$\mathcal{BP}$ has a run ρ from a configuration $\langle p, \omega\rangle$ such that each path of ρ visits configurations with control locations in F at least k times iff $\langle p, \omega\rangle \in X_k$.*

Proof: ($\Longrightarrow$) The proof proceeds by induction on k. We directly obtain that $\langle p, \omega\rangle \in X_0 = P \times \Gamma^*$. We only need to show that $\langle p, \omega\rangle \in X_k$ when $k \geq 1$.

Let $\langle p_1, \omega_1\rangle, .., \langle p_n, \omega_n\rangle$ be the first nodes of ρ that are visited in each path of ρ such that $p_i \in F$. Then we get: (a) $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p_1, \omega_1\rangle, .., \langle p_n, \omega_n\rangle\}$, (b) for every $1 \leq i \leq n$, $p_i \in F$, (c) for every $1 \leq i \leq n$, $\mathcal{BP}$ has a run ρ_i from the configuration $\langle p_i, \omega_i\rangle$ such that all the paths of ρ_i can visit some configurations with control locations in F at least $k-1$ times.

By applying the induction hypothesis to (c): we obtain that $\langle p_i, \omega_i\rangle \in X_{k-1}$ for every $1 \leq i \leq n$. Since $p_i \in F$ for every $1 \leq i \leq n$, we have $\langle p_i, \omega_i\rangle \in X_{k-1} \cap F \times \Gamma^*$ for every $1 \leq i \leq n$. Since $X_k = Pre^+(X_{k-1} \cap F \times \Gamma^*)$ and from (a), we get that $\langle p, \omega\rangle \in X_k$.

($\Longleftarrow$) Let's apply the induction on k. It is straightforward when $k = 0$. We only need to show that $\mathcal{BP}$ has a run ρ from the configuration $\langle p, \omega\rangle$ such that each path of ρ can visit some configurations with control locations in F at least k times when $k \geq 1$.

Since $\langle p, \omega\rangle \in X_k$ where $X_k = Pre^+(X_{k-1} \cap F \times \Gamma^*)$, we obtain that $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p_1, \omega_1\rangle, .., \langle p_n, \omega_n\rangle\}$, and $\langle p_i, \omega_i\rangle \in X_{k-1} \cap F \times \Gamma^*$ for each $1 \leq i \leq n$.

By applying the induction hypothesis to $\langle p_i, \omega_i\rangle \in X_{k-1}$ for each $1 \leq i \leq n$, we get that $\mathcal{BP}$ has a run ρ_i from the configuration $\langle p_i, \omega_i\rangle$ such that each path of ρ_i can visit some configurations with control locations in F at least $k-1$ times. Thus $\mathcal{BP}$ has a run ρ from the configuration $\langle p, \omega\rangle$ such that each path of ρ can visit some configurations with control locations in F at least k times. □

A.2 Proof of Theorem 1

Theorem 1. *$\mathcal{BP}$ has an accepting run from a configuration $\langle p, \omega\rangle$ iff $\langle p, \omega\rangle \in Y_{\mathcal{BP}}$.*

Proof: ($\Longrightarrow$) First we show that if $\mathcal{BP}$ has an accepting run from the configuration $\langle p, \omega \rangle$, then the configuration $\langle p, \omega \rangle$ must be in $Y_{\mathcal{BP}}$. We prove that if the configuration $\langle p, \omega \rangle$ is not in $Y_{\mathcal{BP}}$, then $\mathcal{BP}$ has not any accepting run from $\langle p, \omega \rangle$.

Since $\langle p, \omega \rangle \notin Y_{\mathcal{BP}}$ and $Y_{\mathcal{BP}} = \bigcap_{i \geq 0} X_i$, there exists $k \geq 0$ such that $\langle p, \omega \rangle \notin X_k$. By **Lemma** 1, all the runs from the configuration $\langle p, \omega \rangle$ can visit configurations with control locations in F at most $k-1$ times, otherwise $\langle p, \omega \rangle \in X_k$, which contradicts the fact that $\langle p, \omega \rangle \notin X_k$. Thus, $\mathcal{BP}$ has not any accepting run from the configuration $\langle p, \omega \rangle$.

($\Longleftarrow$) Now we prove the other direction, i.e., we prove that if the configuration $\langle p, \omega \rangle$ is in $Y_{\mathcal{BP}}$, then $\mathcal{BP}$ has an accepting run from $\langle p, \omega \rangle$.

Since $Y_{\mathcal{BP}}$ is the greatest fixpoint of the function $f(X) = Pre^{+}(X \cap F \times \Gamma^{*})$, we get that $Y_{\mathcal{BP}} = Pre^{+}(Y_{\mathcal{BP}} \cap F \times \Gamma^{*})$.

Since $\langle p, \omega \rangle \in Y_{\mathcal{BP}}$, we get $\langle p, \omega \rangle \in Pre^{+}(Y_{\mathcal{BP}} \cap F \times \Gamma^{*})$. By the definition of Pre^{+}, there exists a set of configurations $\{\langle p_1, \omega_1 \rangle, ..., \langle p_n, \omega_n \rangle\} \subseteq Y_{\mathcal{BP}} \cap F \times \Gamma^{*}$ such that $\langle p, \omega \rangle \Longrightarrow_{\mathcal{BP}} \{\langle p_1, \omega_1 \rangle, ..., \langle p_n, \omega_n \rangle\}$.

Since $\{\langle p_1, \omega_1 \rangle, ..., \langle p_n, \omega_n \rangle\} \subseteq Y_{\mathcal{BP}} \cap F \times \Gamma^{*}$, we obtain that $\langle p_i, \omega_i \rangle \in Y_{\mathcal{BP}}$ and $p_i \in F$ for every $1 \leq i \leq n$. Let's construct a finite tree ρ with root $\langle p, \omega \rangle$, the leaves of ρ are $\langle p_1, \omega_1 \rangle, ..., \langle p_n, \omega_n \rangle$, the inner nodes of ρ are the successors during the run $\langle p, \omega \rangle \Longrightarrow_{\mathcal{BP}} \{\langle p_1, \omega_1 \rangle, ..., \langle p_n, \omega_n \rangle\}$. Each path of ρ can visit some configurations with control locations in F at least once.

Since $\langle p_i, \omega_i \rangle \in Y_{\mathcal{BP}}$ for every $1 \leq i \leq n$, we can repeatedly construct a finite tree ρ_i for the configuration $\langle p_i, \omega_i \rangle$ such that ρ_i has the same properties as ρ. Let's replace each leaf $\langle p_i, \omega_i \rangle$ in ρ by the tree ρ_i and obtain a new tree ρ such that each path of the new tree ρ can visit some configurations with control locations in F at least twice.

Now we infinitely repeat this procedure to the leaves of the latest tree ρ. Finally, each path of the latest tree ρ can infinitely often visit some configurations with control locations in F. We obtain that ρ is an accepting run.

□

A.3 Proof of Proposition 2

Proposition 2. *In **Algorithm** 1, for every $\gamma \in \Gamma$, $p \in P, S \subseteq Q$; at each step $i \geq 2$, if $p^i \xrightarrow{\gamma} S \in \delta$, then $p^{i-1} \xrightarrow{\gamma} \pi^{-1}(\pi^i(S)) \in \delta$.*

Proof: The proposition states that $p^i \xrightarrow{\gamma} S \in \delta \Longrightarrow p^{i-1} \xrightarrow{\gamma} \pi^{-1}(\pi^i(S)) \in \delta$ in the **Algorithm** 1. Note that the transition rule is added by the saturation procedure (lines 5−9) and may be replaced by the substitution (line 11). Both situations should be considered. We proceed by induction on i.

Basis. $i = 2$. Let n be the number of transition rules that were added by the saturation procedure. We will prove that $p^2 \xrightarrow{\gamma} S \in \delta \Longrightarrow p^1 \xrightarrow{\gamma} \pi^{-1}(\pi^2(S)) \in \delta$ by induction on n.

- **Basis**. $n = 0$. Then there is only $p^2 \xrightarrow{\epsilon} p^1 \in \delta$ for $p \in F$. So there is no $p^2 \xrightarrow{\gamma} S$ for every $p \in P, \gamma \in \Gamma, S \subseteq P \times \{1, 2\} \cup \{q_f\}$.

– **Step**. $n \geq 1$. Let $t = p^2 \xrightarrow{\gamma} S$ be the n-th transition rule added by the saturation procedure. Then there exists a transition rule

$$\langle p, \gamma \rangle \longrightarrow \{\langle p_1, \omega_1 \rangle, ..., \langle p_m, \omega_m \rangle\} \in \Delta \quad (1)$$

such that $p_j^2 \xrightarrow{\omega_j}_\delta S_j$ for every $1 \leq j \leq m$ and $S = \bigcup_{j=1}^m S_j$.

Because of (1), it is sufficient to prove that $p_j^1 \xrightarrow{\omega_j}_\delta \pi^{-1}(\pi^2(S_j))$ existed in the saturation procedure of the first iteration of $loop_1$ for every $1 \leq j \leq m$ by applying induction on $|\omega_j|$ the length of ω_j.

- **Basis**. $|\omega_j| = 0$. Then $\omega_j = \epsilon$. Since $p_j^2 \xrightarrow{\epsilon}_\delta S_j$ and there are transition rules $p_j^2 \xrightarrow{\epsilon}_\delta p_j^2$ and $p_j^2 \xrightarrow{\epsilon}_\delta p_j^1$ for $p_j \in F$. We get $S_j = \{p_j^2\}$ or $S_j = \{p_j^1\}$. Since $\pi^{-1}(\pi^2(\{p_j^2\})) = p_j^1, \pi^{-1}(\pi^2(\{p_j^1\})) = p_j^1$ and there existed $p_j^1 \xrightarrow{\epsilon}_\delta p_j^1$ in the saturation procedure of the first iteration of $loop_1$, we obtain that $p_j^1 \xrightarrow{\epsilon}_\delta \pi^{-1}(\pi^2(S_j))$ existed in the saturation procedure of the first iteration of $loop_1$.

- **Step**. $|\omega_j| \geq 1$. We prove that $p_j^1 \xrightarrow{\omega_j}_\delta \pi^{-1}(\pi^2(S_j))$ existed in the saturation procedure of the first iteration of $loop_1$ depending on the case whether the first step of $p_j^2 \xrightarrow{\omega_j}_\delta S_j$ is the ϵ-transition $p_j^2 \xrightarrow{\epsilon} p_j^1$ or not.
  - The first step is $p_j^2 \xrightarrow{\epsilon} p_j^1$. Then we have $p_j^2 \xrightarrow{\epsilon} p_j^1 \xrightarrow{\omega_j}_\delta S_j$. By the definition of project function π^i, the substitution procedure will not change any transition rules in the first iteration of $loop_1$, we obtain that $p_j^1 \xrightarrow{\omega_j}_\delta S_j$ already existed in the saturation procedure of first iteration of $loop_1$. Since the successor of the state $q^1 \in P \times \{1\}$ is a subset of $P \times \{1\} \cup \{q_f\}$, we have $\pi^{-1}(\pi^2(S_j)) = S_j$. Hence $p_j^1 \xrightarrow{\omega_j}_\delta \pi^{-1}(\pi^2(S_j))$.

  - The first step is not $p_j^2 \xrightarrow{\epsilon} p_j^1$. Then there exists $\gamma' \in \Gamma, u \in \Gamma^*$ and $R \in P \times \{2, 1\} \cup \{q_f\}$ such that $\omega_j = \gamma' u$ and $p_j^2 \xrightarrow{\gamma'} R \xrightarrow{u}_\delta S_j$.

    Since $p_j^2 \xrightarrow{\gamma'} R$ already existed before adding the n-th transition rule, by applying the induction hypothesis (induction on n), we get that $p_j^1 \xrightarrow{\gamma'} \pi^{-1}(\pi^2(R))$ existed in the saturation procedure of the first iteration of $loop_1$.

    Let $R = \{q_1^2, ..., q_{g_1}^2, q_{g_1+1}^1, ..., q_{g_2}^1\}$, since $R \xrightarrow{u}_\delta S_j$, we obtain that $q_k^2 \xrightarrow{u}_\delta R_k$ existed in the saturation procedure of the second iteration of $loop_1$ for every $1 \leq k \leq g_1$, and $q_k^1 \xrightarrow{u}_\delta R_k$ existed in the saturation procedure of the second iteration of $loop_1$ for every $g_1 + 1 \leq k \leq g_2$ and $S_j = \bigcup_{k=1}^{g_2} R_k$.

    Since $|u| < |\omega_j|$, by applying the induction hypothesis (induction on $|\omega_j|$) to $q_k^2 \xrightarrow{u}_\delta R_k$ for every $1 \leq k \leq g_1$, we get that $q_k^1 \xrightarrow{u}_\delta \pi^{-1}(\pi^2(R_k))$ existed in the saturation procedure of the second iteration of $loop_1$. Then we obtain that $\{q_1^1, ..., q_{g_1}^1, q_{g_1+1}^1, ..., q_{g_2}^1\} \xrightarrow{u}_\delta \bigcup_{k=1}^{g_1} \pi^{-1}(\pi^2(R_k)) \cup \bigcup_{k=g_1+1}^{g_2} R_k$ existed in the saturation procedure of the first iteration of $loop_1$.

Since $\pi^{-1}(\pi^2(S_j)) = \pi^{-1}(\pi^2(\bigcup_{k=1}^{g_1} R_k \cup \bigcup_{k=g_1+1}^{g_2} R_k)) = \bigcup_{k=1}^{g_1} \pi^{-1}(\pi^2(R_k)) \cup \bigcup_{k=g_1+1}^{g_2} R_k$, we obtain that $p_j^1 \xrightarrow{\gamma'} \pi^{-1}(\pi^2(R)) \xrightarrow{u}_{\delta} \pi^{-1}(\pi^2(S_j))$ existed in the saturation procedure of the second iteration of $loop_1$. Since $\omega_j = \gamma' u$ and line 11 will not change any transition rules at the end of the first iteration, we obtain that $p_j^1 \xrightarrow{\omega_j}_{\delta} \pi^{-1}(\pi^2(S_j))$ existed in the saturation procedure of the first iteration of $loop_1$.

Step. $i \geq 3$. Let n be the number of transition rules added by the saturation procedure. We will proceed by applying induction on n.

- **Basis**. $n = 0$. Then there is only $p^i \xrightarrow{\epsilon} p^{i-1}$ for $p \in F$. So there is no $p^i \xrightarrow{\gamma} S$ for every $p \in P, \gamma \in \Gamma, S \subseteq P \times \{i, i-1\} \cup \{q_f\}$. Note that each state of $P \times \{i-2\}$ and transition rules from the states $P \times \{i-2\}$ can be erased as soon as the $(i-1)$-th iteration of $loop_1$ finished.
- **Step**. $n \geq 1$. Let $t = p^i \xrightarrow{\gamma} S$ be the n-th transition added by the saturation procedure. Then there exists a transition rule
$$\langle p, \gamma \rangle \longrightarrow \{\langle p_1, \omega_1 \rangle, ..., \langle p_m, \omega_m \rangle\} \text{ in } \Delta \tag{2}$$
such that $p_j^i \xrightarrow{\omega_j}_{\delta} S_j$ for every $1 \leq j \leq m$ and $S = \bigcup_{j=1}^{m} S_j$.

  Because of (2), it is sufficient to prove that there exists $R_j \subseteq P \times \{i-1, i-2\} \cup \{q_f\}$ such that $p_j^{i-1} \xrightarrow{\omega_j}_{\delta} R_j$ existed in the saturation procedure of the $(i-1)$-th iteration of $loop_1$ and $\pi^{i-1}(R_j) = \pi^{-1}(\pi^i(S_j))$ by applying induction on $|\omega_j|$ the length of ω_j.

  - **Basis**. $|\omega_j| = 0$. Then $\omega_j = \epsilon$. Since $p_j^i \xrightarrow{\epsilon}_{\delta} S_j$ and there are transition rules $p_j^i \xrightarrow{\epsilon}_{\delta} p_j^i$ and $p_j^i \xrightarrow{\epsilon} p_j^{i-1}$ for $p \in F$. We get $S_j = \{p_j^i\}$ or $S_j = \{p_j^{i-1}\}$. Since $\pi^{-1}(\pi^i(\{p_j^i\})) = \{p_j^{i-1}\}$, $\pi^{-1}(\pi^i(\{p_j^{i-1}\})) = p_j^{i-1}$ and there existed $p_j^{i-1} \xrightarrow{\epsilon}_{\delta} p_j^{i-1}$ in the saturation procedure of the $(i-1)$-th iteration of $loop_1$, we obtain that $p_j^{i-1} \xrightarrow{\epsilon}_{\delta} \pi^{-1}(\pi^i(S_j))$ existed in the saturation procedure of the $(i-1)$-th iteration of $loop_1$.
  - **Step**. $|\omega_j| \geq 1$. We will prove that there exists $R_j \subseteq P \times \{i-1, i-2\} \cup \{q_f\}$ such that $p_j^{i-1} \xrightarrow{\omega_j}_{\delta} R_j$ existed in the saturation procedure at the $(i-1)$-th iteration of $loop_1$ and $\pi^{i-1}(R_j) = \pi^{-1}(\pi^i(S_j))$ depending on the case whether the first step of $p_j^i \xrightarrow{\omega_j}_{\delta} S_j$ is the ϵ-transition $p_j^i \xrightarrow{\epsilon} p_j^{i-1}$ or not.
    * The first step is $p_j^i \xrightarrow{\epsilon} p_j^{i-1}$. Then we have $p_j^i \xrightarrow{\epsilon} p_j^{i-1} \xrightarrow{\omega_j}_{\delta} S_j$ in the i-th iteration of $loop_1$ and $p_j \in F$. Since the $(i-1)$-th iteration of $loop_1$ has finished, by line 11, we obtain $S_j \subseteq P \times \{i-1\} \cup \{q_f\}$. First we will show that there exists $p_j^{i-2} \xrightarrow{\omega_j}_{\delta} \pi^{-1}(S_j)$ in the $(i-1)$-th iteration of $loop_1$ by induction

on $|\omega_j|$ the length of ω_j.

- **Basis**. $|\omega_j| = 1$. Then $\omega_j \in \Gamma$. Since $p_j^{i-1} \xrightarrow{\omega_j}_\delta S_j$ existed in the i-th iteration of $loop_1$, we get that there exists $P_2 \subseteq P \times \{i-1, i-2\} \cup \{q_f\}$ such that $p_j^{i-1} \xrightarrow{\omega_j} P_2$ is added by the saturation procedure of the $(i-1)$-th iteration of $loop_1$ and $P_2^{i-1} = S_j$, by applying the induction hypothesis (induction on i), we get that $p_j^{i-2} \xrightarrow{\omega_j}_\delta \pi^{-1}(\pi^{i-1}(P_2))$ existed in the $(i-1)$-th iteration of $loop_1$.

- **Step**. $|\omega_j| \geq 2$. Since $S_j \subseteq P \times \{i-1\} \cup \{q_f\}$, there exists $\gamma' \in \Gamma, u \in \Gamma^*, R \subseteq P \times \{i-1\} \cup \{q_f\}$ such that $\omega_j = \gamma' u$ and $p_j^{i-1} \xrightarrow{\gamma'} R \xrightarrow{u}_\delta S_j$ exists in the i-th iteration of $loop_1$.

  Since $p_j^{i-1} \xrightarrow{\gamma'} R$ existed in the $(i-1)$-th iteration of $loop_1$, there exists $R_1 \subseteq P \times \{i-1, i-2\}$ such that $R = \pi^{i-1}(R_1)$ and $p_j^{i-1} \xrightarrow{\gamma'} R_1$ is added by the saturation procedure of the $(i-1)$-th iteration of $loop_1$. By applying the induction hypothesis (induction on i) to $p_j^{i-1} \xrightarrow{\gamma'} R_1$, we get that $p_j^{i-2} \xrightarrow{\gamma'} \pi^{-1}(R)$ existed in the $(i-1)$-th iteration of $loop_1$.

  Since $|u| < |\omega_j|$ and $R \subseteq P \times \{i-1\} \cup \{q_f\}$, by applying the induction hypothesis (induction on $|\omega_j|$) to each transition rule whose left-hand side is an element of R and the input is u, we get that $\pi^{-1}(R) \xrightarrow{u}_\delta \pi^{-1}(S_j)$ existed in the $(i-1)$-th iteration of $loop_1$. Hence we obtain that $p_j^{i-2} \xrightarrow{\gamma'} \pi^{-1}(R) \xrightarrow{u}_\delta \pi^{-1}(S_j)$ existed in the $(i-1)$-th iteration of $loop_1$. Since $\omega_j = \gamma' u$, we obtain that $p_j^{i-2} \xrightarrow{\omega_j}_\delta \pi^{-1}(S_j)$ existed in the $(i-1)$-th iteration of $loop_1$.

Since $p_j \in F$ and line 4, there existed $p_j^{i-1} \xrightarrow{\epsilon} p_j^{i-2} \xrightarrow{\omega_j}_\delta R_j$ in the saturation procedure of the $(i-1)$-th iteration of $loop_1$, where $R_j = \pi^{-1}(S_j)$.

* The first step is not $p_j^i \xrightarrow{\epsilon} p_j^{i-1}$. Then there exists $\gamma' \in \Gamma, u \in \Gamma^*, R = \{g_1^i, ..., g_{n_1}^i, g_{n_1+1}^{i-1}, ..., g_{n_2}^{i-1}\} \subseteq P \times \{i, i-1\} \cup \{q_f\}$ such that
  - $\omega_j = \gamma' u$,
  - $p_j^i \xrightarrow{\gamma'} \{g_1^i, ..., g_{n_1}^i, g_{n_1+1}^{i-1}, ..., g_{n_2}^{i-1}\}$ existed in the i-th iteration of $loop_1$,
  - $g_k^i \xrightarrow{u}_\delta P_k$ existed in the i-th iteration of $loop_1$ for every $1 \leq k \leq n_1$,
  - $g_k^{i-1} \xrightarrow{u}_\delta P_k$ existed in the i-th iteration of $loop_1$ for every $n_1 + 1 \leq k \leq n_2$,
  - $S_j = \bigcup_{k=1}^{n_2} P_k$.

Since $p_j^i \xrightarrow{\gamma'} R$ already existed before the n-th transition rule added, by applying the induction hypothesis (induction on n) to $p_j^i \xrightarrow{\gamma'} R$, we get that $p_j^{i-1} \xrightarrow{\gamma'} \{g_1^{i-1}, ..., g_{n_2}^{i-1}\}$ existed in the i-th iteration of $loop_1$. Thus there exists $R_1 \subseteq P \times \{i-1, i-2\} \cup \{q_f\}$ such that $\pi^{i-1}(R_1) = \{g_1^{i-1}, ..., g_{n_2}^{i-1}\}$ and $p_j^{i-1} \xrightarrow{\gamma'} R_1$ was added by the saturation procedure of the $(i-1)$-th iteration of $loop_1$.

Since $g_k^i \xrightarrow{u}_\delta P_k$ existed in the i-th iteration of $loop_1$ for every $1 \leq k \leq n_1$ and $|u| < |\omega_j|$, by applying the induction hypothesis (induction on $|\omega_j|$), we get that $g_k^{i-1} \xrightarrow{u}_\delta \pi^{-1}(\pi^i(P_k))$ existed in the i-th iteration of $loop_1$ for every $1 \leq k \leq n_1$. This implies that there exists $Q_k \subseteq P \times \{i-1, i-2\} \cup \{q_f\}$ such that $g_k^{i-1} \xrightarrow{u}_\delta Q_k$ was added by the saturation procedure of the $(i-1)$-th iteration of $loop_1$ and $\pi^{i-1}(Q_k) = \pi^{-1}(\pi^i(P_k))$ for every $1 \leq k \leq n_1$.

Since $g_k^{i-1} \xrightarrow{u}_\delta P_k$ existed in the i-th iteration of $loop_1$ for every $n_1 + 1 \leq k \leq n_2$ and $g_k^{i-1} \xrightarrow{u}_\delta P_k$ is obtained from the $(i-1)$-th iteration of $loop_1$, there exists $Q_k \subseteq P \times \{i-1, i-2\} \cup \{q_f\}$ such that $g_k^{i-1} \xrightarrow{u}_\delta Q_k$ was added by the saturation procedure of the $(i-1)$-th iteration of $loop_1$ and $\pi^{i-1}(Q_k) = \pi^{-1}(\pi^i(P_k))$ for every $n_1 + 1 \leq k \leq n_2$. By applying the induction on the length of u and by the induction hypothesis (induction on i), we can get that $g_k^{i-2} \xrightarrow{u}_\delta \pi^{-1}(\pi^{i-1}(Q_k))$ existed in the $(i-1)$-th iteration of $loop_1$ for every $n_1 + 1 \leq k \leq n_2$.

Putting $p_j^{i-1} \xrightarrow{\gamma'} R_1$, $g_k^{i-1} \xrightarrow{u}_\delta Q_k$ and $g_k^{i-2} \xrightarrow{u}_\delta \pi^{-1}(\pi^{i-1}(Q_k))$ for every $1 \leq k \leq n_1$, and $g_k^{i-2} \xrightarrow{u}_\delta \pi^{-1}(\pi^{i-1}(Q_k))$ for every $n_1 + 1 \leq k \leq n_2$ together, we obtain $p_j^{i-1} \xrightarrow{\gamma'} R_1 \xrightarrow{u}_\delta$ existed in the saturation procedure of the $(i-1)$-th iteration of $loop_1$, where R_j is computed depending on the states of R_1.

Since $\omega_j = \gamma' u$, we obtain that $p_j^{i-1} \xrightarrow{\omega}_\delta R_j$ existed in the saturation procedure of the $(i-1)$-th iteration of $loop_1$.

By applying the saturation procedure to (2), we get $p^{i-1} \xrightarrow{\gamma} \bigcup_{j=1}^{m} R_j$ was added by the saturation procedure of the $(i-1)$-th iteration of $loop_1$. Because of line 11, we get that $p^{i-1} \xrightarrow{\gamma} \pi^{-1}(\pi^i(S))$ existed in the i-th iteration of $loop_1$.

Now we consider that the transition rule is added by line 11. Suppose the transition rule $p^i \xrightarrow{\gamma} S$ is substituted by $p^i \xrightarrow{\gamma} R$ where $R = \pi^i(S)$, since $\pi^{-1}(\pi^i(R)) = \pi^{-1}(\pi^i(\pi^i(S))) = \pi^{-1}(\pi^i(S))$ and $p^i \xrightarrow{\gamma} S$ is added by the saturation procedure, we still have $p^{i-1} \xrightarrow{\gamma} \pi^{-1}(\pi^i(S))$ in the i-th iteration of $loop_1$.

□

A.4 Proof of Theorem 2

Theorem 2. ***Algorithm*** *1 always terminates and produces* $Y_{\mathcal{BP}}$*.*

In order to proof theorem 2, we first prove some auxiliary lemmas.

Lemma 2. *In* ***Algorithm*** *1, for every* $i \geq 2, \omega \in \Gamma^*$*, if* $p^i \xrightarrow{\omega}_\delta \{q^i\} \cup R$*, or* $p^i \xrightarrow{\omega}_\delta \{q^{i-1}\} \cup R$*, for some* $R \subseteq Q$ *and* $q \in P$*, then there exist* $\langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle \in P \times \Gamma^*$ *s.t.* $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle\}$*.*

Proof: First, we consider the case $\omega = \epsilon$, then $p^i \xrightarrow{\epsilon}_\delta \{q^{i-1}\} \cup R$ or $p^i \xrightarrow{\epsilon}_\delta \{q^i\} \cup R$. Since every transition in the form of $p^i \xrightarrow{\epsilon} \{p^{i-1}\}$ is only added by line 4 for $p \in F$, and there is a transition $p^i \xrightarrow{\epsilon}_\delta \{p^i\}$ according to the definition of the relation $\longrightarrow_\delta$. Then, necessarily, q is equal to p and $R = \emptyset$. So we have $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle\}$ according to the definition of the relation $\Longrightarrow_{\mathcal{BP}}$.

Now, we consider $\omega \in \Gamma^+$. Let k be the number of transition rules added by Algorithm 1 after $i \geq 2$. We can show that $k \geq 1$. Since there does not exist any transition rule in the form of $p^i \xrightarrow{\gamma}_\delta S$ when $k = 0$ and $i \geq 2$. We proceed by induction on k.

- **Basis** $k = 1$: We can show that the first transition rule must be added at the second step. If it does not, then there does not exist any transition rule in the form of $p^2 \xrightarrow{\gamma} S$ and there does not exist any state $p \in P$ s.t. $p \in F$ (Note that if there exists a state $p \in P$ s.t. $p \in F$, then line 4 will add a transition $p^2 \xrightarrow{\epsilon} \{p^1\}$). This implies that the third step will not add any transition rule and the algorithm will terminate at the third step. The proof depends on the case whether the first transition rule is added by line 4, by the saturation procedure (lines 6-8) or by the substitution (line 11).

  - Suppose the first transition rule $p^2 \xrightarrow{\epsilon} \{p^1\}$ is added by line 4. Then the path $p^i \xrightarrow{\omega}_\delta \{q^i\} \cup R$ exists only if $\omega = \epsilon$, this case has been discussed above.

    Let us consider the path $p^i \xrightarrow{\omega}_\delta \{q^{i-1}\} \cup R$ which can be decomposed as $p^2 \xrightarrow{\epsilon} p^1 \xrightarrow{\omega}_\delta \{q^1\} \cup R$. It is sufficient to prove that $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle\}$ for every $p^1 \xrightarrow{\omega}_\delta \{q^1\} \cup R$. Let m be the number of transition rules added by the saturation procedure at the first step. Since each transition rule added by line 4 is in the form of $p^1 \xrightarrow{\epsilon} \{q_f\}$, then there does not exist any path in the form of $p^1 \xrightarrow{\omega}_\delta \{q^1\} \cup R$. We proceed by induction on m to show that $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle\}$. We can show that $m \geq 1$. Since there are only transition rules in the form of $q_f \xrightarrow{\gamma} \{q_f\}$ when $m = 0$.

* **Basis** $m = 1$: Let $p^1 \xrightarrow{\gamma} \{q^1\} \cup R$ be the first transition rule added by the saturation procedure at the first step. Then, necessarily, there exists $\langle p, \gamma\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle\}$ according to the definition of the saturation procedure.

* **Step** $m \geq 2$: Let $p'^1 \xrightarrow{\gamma} Q$ be the m^{th} transition rule added by the saturation procedure at the first step and used by $p^1 \xrightarrow{\omega}_\delta \{q^1\} \cup R$. Since if the m^{th} transition is not used by $p^1 \xrightarrow{\omega}_\delta \{q^1\} \cup R$, then $p^1 \xrightarrow{\omega}_\delta \{q^1\} \cup R$ already existed before adding the m^{th} transition rule, by applying the induction hypothesis (induction on m), we obtain that $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle\}$ for every $p^1 \xrightarrow{\omega}_\delta \{q^1\} \cup R$. The proof depends on the case whether the transition rule $p'^1 \xrightarrow{\gamma} Q$ is in the form of $p'^1 \xrightarrow{\gamma} \{q_f\}$ or $p'^1 \xrightarrow{\gamma} \{q'^1\} \cup R_1$.

  First, suppose the transition $p'^1 \xrightarrow{\gamma} Q$ is in the form of $p'^1 \xrightarrow{\gamma} \{q_f\}$. Then, there exist $u, v \in \Gamma^*, \gamma \in \Gamma, R_2, R_3, R_4 \subseteq Q$ such that $\omega = u\gamma v$, and
  1. $p^1 \xrightarrow{u}_\delta \{p'^1\} \cup R_2$;
  2. $R_2 \xrightarrow{\gamma v}_\delta R_3$;
  3. $\{q^1\} \cup R = R_3 \cup \{q_f\}$;

  Since $p'^1 \xrightarrow{\gamma} \{q_f\}$ and $q_f \xrightarrow{\gamma'} \{q_f\}$ for every $\gamma' \in \Gamma$ according to the definition of $\mathcal{A}$, we obtain that $p'^1 \xrightarrow{\gamma v}_\delta \{q_f\}$.

  Since $\{q^1\} \cup R = R_3 \cup \{q_f\}$, $q \in P$ and $q_f \notin P$, we get $\{q^1\} \in R_3$. We can show that there exists a state $q'^1 \in R_2$ such that $q'^1 \xrightarrow{\gamma v}_\delta \{q^1\} \cup R_4$ for some $R_4 \subseteq R_3$. By applying the induction hypothesis (induction on m), we get that $\langle q', \gamma v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle q_1, v_1\rangle, ..., \langle q_h, v_h\rangle\}$.

  Since $q'^1 \in R_2$, by applying the induction hypothesis (induction on m) to $p^1 \xrightarrow{u}_\delta \{p'^1\} \cup R_2$, we have $\langle p, u\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q', \epsilon\rangle, \langle q'_1, v'_1\rangle, ..., \langle q'_{h'}, v'_{h'}\rangle\}$. Thus, we obtain that $\langle p, u\gamma v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle q_1, v_1\rangle, ..., \langle q_h, v_h\rangle, \langle q'_1, u'_1\gamma v\rangle, ..., \langle q'_{h'}, u'_{h'}\gamma v\rangle\}$.

  Now, we consider that the transition $p'^1 \xrightarrow{\gamma} Q$ is in the form of $p'^1 \xrightarrow{\gamma} \{q'^1\} \cup R_1$. Then, there exist $u, v \in \Gamma^*, \gamma \in \Gamma, R_2, R_3, R_4 \subseteq Q$ such that $\omega = u\gamma v$, and
  1. $p^1 \xrightarrow{u}_\delta \{p'^1\} \cup R_2$;
  2. $R_2 \xrightarrow{\gamma v}_\delta R_3$;
  3. $\{q'^1\} \cup R_1 \xrightarrow{v}_\delta R_4$;
  4. $\{q^1\} \cup R = R_3 \cup R_4$;

  Since $\{q^1\} \cup R = R_3 \cup R_4$, we get that $\{q^1\} \in R_3$ or $\{q^1\} \in R_4$. The proof depends on the case whether $\{q^1\} \in R_3$ or $\{q^1\} \in R_4$.

  · Suppose $\{q^1\} \in R_3$, then there exists a state $g^1 \in R_2$ s.t. $g^1 \xrightarrow{\gamma v}_\delta \{q^1\} \cup R_5$ and $R_5 \subseteq R_3$. By applying the induction hypothesis (induction on m) to $g^1 \xrightarrow{\gamma v}_\delta \{q^1\} \cup R_5$, we obtain that $\langle g, \gamma v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle g_1, v_1\rangle, ..., \langle g_h, v_h\rangle\}$.

Since $\{g^1\} \in R_2$, by applying the induction hypothesis (induction on m) to $p^1 \xrightarrow{u}_\delta \{p'^1\} \cup R_2$, we get that $\langle p, u\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p', \epsilon\rangle, \langle g, \epsilon\rangle, \langle q_1, u_1\rangle, ..., \langle q_{h'}, u_{h'}\rangle\}$.

Since $p'^1 \xrightarrow{\gamma} \{q'^1\} \cup R_1$ is added by the saturation procedure at the first step, then we obtain that $\langle p', \gamma\rangle \hookrightarrow \{\langle p'_1, \omega'_1\rangle, ..., \langle p'_{n'}, \omega'_{n'}\rangle\}$, and $p'_j \xrightarrow{\omega'_j}_\delta S_j$ for every $1 \leq j \leq n'$ s.t. $\{q'^1\} \cup R_1 = \bigcup_{j=1}^{n'} S_j$. Then, necessarily, there exist $1 \leq j \leq n'$ s.t. $\{q'^1\} \in S_j$. W.l.o.g., we suppose $\{q'^1\} \in S_1$. Let $S_1 = \{q'^1\} \cup S$, then $p'_1 \xrightarrow{\omega'_1}_\delta \{q'^1\} \cup S$. By applying the induction hypothesis (induction on m) to $p'_1 \xrightarrow{\omega'_1}_\delta \{q'^1\} \cup S$ (since it existed before adding the m^{th} transition rule), we get that $\langle p'_1, \omega'_1\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q', \epsilon\rangle, \langle p''_1, \omega''_1\rangle, ..., \langle p''_{n''}, \omega''_{n''}\rangle\}$. Then, we have $\langle p', \gamma\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q', \epsilon\rangle, \langle p''_1, \omega''_1\rangle, ..., \langle p''_{n''}, \omega''_{n''}\rangle, \langle p'_2, \omega'_2\rangle, ..., \langle p'_{n'}, \omega'_{n'}\rangle\}$. Thus, we obtain that $\langle p, u\gamma v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p', \gamma v\rangle, \langle q, \epsilon\rangle, \langle g_1, v_1\rangle, ..., \langle g_h, v_h\rangle, \langle q_1, u_1\gamma v\rangle, ..., \langle q_{h'}, u_{h'}\gamma v\rangle\}$.

- Suppose $\{q^1\} \in R_4$, then $q'^1 \xrightarrow{v}_\delta \{q^1\} \cup R_5$ and $R_5 \subseteq R_4$ or there exists $g^1 \in R_1$ s.t. $g^1 \xrightarrow{v}_\delta \{q^1\} \cup R_5$ and $R_5 \subseteq R_4$. The proof depends on the case whether $q'^1 \xrightarrow{v}_\delta \{q^1\} \cup R_5$ and $R_5 \subseteq R_4$ or there exists $g^1 \in R_1$ s.t. $g^1 \xrightarrow{v}_\delta \{q^1\} \cup R_5$ and $R_5 \subseteq R_4$.

Case 1: Suppose $q'^1 \xrightarrow{v}_\delta \{q^1\} \cup R_5$, then by applying the induction hypothesis (induction on m), we obtain that $\langle q', v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle q_1, v_1\rangle, ..., \langle q_h, v_h\rangle\}$.

Since $p'^1 \xrightarrow{\gamma} \{q'^1\} \cup R_1$ is added by the saturation procedure at the first step, then we obtain that $\langle p', \gamma\rangle \hookrightarrow \{\langle p'_1, \omega'_1\rangle, ..., \langle p'_{n'}, \omega'_{n'}\rangle\}$, and $p'_j \xrightarrow{\omega'_j}_\delta S_j$ for every $1 \leq j \leq n'$ s.t. $\{q'^1\} \cup R_1 = \bigcup_{j=1}^{n'} S_j$. Then, necessarily, there exist $1 \leq j \leq n'$ s.t. $\{q'^1\} \in S_j$. W.l.o.g., we suppose $\{q'^1\} \in S_1$. Let $S_1 = \{q'^1\} \cup S$, then $p'_1 \xrightarrow{\omega'_1}_\delta \{q'^1\} \cup S$. By applying the induction hypothesis (induction on m) to $p'_1 \xrightarrow{\omega'_1}_\delta \{q'^1\} \cup S$ (since it existed before adding the m^{th} transition rule), we get that $\langle p'_1, \omega'_1\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q', \epsilon\rangle, \langle p''_1, \omega''_1\rangle, ..., \langle p''_{n''}, \omega''_{n''}\rangle\}$. Then, we have $\langle p', \gamma\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q', \epsilon\rangle, \langle p''_1, \omega''_1\rangle, ..., \langle p''_{n''}, \omega''_{n''}\rangle, \langle p'_2, \omega'_2\rangle, ..., \langle p'_{n'}, \omega'_{n'}\rangle\}$. Thus, we obtain that $\langle p', \gamma v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle q_1, v_1\rangle, ..., \langle q_h, v_h\rangle, \langle p''_1, \omega''_1 v\rangle, ..., \langle p''_{n''}, \omega''_{n''} v\rangle, \langle p'_2, \omega'_2 v\rangle, ..., \langle p'_{n'}, \omega'_{n'} v\rangle\}$.

By applying the induction hypothesis (induction on m) to $p^1 \xrightarrow{u}_\delta \{p'^1\} \cup R_2$, we get that $\langle p, u\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p', \epsilon\rangle, \langle q_1, u_1\rangle, ..., \langle q_{h'}, u_{h'}\rangle\}$. Hence, $\langle p, u\gamma v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle q_1, v_1\rangle, ..., \langle q_h, v_h\rangle, \langle p''_1, \omega''_1 v\rangle, ..., \langle p''_{n''}, \omega''_{n''} v\rangle, \langle p'_2, \omega'_2 v\rangle, ..., \langle p'_{n'}, \omega'_{n'} v\rangle, \langle q_1, u_1\gamma v\rangle, ..., \langle q_{h'}, u_{h'}\gamma v\rangle\}$.

Case 2: Suppose there exists $g^1 \in R_1$ s.t. $g^1 \xrightarrow{v}_\delta \{q^1\} \cup R_5$ and $R_5 \subseteq R_4$, then by applying the induction hypothesis (induction on m), we obtain that $\langle g, v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle g_1, v_1\rangle, ..., \langle g_h, v_h\rangle\}$.

Since $p'^1 \xrightarrow{\gamma} \{q'^1\} \cup R_1$ is added by the saturation procedure at the first step, then we obtain that $\langle p', \gamma\rangle \hookrightarrow \{\langle p'_1, \omega'_1\rangle, ..., \langle p'_{n'}, \omega'_{n'}\rangle\}$, and $p'_j \xrightarrow{\omega'_j}_\delta S_j$ for every $1 \leq j \leq n'$ s.t. $\{q'^1\} \cup R_1 = \bigcup_{j=1}^{n'} S_j$. Then, necessarily, there exist $1 \leq j \leq n'$ s.t. $\{g^1\} \in S_j$. W.l.o.g., we suppose $\{g^1\} \in S_1$. Let $S_1 = \{g^1\} \cup S$, then $p'_1 \xrightarrow{\omega'_1}_\delta \{g^1\} \cup S$. By applying the induction hypothesis (induction on m) to $p'_1 \xrightarrow{\omega'_1}_\delta \{g^1\} \cup S$ (since it existed before adding the m^{th} transition rule), we get that $\langle p'_1, \omega'_1\rangle \Longrightarrow_{\mathcal{BP}} \{\langle g, \epsilon\rangle, \langle p''_1, \omega''_1\rangle, ..., \langle p''_{n''}, \omega''_{n''}\rangle\}$. Then, we have $\langle p', \gamma\rangle \Longrightarrow_{\mathcal{BP}} \{\langle g, \epsilon\rangle, \langle p''_1, \omega''_1\rangle, ..., \langle p''_{n''}, \omega''_{n''}\rangle, \langle p'_2, \omega'_2\rangle, ..., \langle p'_{n'}, \omega'_{n'}\rangle\}$. Thus, we obtain that $\langle p', \gamma v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle g_1, v_1\rangle, ..., \langle g_h, v_h\rangle, \langle p''_1, \omega''_1 v\rangle, ..., \langle p''_{n''}, \omega''_{n''} v\rangle, \langle p'_2, \omega'_2 v\rangle, ..., \langle p'_{n'} v, \omega'_{n'} v\rangle\}$.

By applying the induction hypothesis (induction on m) to $p^1 \xrightarrow{u}_\delta \{p'^1\} \cup R_2$, we get that $\langle p, u\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p', \epsilon\rangle, \langle q_1, u_1\rangle, ..., \langle q_{h'}, u_{h'}\rangle\}$. Hence, $\langle p, u\gamma v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle g_1, v_1\rangle, ..., \langle g_h, v_h\rangle, \langle p''_1, \omega''_1 v\rangle, ..., \langle p''_{n''}, \omega''_{n''} v\rangle, \langle p'_2, \omega'_2 v\rangle, ..., \langle p'_{n'}, \omega'_{n'} v\rangle, \langle q_1, u_1\gamma v\rangle, ..., \langle q_{h'}, u_{h'}\gamma v\rangle\}$.

- Suppose the **first** transition rule $p'^2 \xrightarrow{\gamma} R'$ is added by the saturation procedure. Then we get that there does not exist any transition rule added by line 4 at the second step, which implies that there only exist $p^2 \xrightarrow{\epsilon}_\delta \{p^2\}$ according to the definition of the relation $\longrightarrow_\delta$. Then there does not exist a path in the form of $p^2 \xrightarrow{\omega}_\delta \{q^1\} \cup R$.

  Since $p'^2 \xrightarrow{\gamma} R'$ is added by the saturation procedure, then we obtain that $\langle p', \gamma\rangle \hookrightarrow \{\langle p'_1, \omega'_1\rangle, ..., \langle p'_{n'}, \omega'_{n'}\rangle\}$, and $p'_j \xrightarrow{\omega'_j}_\delta S_j$ for every $1 \leq j \leq n'$ s.t. $R = \bigcup_{j=1}^{n'} S_j$. Then, necessarily, $\omega'_j = \epsilon$ and $S_j = \{p'_j\}$ for every $1 \leq j \leq n'$. Hence $\langle p', \gamma\rangle \hookrightarrow \{\langle p'_1, \epsilon\rangle, ..., \langle p'_{n'}, \epsilon\rangle\}$, which implies that $\langle p, \omega\rangle \hookrightarrow \{\langle q, \epsilon\rangle, \langle p_1, \epsilon\rangle, ..., \langle p_n, \epsilon\rangle\}$ for every $p^2 \xrightarrow{\omega}_\delta \{q^2\} \cup R$.
- The first transition rule is added by the substitution (line 11). This case is not possible. Indeed, since the substitution can be fired iff there are some transition rules in the form of $p'^2 \xrightarrow{\gamma} R'$ which should be added by the saturation procedure. Since there does not exist any transition rule added by the saturation procedure. Thus, the substitution will not add any transition rule.

– **Step** $k \geq 2$: Let i be the step when the k^{th} transition rule is added. We note that the k^{th} transition can be added by (1) the line 4, (2) the saturation procedure (lines 6-8) and (3) the substitution (line 11). We show that $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle\}$ by depending on the case whether the k^{th} transition is added by line 4, the saturation procedure or the substitution.

- **Case** (1): Suppose the k^{th} transition rule $p^i \xrightarrow{\epsilon} p^{i-1}$ is added by line 4. Then there does not exist any path in the form of $p^i \xrightarrow{\omega}_\delta \{q^i\} \cup R$ (Indeed there does not exist any transition rule in the form of $p^i \xrightarrow{\gamma} \{q^i\} \cup R$), and every path in the form of $p^i \xrightarrow{\omega}_\delta \{q^i\} \cup R$ can be decomposed as $p^i \xrightarrow{\epsilon} p^{i-1} \xrightarrow{\omega}_\delta \{q^{i-1}\} \cup R$. By applying the induction hypothesis (induction on k) to $p^{i-1} \xrightarrow{\omega}_\delta \{q^{i-1}\} \cup R$, we obtain $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle\}$.

- **Case** (2): Suppose the k^{th} transition rule $t = p'^i \xrightarrow{\gamma} S$ for some $S \subseteq Q$ is added by the saturation procedure. W.l.o.g., we suppose that this transition rule is used by $p^i \xrightarrow{\omega}_\delta \{q^i\} \cup R$ or $p^i \xrightarrow{\omega}_\delta \{q^{i-1}\} \cup R$. Indeed, if the transition t is not used, we can apply the induction hypothesis (induction on k) to obtain $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle\}$. Then there exist $u, v \in \Gamma^*, R_1, R_2, R_3 \subseteq Q$ such that $\omega = u\gamma v$, and
  1. $p^i \xrightarrow{u}_\delta \{p'^i\} \cup R_1$;
  2. $R_1 \xrightarrow{\gamma v}_\delta R_2$;
  3. $S \xrightarrow{v}_\delta R_3$;
  4. $\{q^i\} \cup R = R_2 \cup R_3$ for $p^i \xrightarrow{\omega}_\delta \{q^i\} \cup R$,
  5. $\{q^{i-1}\} \cup R = R_2 \cup R_3$ for $p^i \xrightarrow{\omega}_\delta \{q^{i-1}\} \cup R$;

  Let $\alpha \in \{i, i-1\}$, since $\{q^\alpha\} \cup R = R_2 \cup R_3$, then we get that $\{q^\alpha\} \in R_2$ or $\{q^\alpha\} \in R_3$. The proof depends on the case whether $\{q^\alpha\} \in R_2$ or $\{q^\alpha\} \in R_3$.

  * Suppose $\{q^\alpha\} \in R_2$, then there exists a state $g^\beta \in R_1$ such that $g^\beta \xrightarrow{\gamma v}_\delta \{q^\alpha\} \cup R_4$ and $R_4 \subseteq R_2$ where $\beta \in \{i, i-1\}$. Note that if $\alpha = i$, then β must be i. Since there does not exist any transition rule in the form of $p'^{i-1} \xrightarrow{\gamma'} \{q'^i\} \cup Q$.

    By applying the induction hypothesis (induction on k) to $g^\beta \xrightarrow{\gamma v}_\delta \{q^\alpha\} \cup R_4$, we obtain that $\langle g, \gamma v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle p'_1, \omega'_1\rangle, ..., \langle p'_{n'}, \omega'_{n'}\rangle\}$. Since $g^\beta \in R_1$, by applying the induction hypothesis (induction on k) to $p^i \xrightarrow{u}_\delta \{p'^i\} \cup R_1$, we obtain $\langle p, u\rangle \Longrightarrow_{\mathcal{BP}} \{\langle g, \epsilon\rangle, \langle p''_1, \omega''_1\rangle, ..., \langle p''_{n''}, \omega''_{n''}\rangle\}$. Thus we have $\langle p, u\gamma v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle p'_1, \omega'_1\rangle, ..., \langle p'_{n'}, \omega'_{n'}\rangle, \langle p''_1, \omega''_1\gamma v\rangle, ..., \langle p''_{n''}, \omega''_{n''}\gamma v\rangle\}$.

  * Suppose $\{q^\alpha\} \in R_3$, then there exists a state $g^\beta \in S$ such that $g^\beta \xrightarrow{v}_\delta \{q^\alpha\} \cup R_4$ and $R_4 \subseteq R_3$ where $\beta \in \{i, i-1\}$. Note that if $\alpha = i$, then β must be i. Since there does not exist any transition rule in the form of $p'^{i-1} \xrightarrow{\gamma'} \{q'^i\} \cup Q$.

    By applying the induction hypothesis (induction on k) to $g^\beta \xrightarrow{v}_\delta \{q^\alpha\} \cup R_4$, we obtain that $\langle g, v\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon\rangle, \langle p'_1, \omega'_1\rangle, ..., \langle p'_{n'}, \omega'_{n'}\rangle\}$.

    By applying the induction hypothesis (induction on k) to $p^i \xrightarrow{u}_\delta \{p'^i\} \cup R_1$, we obtain $\langle p, u\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p', \epsilon\rangle, \langle p''_1, \omega''_1\rangle, ..., \langle p''_{n''}, \omega''_{n''}\rangle\}$.

    Since the transition rule $p'^i \xrightarrow{\gamma} S$ is added by the saturation procedure, then there exists $\langle p', \gamma\rangle \hookrightarrow \{\langle q_1, u_1\rangle, ..., \langle q_h, u_h\rangle\}$ s.t. $q^i_j \xrightarrow{u_j}_\delta S_j$ for every $1 \le j \le h$ and $S = \bigcup_{j=1}^h S_j$. Since $g^\beta \in S$, then there exists $1 \le j \le h$ s.t.

$g^{\beta} \in S_j$. W.l.o.g., we suppose $g^{\beta} \in S_1$. Let $S_1 = \{g^{\alpha}\} \cup S'$ for some $S' \subseteq Q$. By applying the induction hypothesis (induction on k) to $q_1^i \xrightarrow{u_1}_{\delta} \{g^{\alpha}\} \cup S'$, we obtain that $\langle q_1, u_1 \rangle \Longrightarrow_{\mathcal{BP}} \{\langle g, \epsilon \rangle, \langle q'_1, u'_1 \rangle, ..., \langle q'_{h'}, u'_{h'} \rangle\}$. Thus we have $\langle p', \gamma \rangle \Longrightarrow_{\mathcal{BP}} \{\langle g, \epsilon \rangle, \langle q'_1, u'_1 \rangle, ..., \langle q'_{h'}, u'_{h'} \rangle, \langle q_2, u_2 \rangle, ..., \langle q_h, u_h \rangle\}$.

Hence, we obtain that $\langle p, u\gamma v \rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon \rangle, \langle p'_1, \omega'_1 \rangle, ..., \langle p'_{n'}, \omega'_{n'} \rangle, \langle q'_1, u'_1 v \rangle, ..., \langle q'_{h'}, u'_{h'} v \rangle, \langle q_2, u_2 v \rangle, ..., \langle q_h, u_h v \rangle, \langle p''_1, \omega''_1 \gamma v \rangle, ..., \langle p''_{n''}, \omega''_{n''} \gamma v \rangle\}$.

- **Case** (3): Suppose the k^{th} transition rule $p'^i \xrightarrow{\gamma} \pi^i(S)$ is added by the substitution because there exists a transition rule $p'^i \xrightarrow{\gamma} S$. Let $\alpha \in \{i, i-1\}$. We proceed by induction on the length of ω.
  - ∗ **Basis** $|\omega| = 0$: then the path $p^i \xrightarrow{\omega}_{\delta} \{q^{\alpha}\} \cup R$ does not use the transition rule $p'^i \xrightarrow{\gamma} \pi^i(S)$. By applying the induction hypothesis (induction on k), we have $\langle p, \omega \rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon \rangle, \langle p_1, \omega_1 \rangle, ..., \langle p_n, \omega_n \rangle\}$.
  - ∗ **Step** $|\omega| \geq 1$: suppose the path $p^i \xrightarrow{\omega}_{\delta} \{q^{\alpha}\} \cup R$ uses the transition rule $p'^i \xrightarrow{\gamma} \pi^i(S)$. Indeed if it does not, then by applying the induction hypothesis (induction on k), we have $\langle p, \omega \rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon \rangle, \langle p_1, \omega_1 \rangle, ..., \langle p_n, \omega_n \rangle\}$.

    Since $p'^i \xrightarrow{\gamma} \pi^i(S)$ is used by the path $p^i \xrightarrow{\omega}_{\delta} \{q^{\alpha}\} \cup R$, then there exist $u, v \in \Gamma^*, R_1, R_2, R_3 \subseteq Q$ such that $\omega = u\gamma v$, and
    - · $p^i \xrightarrow{u}_{\delta} \{p'^i\} \cup R_1$;
    - · $R_1 \xrightarrow{\gamma v}_{\delta} R_2$;
    - · $\pi^i(S) \xrightarrow{v}_{\delta} R_3$;
    - · $\{q^{\alpha}\} \cup R = R_2 \cup R_3$.

    Since $\{q^{\alpha}\} \cup R = R_2 \cup R_3$, then $\{q^{\alpha}\} \in R_2$ or $\{q^{\alpha}\} \in R_3$. The proof depends on the case whether $\{q^{\alpha}\} \in R_2$ or $\{q^{\alpha}\} \in R_3$.
    - · Suppose $\{q^{\alpha}\} \in R_2$, then these exists a state $g^{\beta} \in R_1$ such that $g^{\beta} \xrightarrow{\gamma v}_{\delta} \{q^{\alpha}\} \cup R_4$ for some $R_4 \subseteq R_2$ and $\beta \in \{i, i-1\}$. By applying the induction hypothesis to $g^{\beta} \xrightarrow{\gamma v}_{\delta} \{q^{\alpha}\} \cup R_4$, we obtain that $\langle g, \gamma v \rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon \rangle, \langle p'_1, \omega'_1 \rangle, ..., \langle p'_{n'}, \omega'_{n'} \rangle\}$. By applying the induction hypothesis to $p^i \xrightarrow{u}_{\delta} \{p'^i\} \cup R_1$, we obtain that $\langle p, u \rangle \Longrightarrow_{\mathcal{BP}} \{\langle g, \epsilon \rangle, \langle p''_1, \omega''_1 \rangle, ..., \langle p''_{n''}, \omega''_{n''} \rangle\}$. Thus, we have $\langle p, u\gamma v \rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon \rangle, \langle p'_1, \omega'_1 \rangle, ..., \langle p'_{n'}, \omega'_{n'} \rangle, \langle p''_1, \omega''_1 \gamma v \rangle, ..., \langle p''_{n''}, \omega''_{n''} \gamma v \rangle\}$.
    - · Suppose $\{q^{\alpha}\} \in R_3$, then there exists a state $g^i \in \pi^i(S)$ such that $g^i \xrightarrow{v}_{\delta} \{q^{\alpha}\} \cup R_4$ for some $R_4 \subseteq R_3$. By applying the induction hypothesis to $g^i \xrightarrow{v}_{\delta} \{q^{\alpha}\} \cup R_4$, we obtain that $\langle g, v \rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon \rangle, \langle p'_1, \omega'_1 \rangle, ..., \langle p'_{n'}, \omega'_{n'} \rangle\}$.

      By applying the induction hypothesis to $p^i \xrightarrow{u}_{\delta} \{p'^i\} \cup R_1$, we obtain that $\langle p, u \rangle \Longrightarrow_{\mathcal{BP}} \{\langle p', \epsilon \rangle, \langle p''_1, \omega''_1 \rangle, ..., \langle p''_{n''}, \omega''_{n''} \rangle\}$.

      Since $p'^i \xrightarrow{\gamma} \pi^i(S)$ is added because the transition $p'^i \xrightarrow{\gamma} S$ already exist at the saturation procedure. Since $g^i \in \pi^i(S)$, according to the definition of the projection function π^i, we obtain that there exists $\beta \in \{i, i-1\}$

such that $g^{\beta} \in S$. By applying the induction hypothesis (induction on k) to $p'^{i} \xrightarrow{\gamma} S$, we obtain that $\langle p', \gamma \rangle \Longrightarrow_{\mathcal{BP}} \{\langle g, \epsilon \rangle, \langle q_1, u_1 \rangle, ..., \langle q_h, u_h \rangle\}$.

Thus, we have $\langle p, u\gamma v \rangle \Longrightarrow_{\mathcal{BP}} \{\langle q, \epsilon \rangle, \langle p'_1, \omega'_1 \rangle, ..., \langle p'_{n'}, \omega'_{n'} \rangle, \langle q_1, u_1 v \rangle, ..., \langle q_h, v_h v \rangle, \langle p''_1, \omega''_1 \gamma v \rangle, ..., \langle p''_{n''}, \omega''_{n''} \gamma v \rangle\}$

□

Lemma 3. *Let n be the first number in Algorithm 1 such that for every $p \in P, \gamma \in \Gamma, S \subseteq P \times \{n+1\} \cup \{q_f\}, p^{n+1} \xrightarrow{\gamma} S \in \delta \Longleftrightarrow p^n \xrightarrow{\gamma} \pi^{-1}(S) \in \delta$. If we remove the termination condition of $loop_1$, then for every $i \geq n, L(A_{i+1}) = L(A_n)$.*

Proof: Since line 11 of Algorithm 1 will replace $p^{i+1} \xrightarrow{\gamma} S$ by $p^{i+1} \xrightarrow{\gamma} \pi^{i+1}(S)$, then each path $p^{i+1} \xrightarrow{\omega}_{\delta} \{q_f\}$ only uses states of $P \times \{i+1\} \cup \{q_f\}$. In order to prove $L(A_{i+1}) = L(A_n)$, for every $i \geq n$, it is sufficient to prove that for every $p \in P$, $\gamma \in \Gamma, p^{i+1} \xrightarrow{\gamma} \{q_1^{i+1}, ..., q_m^{i+1}\} \in \delta \Longleftrightarrow p^n \xrightarrow{\gamma} \{q_1^n, ..., q_m^n\} \in \delta$ by applying induction on i.

– **Basis**. $i = n$. We get directly from the condition of n that

$$p^{n+1} \xrightarrow{\gamma} \{q_1^{n+1}, ..., q_m^{n+1}\} \in \delta \Longleftrightarrow p^n \xrightarrow{\gamma} \{q_1^n, ..., q_m^n\} \in \delta \quad (0)$$

– **Step**. $i > n$. Since the transition rule
$p^{i+1} \xrightarrow{\gamma} \{q_1^{i+1}, ..., q_m^{i+1}\}$ is added based on Δ and A_i, for every $p \in P$, $\gamma \in \Gamma$. (1)
$p^{n+1} \xrightarrow{\gamma} \{q_1^{n+1}, ..., q_m^{n+1}\}$ is added based on Δ and A_n, for every $p \in P$, $\gamma \in \Gamma$ (2)

By applying the induction hypothesis (induction on i): we have
for every $p \in P, \gamma \in \Gamma$, $p^i \xrightarrow{\gamma} \{q_1^i, ..., q_m^i\} \in \delta \Longleftrightarrow p^n \xrightarrow{\gamma} \{q_1^n, ..., q_m^n\} \in \delta$ (3)

From (1), (2) and (3), we obtain:

$$p^{i+1} \xrightarrow{\gamma} \{q_1^{i+1}, ..., q_m^{i+1}\} \in \delta \Longleftrightarrow p^{n+1} \xrightarrow{\gamma} \{q_1^{n+1}, ..., q_m^{n+1}\} \in \delta$$

From (0), we get $p^{i+1} \xrightarrow{\gamma} \{q_1^{i+1}, ..., q_m^{i+1}\} \in \delta \Longleftrightarrow p^n \xrightarrow{\gamma} \{q_1^n, ..., q_m^n\} \in \delta$.

□

Proof of Theorem 2:
Proof: we prove termination and correctness.
Termination: There are two loops in **Algorithm** 1, we need to prove that both loops terminate.

Loop$_2$: Suppose $loop_2$ is in the i-th iteration of $loop_1$. Due to the line 11, there is no $S \nsubseteq P \times \{i-1\} \cup \{q_f\}$ such that $p^{i-1} \xrightarrow{\gamma} S \in \delta$.

Since the addition of the ϵ-transition does not introduce any new state into $\mathcal{A}$, then the number of states of $\mathcal{A}$ is finite. This implies that there is a finite number of transition rules from the initial states $P \times \{i\}$ in $\mathcal{A}$. Hence $loop_2$ will always terminate.

$Loop_1$: Now we consider the termination of $loop_1$. As discussed above, the number of transition rules in $\mathcal{A}$ is bounded. By **Proposition** 2, the number of transition rules in the $(i+1)$-th iteration is smaller than in the i-th iteration where $i \geq 1$. Thus **Algorithm** 1 will always terminate.

Correctness: Let A_i denote the AMA $\mathcal{A}$ in the i-th iteration of $loop_1$, let $L(A_i)$ represent the configurations recognized by A_i from the initial states $P \times \{i\}$. Let n be the fixpoint of Algorithm 1 such that for every $p \in P, \gamma \in \Gamma,\ S \subseteq P \times \{n+1\} \cup \{q_f\}, p^{n+1} \xrightarrow{\gamma} S \in \delta \Longleftrightarrow p^n \xrightarrow{\gamma} \pi^{-1}(S) \in \delta$. Then $L(A_n) = L(A_{n+1})$. We will prove $L(A_n) = Y_{\mathcal{BP}}$.

Suppose Algorithm 1 removed the termination condition of $loop_1$, then there is an infinite sequence of AMA A_i for every $i \geq 0$. By the **Lemma** 3, we have

$$L(A_i) = L(A_{i-1}) =, ..., = L(A_n) \subseteq L(A_{n-1}) \subseteq, ..., \subseteq L(A_1) \qquad (1).$$

($\subseteq$) We prove $L(A_n) \subseteq Y_{\mathcal{BP}}$. From (1), the definition of $Y_{\mathcal{BP}} = \bigcap_{i\geq 0} X_i$ and the definition of $X_{i+1} = Pre^+(X_i \cap F \times \Gamma^*)$: it is sufficient to that prove $L(A_i) \subseteq X_i$ for every $i \geq 1$. We proceed by induction on i.

- **Basis**. $i = 1$. We will show that $L(A_1) \subseteq X_1$. Since $X_1 = Pre^+(X_0 \cap F \times \Gamma^*) = Pre^+(F \times \Gamma^*)$, by the initialization of **Algorithm** 1: $L(A_0) = P \times \Gamma^*$, then we get that

  $$X_1 = Pre^+(L(A_0)) \qquad (2)$$

  From the algorithm 1, A_1 is constructed based on A_0, after the saturation procedure (lines 5-9), we have $Pre^*(L(A_0))$.

  By line 10, the ϵ-transition rules are removed, we get $Pre^+(L(A_0))$.

  By the definition of function π^i: line 11 of **Algorithm** 1 will not change any transition rule at the end of the first iteration of $loop_1$. Hence we have $L(A_1) = X_1$ (3)

  We get $L(A_1) \subseteq X_1$.
- **Step**. $i \geq 2$. We will show that $L(A_i) \subseteq X_i$. By applying the induction hypothesis (induction on i): we get $L(A_{i-1}) \subseteq X_{i-1}$. By the definition of $X_i = Pre^+(X_{i-1} \cap F \times \Gamma^*)$, we obtain

  $$Pre^+(L(A_{i-1}) \cap F \times \Gamma^*) \subseteq X_i \qquad (4)$$

  Before line 11 of the algorithm: we have $Pre^+(L(A_{i-1}) \cap F \times \Gamma^*)$. By **Proposition** 2, the line 11 can only reduce the language of A_i, we obtain

  $$L(A_i) \subseteq Pre^+(L(A_{i-1}) \cap F \times \Gamma^*). \qquad (5)$$

  From (4) and (5): we get $L(A_i) \subseteq X_i$.

($\supseteq$) We show $Y_{\mathcal{BP}} \subseteq L(A_n)$. From (1), it is sufficient to prove that $Y_{\mathcal{BP}} \subseteq L(A_i)$ for every $i \geq 1$. We proceed by induction on i.

– **Basis**. $i = 1$. We show that $Y_{\mathcal{BP}} \subseteq L(A_1)$. By the definition of $Y_{\mathcal{BP}} = \bigcap_{i\geq 0} X_i$ and the definition of $X_{i+1} = Pre^+(X_i \cap F \times \Gamma^*)$, we get $Y_{\mathcal{BP}} \subseteq X_1$. (6)

From (3) and (6): we obtain that $Y_{\mathcal{BP}} \subseteq L(A_1)$.

– **Step**. $i \geq 2$. Since $Y_{\mathcal{BP}} = Pre^+(Y_{\mathcal{BP}} \cap F \times \Gamma^*)$ and by the induction hypothesis $Y_{\mathcal{BP}} \subseteq L(A_{i-1})$, we get $Y_{\mathcal{BP}} \subseteq Pre^+(L(A_{i-1}) \cap F \times \Gamma^*)$. (7)

From the **Algorithm** 1, $Pre^+(L(A_{i-1}) \cap F \times \Gamma^*)$ represents the configurations accepted by $\mathcal{A}$ after line 10. We will prove that for every configuration $\langle p, \omega\rangle \in Pre^+(L(A_{i-1}) \cap F \times \Gamma^*)$, if $\langle p, \omega\rangle \notin L(A_i)$, then $\langle p, \omega\rangle \notin Y_{\mathcal{BP}}$, which implies that the configurations removed by line 11 are not in $Y_{\mathcal{BP}}$. We proceed by induction on $|\omega|$ the length of ω. Note that in this case $|\omega| \geq 2$, because transition rules in the form of $p^i \xrightarrow{\epsilon}_\delta q_f$ do not exist after line 10 and line 11 does not change transition rules of the form $p^i \xrightarrow{\gamma}_\delta q_f$.

- **Basis**. $|\omega| = 2$. There exist $\gamma_1, \gamma_2 \in \Gamma$ such that $\omega = \gamma_1\gamma_2$. Since $\langle p, \omega\rangle \in Pre^+(L(A_{i-1}) \cap F \times \Gamma^*)$, we have $r = p^i \xrightarrow{\gamma_1} R \xrightarrow{\gamma_1} q_f$ after line 10.

  Since $\langle p, \omega\rangle \notin L(A_i)$, then there exists a state q^{i-1} in R such that $q^i \xrightarrow{\gamma_2} q_f$ is not in A_i. Since line 11 does not change transition rules of the form $q^i \xrightarrow{\gamma_2} q_f$, we get that $\langle q, \gamma_2\rangle \notin Pre^+(L(A_{i-1}) \cap F \times \Gamma^*)$. From (7), we obtain that $\langle q, \gamma_2\rangle \notin Y_{\mathcal{BP}}$.

  By Lemma 2, we obtain that $\langle p, \gamma_1\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle, \langle q, \epsilon\rangle\}$. Then we have $\langle p, \gamma_1\gamma_2\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p_1, \omega_1\gamma_2\rangle, ..., \langle p_n, \omega_n\gamma_2\rangle, \langle q, \gamma_2\rangle\}$ such that $\langle q, \gamma_2\rangle \notin Y_{\mathcal{BP}}$.

  Since $Y_{\mathcal{BP}} = Pre^+(Y_{\mathcal{BP}} \cap F \times \Gamma^*)$, suppose there exists a path $t = \langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q_1, u_1\rangle, ..., \langle q_m, u_m\rangle\}$ such that $\langle p, \omega\rangle \in Y_{\mathcal{BP}}$ and for each $1 \leq j \leq m$, $\langle q_j, u_j\rangle \in Y_{\mathcal{BP}}$.

  By the definition of the saturation procedure there exists a corresponding path r' in the form of $p^i \xrightarrow{\omega} q_f$ in A_i after line 10.

  We apply the same reasoning to path r', we obtain that there exists necessarily an index j s.t. $1 \leq j \leq m$ and $\langle q_j, u_j\rangle \notin Y_{\mathcal{BP}}$. This implies that for every run of the form $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q_1, u_1\rangle, ..., \langle q_m, u_m\rangle\}$, there exists $1 \leq j \leq m$ s.t. $\langle q_j, u_j\rangle \notin Y_{\mathcal{BP}}$. Since $Y_{\mathcal{BP}} = Pre^+(Y_{\mathcal{BP}} \cap F \times \Gamma^*)$, we get $\langle p, \omega\rangle \notin Y_{\mathcal{BP}}$.

- **Step**. $|\omega| \geq 3$. Then $p^i \xrightarrow{\omega}_\delta q_f$; existed after line 10. Since $\langle p, \omega\rangle \notin L(A_i)$ and there does not exist any transition in the form of $p^i \xrightarrow{\epsilon}_\delta \{p^{i-1}\}$ for every $p \in P$ after line 10, there exist $u, v \in \Gamma^*, \gamma \in \Gamma, R_1, R_2 \subseteq Q$ such that $\omega = u\gamma v$, and
  1. $p^i \xrightarrow{u}_\delta \{p'^i\} \cup R_1$;
  2. $p'^i \xrightarrow{\gamma} \{q^{i-1}\} \cup R_2 \xrightarrow{v}_\delta q_f$;
  3. $R_1 \xrightarrow{\gamma v}_\delta q_f$;

4. $q^i \xrightarrow{u}_\delta q_f$ is not in A_i;

Item 4 implies that $\langle q, u\rangle \notin L(A_i)$. Let's first show that $\langle q, u\rangle \notin Y_{\mathcal{BP}}$.

- If $q^i \xrightarrow{u}_\delta q_f$ exists after line 10, then we get $\langle q, u\rangle \in Pre^+(L(A_i) \cap F \times \Gamma^*)$, by applying the induction hypothesis to $\langle q, u\rangle$, we get $\langle q, u\rangle \notin Y_{\mathcal{BP}}$.

- If $q^i \xrightarrow{u}_\delta q_f$ does not exist after line 10, then $\langle q, u\rangle \notin Pre^+(L(A_i) \cap F \times \Gamma^*)$. Since $Y_{\mathcal{BP}} \subseteq Pre^+(L(A_i) \cap F \times \Gamma^*)$, we obtain that $\langle q, u\rangle \notin Y_{\mathcal{BP}}$.

Since $q^i \xrightarrow{u}_\delta q_f$ does not exist after the line 11, we get $q \neq q_f$. By applying **Lemma** 2 to Item (1) and $p'^i \xrightarrow{\gamma} q^{i-1} \cup R_2$, we obtain that $\langle p, u\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p'_1, \omega'_1\rangle, ..., \langle p'_{n'}, \omega'_{n'}\rangle, \langle p', \epsilon\rangle\}$ and $\langle p', \gamma\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p_1, \omega_1\rangle, ..., \langle p_n, \omega_n\rangle, \langle q, \epsilon\rangle\}$. Then we have $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle p_1, \omega_1 v\rangle, ..., \langle p_n, \omega_n v\rangle, \langle p'_1, \omega'_1 \gamma v\rangle, ..., \langle p'_{n'}, \omega'_{n'} \gamma v\rangle, \langle q, v\rangle\}$ such that $\langle q, v\rangle \notin Y_{\mathcal{BP}}$.

Since $Y_{\mathcal{BP}} = Pre^+(Y_{\mathcal{BP}} \cap F \times \Gamma^*)$, suppose there exist a path $t = \langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q_1, u_1\rangle, ..., \langle q_m, u_m\rangle\}$ such that $\langle p, \omega\rangle \in Y_{\mathcal{BP}}$ and for every $1 \leq j \leq m$: $\langle q_j, u_j\rangle \in Y_{\mathcal{BP}}$.

By the definition of the saturation procedure there exists a corresponding path r' in the form of $p^i \xrightarrow{\omega} q_f$ in A_i after line 10. We apply the same reasoning to path r', we obtain that there exists necessarily an index j s.t. $1 \leq j \leq m$ and $\langle q_j, u_j\rangle \notin Y_{\mathcal{BP}}$. This implies that for every run of the form $\langle p, \omega\rangle \Longrightarrow_{\mathcal{BP}} \{\langle q_1, u_1\rangle, ..., \langle q_m, u_m\rangle\}$, there exists $1 \leq j \leq m$ s.t. $\langle q_j, u_j\rangle \notin Y_{\mathcal{BP}}$. Since $Y_{\mathcal{BP}} = Pre^+(Y_{\mathcal{BP}} \cap F \times \Gamma^*)$, we get $\langle p, \omega\rangle \notin Y_{\mathcal{BP}}$.

□

A.5 Proof of Theorem 3

Theorem 3. *Given an ABPDS $\mathcal{BP} = (P, \Gamma, \Delta, F)$, we can effectively compute an AMA $\mathcal{A}$ with $O(|P|)$ states and $O(|P| \cdot |\Gamma| \cdot 2^{|P|})$ transition rules that recognizes $\mathcal{L}(\mathcal{BP})$. This AMA can be computed in time $O(|P|^2 \cdot |\Delta| \cdot |\Gamma| \cdot 2^{5|P|})$.*

Proof: The correctness follows from **Theorem** 1 and **Theorem** 2.

Complexity: D. Suwimonteerabuth et al implemented an efficient algorithm computing Pre^* of a given AMA A for alternating pushdown systems in $O(n \cdot |\Delta| \cdot 2^{2n})$ time [SSE06], where n is the number of states of A. We integrated this efficient algorithm into our saturation procedure ($loop_2$). First, let's consider the number of states n in our algorithm.

Thanks to the line 11 of **Algorithm** 1, we only need to keep $P \times \{i, i-1\}$ states in the i-th iteration of $loop_1$. Thus the state space of $loop_2$ is at most $2|P| + 1$ which implies that n is equal to $2|P| + 1$.

In line 4 and line 10, adding or removing ϵ-transition rule is executed $|F|$ times.

Since the number of transition rules of $\mathcal{A}$ is at most $|\Gamma| \cdot |P| \cdot 2^{2|P|+1}$ after line 10 and at most $|\Gamma| \cdot |P| \cdot 2^{|P|+1}$ after line 11. The number of times of substitution (line 11) is at most $|\Gamma| \cdot |P| \cdot 2^{2|P|+1}$. The termination condition can be done in time $|\Gamma| \cdot |P| \cdot 2^{|P|+1}$. At each iteration of $loop_1$, the number of transition rules of $\mathcal{A}$ will be smaller and smaller until reaching a fixpoint. The number of times that $loop_1$ is executed is at most $|P| \cdot |\Gamma| \cdot 2^{|P|+1}$.

The global complexity of **Algorithm** 1 is $O(((2|P| + 1) \cdot |\Delta| \cdot 2^{4|P|+2} + |\Gamma| \cdot |P| \cdot 2^{|P|+1} + |\Gamma| \cdot |P| \cdot 2^{2|P|+1}) \cdot |P| \cdot |\Gamma| \cdot 2^{|P|+1})$ simplified to $O(|P|^2 \cdot |\Delta| \cdot |\Gamma| \cdot 2^{5|P|})$. □

A.6 Proof of Theorem 4

Theorem 4. *Let $\mathcal{P} = (P, \Gamma, \Delta, \sharp)$ be a PDS, $f : AP \longrightarrow 2^P$ a labelling function, φ a CTL formula, and $\langle p, \omega \rangle$ a configuration of $\mathcal{P}$. $(\mathcal{P}, \langle p, \omega \rangle) \models_{\lambda_f} \varphi$ iff $\mathcal{BP}_\varphi$ has an accepting run from the configuration $\langle [p, \varphi], \omega \rangle$.*

Proof: Theorem 4 is a special case of **Theorem 5**. We refer the reader to the proof of **Theorem** 5. □

A.7 Proof of Theorem 5

Let us start by introducing the reachability relation of a PDS. Given a PDS $\mathcal{P} = (P, \Gamma, \Delta, \sharp)$, the reachability relation $\Longrightarrow_{\mathcal{P}} \subseteq (P \times \Gamma^*) \times (P \times \Gamma^*)$ is the reflexive and transitive closure of the immediate successor relation. Formally $\Longrightarrow_{\mathcal{P}}$ is defined as follows: (1) $c \Longrightarrow_{\mathcal{P}} c$ for every $c \in P \times \Gamma^*$, (2) if $\langle p, \gamma \rangle \hookrightarrow \langle q, \omega \rangle$, then $\langle p, \gamma\omega' \rangle \Longrightarrow_{\mathcal{P}} \langle q, \omega\omega' \rangle$ for every $\omega' \in \Gamma^*$, (3) if $c \Longrightarrow_{\mathcal{P}} c''$ and $c'' \Longrightarrow_{\mathcal{P}} c'$, then $c \Longrightarrow_{\mathcal{P}} c'$.

Theorem 5. *$(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \varphi$ iff $\mathcal{BP}'_\varphi$ has an accepting run from the configuration $\langle [p, \varphi], \omega \rangle$.*

Proof: ($\Longrightarrow$) Let $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi$, we show that $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi], \omega \rangle$ by induction on the structure of ψ.

Case $\psi = a$: Since $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi$, then $\langle p, \omega \rangle \in \lambda(a)$. By the definition of M_a, M_a has an accepting run from the initial state, $p_a \xrightarrow{\omega}_\delta f$ where $f \in F_a$.

We will prove that $\mathcal{BP}'_\psi$ has an accepting run from $\langle p_a, \omega \rangle$ by induction on m the length of ω which should be greater than 0.

- **Basis**. $m = 1$ (Note that $\sharp$ will never be popped). Then $p_a \xrightarrow{\sharp}_\delta f$. We get $\langle p_a, \sharp \rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle f, \sharp \rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle f, \sharp \rangle$.

  Since f is an accepting control location, $\mathcal{BP}'_\psi$ has an accepting run from $\langle p_a, \sharp \rangle$. In this special case, we have $p_a = f$.

- **Step**. $m \geq 2$. Then there exists $\gamma \in \Gamma, u \in \Gamma^*, q \in Q_a$ such that $\omega = \gamma u$ and

$$p_a \xrightarrow{\gamma} q \xrightarrow{u}_\delta f \text{ in } M_a.$$

By applying the induction hypothesis (induction on m) to $q \xrightarrow{u}_\delta f$, $\mathcal{BP}'_\psi$ has an accepting run from $\langle q, u\rangle$. Since $\langle p_a, \gamma u\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle q, u\rangle$, $\mathcal{BP}'_\psi$ has an accepting from $\langle p_a, \omega\rangle$.

Since $\langle [p, a], \omega\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle p_a, \omega\rangle$, we get that $\mathcal{BP}'_\psi$ has an accepting from $\langle [p, a], \omega\rangle$.

Case $\psi = \neg a$: Since $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda \psi$, then $\langle p, \omega\rangle \notin \lambda(a)$. By the definition of $M_{\neg a}$, $M_{\neg a}$ has an accepting path $p_{\neg a} \xrightarrow{\omega}_\delta f$ where $f \in F_{\neg a}$.

We will prove that $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p, p_{\neg a}], \omega\rangle$ by induction on m the length of ω.

- **Basis**. $m = 1$. then $p_{\neg a} \xrightarrow{\sharp}_\delta f$. Since we have $\langle [p, p_{\neg a}], \sharp\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle f, \sharp\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle f, \sharp\rangle$, $\mathcal{BP}'_\psi$ has an accepting run from $\langle f, \sharp\rangle$.

- **Step**. $m \geq 2$. Then there exists $\gamma \in \Gamma, u \in \Gamma^*, q \in Q_{\neg a}$ such that $\omega = \gamma u$ and

$$p_q \xrightarrow{\gamma} q \xrightarrow{u}_\delta f \text{ in } M_{\neg a}.$$

By the induction hypothesis (induction on m), $\mathcal{BP}'_\psi$ has an accepting from $\langle q, u\rangle$.
Since $\langle p_{\neg a}, \omega\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle q, u\rangle$, $\mathcal{BP}'_\psi$ has an accepting run from $\langle p_{\neg a}, \omega\rangle$.

Since $\langle [p, \psi], \omega\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle p_{\neg a}, \omega\rangle$, $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p, \psi], \omega\rangle$.

Case $\psi = \psi_1 \wedge \psi_2$: Since $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda \psi$, we get $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda \psi_1$ and $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda \psi_2$.

By applying the induction hypothesis: $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi_1], \omega\rangle$, and $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi_2], \omega\rangle$.

Since $\langle [p, \psi], \gamma\rangle \hookrightarrow \langle [p, \psi_1], \gamma\rangle \wedge \langle [p, \psi_2], \gamma\rangle$, we get $\langle [p, \psi], \omega\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle [p, \psi_1], \omega\rangle \wedge \langle [p, \psi_2], \omega\rangle$.

So $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi], \omega\rangle$.

Case $\psi = \psi_1 \vee \psi_2$: Since $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda \psi$, we get $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda \psi_1$ or $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda \psi_2$.

By applying the induction hypothesis: $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi_1], \omega\rangle$ or $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi_2], \omega\rangle$.

Since $\langle [p, \psi], \gamma\rangle \hookrightarrow \langle [p, \psi_1], \gamma\rangle \vee \langle [p, \psi_2], \gamma\rangle$, we get $\langle [p, \psi], \omega\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle [p, \psi_1], \omega\rangle \vee \langle [p, \psi_2], \omega\rangle$.

So $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi], \omega\rangle$.

Case $\psi = EX\psi_1$: Since $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda \psi$, then there exists an immediate successor $\langle p', \omega'\rangle$ of $\langle p, \omega\rangle$, such that $(\mathcal{P}, \langle p', \omega'\rangle) \models_\lambda \psi_1$. By applying the induction hypothesis: $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p', \psi_1], \omega'\rangle$.

Since $\langle [p, \psi], \gamma\rangle \hookrightarrow \bigvee_{\langle p, r\rangle \hookrightarrow \langle p, \omega\rangle} \langle [p', \psi_1], \omega\rangle$, we have $\langle [p, \psi], \omega\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle [p', \psi_1], \omega'\rangle$. Hence $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi], \omega\rangle$.

Case $\psi = AX\psi_1$ is similar to $\psi = EX\psi_1$.

Case $\psi = E[\psi_1 U \psi_2]$: Since $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda E[\psi_1 U \psi_2]$, then there exists a path $\langle p_0, \omega_0\rangle$, $\langle p_1, \omega_1\rangle, \langle p_2, \omega_2\rangle$... from $\langle p, \omega\rangle$ such that there exists $i \geq 0$, $(\mathcal{P}, \langle p_i, \omega_i\rangle) \models_\lambda \psi_2$ and for every $0 \leq j < i, (\mathcal{P}, \langle p_j, \omega_j\rangle) \models_\lambda \psi_1$. Since $(\mathcal{P}, \langle p_i, \omega_i\rangle) \models_\lambda \psi_2$ and $(\mathcal{P}, \langle p_j, \omega_j\rangle) \models_\lambda \psi_1$, for every $0 \leq j < i$. By the induction hypothesis, we have $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_i, \psi_2], \omega_i\rangle$ and

For all $0 \leq j < i$, $\mathcal{BP}'_\psi$ is an accepting run from the configuration $\langle [p_j, \psi_1], \omega_j\rangle$.

Since $\langle [p_i, \psi], \gamma\rangle \hookrightarrow \langle [p_i, \psi_2], \gamma\rangle \vee \bigvee_{\langle p_i, r\rangle \hookrightarrow \langle p', \omega\rangle} (\langle [p_i, \psi_1], \gamma\rangle \wedge \langle [p', \psi], \omega\rangle)$, we get $\langle [p_i, \psi], \omega_i\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle [p_i, \psi_2], \omega_i\rangle$, so $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_i, \psi], \omega_i\rangle$.

If $i = 0$, then $\langle [p, \psi], \omega\rangle = \langle [p_i, \psi], \omega_i\rangle$, $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p, \psi], \omega\rangle$.
Otherwise $i > 0$, we show that $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_j, \psi], \omega_j\rangle$ by induction on $l = i - j$. (Notes that $\langle [p_0, \psi], \omega_0\rangle = \langle [p, \psi], \omega\rangle$.)

- **Basis**. $l = 1$. $\langle p_i, \omega_i\rangle$ is an immediate successor of $\langle p_j, \omega_j\rangle$. Since $\langle [p_j, \psi], \omega_j\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle [p_j, \psi_1], \omega_j\rangle \wedge \langle [p_i, \psi], \omega_i\rangle$, $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_j, \psi], \omega_j\rangle$.

- **Step**. $l > 1$. then there exists $\langle p_{j+1}, \omega_{j+1}\rangle$, such that $\langle p_j, \omega_j\rangle \Longrightarrow_{\mathcal{P}} \langle p_{j+1}, \omega_{j+1}\rangle \Longrightarrow_{\mathcal{P}} \langle p_i, \omega_i\rangle$. By the induction hypothesis (induction on l), we obtain $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_{j+1}, \psi], \omega_{j+1}\rangle$.

  Since $(\mathcal{P}, \langle p_j, \omega_j\rangle) \models_\lambda \psi_1$, by induction hypothesis (induction on the structure of ψ): $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_j, \psi_1], \omega_j\rangle$.

  Since $\langle [p_j, \psi], \omega_j\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle [p_j, \psi_1], \omega_j\rangle \wedge \langle [p_{j+1}, \psi], \omega_{j+1}\rangle$, $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p, \psi], \omega\rangle$.

Case $\psi = A[\psi_1 U \psi_2]$ is similar to $\psi = E[\psi_1 U \psi_2]$.

Case $\psi = E[\psi_1 \tilde{U} \psi_2]$: Since $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda E[\psi_1 \tilde{U} \psi_2]$, according to the semantic of CTL, $\mathcal{P}$ has a path $\langle p_0, \omega_0\rangle, \langle p_1 \omega_1\rangle, \langle p_2, \omega_2\rangle$... from $\langle p, \omega\rangle$ such that

1. for every $i \geq 0$, $(\mathcal{P}, \langle p_i, \omega_i\rangle) \models_\lambda \psi_2$,
2. or there exists $i \geq 0$ such that $(\mathcal{P}, \langle p_i, \omega_i\rangle) \models_\lambda \psi_1$ and for every $0 \leq j \leq i$, $(\mathcal{P}, \langle p_i, \omega_i\rangle) \models_\lambda \psi_2$

First consider Item 2, it can be proved that $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p, \psi], \omega\rangle$ by applying the induction on $i - j$ as in the case of $\psi = E[\psi_1 U \psi_2]$.

Let's consider the Item 1), we will show that $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p, \psi], \omega\rangle$. According to the semantic of CTL, $\mathcal{P}$ has an infinite path $r =$

$\langle p_0, \omega_0\rangle, \langle p_1\omega_1\rangle, \langle p_2, \omega_2\rangle, ..., \langle p_i, \omega_i\rangle, ...$ such that $\langle p_i, \omega_i\rangle \models_\lambda \psi_2$. Since the number of control locations and stack alphabet of $\mathcal{P}$ is finite and the path r is infinite, there exists a configuration $\langle p_m, \gamma u\rangle$ such that $\omega_m = \gamma u$, $\langle p_0, \omega_0\rangle \Longrightarrow_{\mathcal{P}} \langle p_m, \gamma u\rangle$ and $\langle p_m, \gamma\rangle \Longrightarrow_{\mathcal{P}} \langle p_m, \gamma v\rangle$ from the proposition 3.1 of [BEM97]. This implies that $\langle p_m, \gamma u\rangle \Longrightarrow_{\mathcal{P}} \langle p_m, \gamma v u\rangle$. Let $\langle p_n, \omega_n\rangle$ be the first configuration such that $\langle p_n, \omega_n\rangle = \langle p_m, \gamma v u\rangle$.

Since we have

$$\langle [p_k, \psi], \omega_k\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle [p_k, \psi_2], \omega_k\rangle \wedge \langle [p_{k+1}, \psi], \omega_{k+1}\rangle, \text{ for every } k \geq 0$$

each configuration $\langle p_k, \omega_k\rangle$ in $\langle p_m, \gamma u\rangle \Longrightarrow_{\mathcal{P}} \langle p_m, \gamma v u\rangle$ has $\langle [p_k, \psi], \omega_k\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle [p_k, \psi_2], \omega_k\rangle \wedge \langle [p_{k+1}, \psi], \omega_{k+1}\rangle$. According to the definition of the reachability relation $\Longrightarrow_{\mathcal{BP}'_\psi}$, we obtain that

$$\langle [p_m, \psi], \gamma u\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \textstyle\bigwedge_{j=0}^{n-m} \langle [p_{m+j}, \psi_2], \omega_{m+j}\rangle \wedge \langle [p_m, \psi], \gamma v u\rangle$$

Since $(\mathcal{P}, \langle p_i, \omega_i\rangle) \models_\lambda \psi_2$ for all $i \geq 0$, by the induction hypothesis, $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_i, \psi_2], \omega_i\rangle$.

Since for each $i \geq 0$ $[p_i, \psi]$ is an accepting control location, then $\mathcal{BP}'_\psi$ has a run from $\langle [p, \psi], \omega\rangle$ such that each path will infinitely often visit some configurations $\langle [p_i, \psi], \omega_i\rangle$ with control location $[p_i, \psi] \in F$. Thus $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p, \psi], \omega\rangle$.

Case $\psi = A[\psi_1 \tilde{U} \psi_2]$: it can be proved as for $\psi = E[\psi_1 \tilde{U} \psi_2]$.

($\Longleftarrow$)$\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi], \omega\rangle$, we show that $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda \psi$ by induction on the structure of ψ.

Case $\psi = a$: We have $\langle [p, \psi], \gamma\rangle \hookrightarrow \langle p_a, \gamma\rangle$ for every $\gamma \in \Gamma$, $\langle q_1, \gamma\rangle \hookrightarrow \langle q_2, \epsilon\rangle$ for every $q_1 \xrightarrow{\gamma}_\delta q_2$ in δ_a and $\langle f, \sharp\rangle \hookrightarrow \langle f, \sharp\rangle$ for every $f \in F_a$. Since $\mathcal{BP}'_\psi$ has an accepting from $\langle [p, \psi], \omega\rangle$, there exists a state $f \in F_a$ such that $\langle [p, a], \omega\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle p_a, \omega\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle f, \sharp\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle f, \sharp\rangle$. Thus M_a has a corresponding path: $p_a \xrightarrow{\omega}_\delta f$ which implies that $\langle p, \omega\rangle \in L(M_a)$. Thus we have $\langle p, \omega\rangle \in \lambda(a)$. We obtain that $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda \psi$.

Case $\psi = \neg a$: from the product of $\mathcal{BP}'_\psi$, we have $\langle [p, \neg a], \gamma\rangle \hookrightarrow \langle p_{\neg a}, \gamma\rangle$ for all $\gamma \in \Gamma$, $\langle q_1, \gamma\rangle \hookrightarrow \langle q_2, \epsilon\rangle$, for all $q_1 \xrightarrow{\gamma}_\delta q_2$ in $\delta_{\neg a}$ and $\langle f, \sharp\rangle \hookrightarrow \langle f, \sharp\rangle$, for every $f \in F_{\neg a}$. Since $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p, \neg a], \omega\rangle$, there exists a state $f \in F_{\neg a}$ such that

$$\langle [p, \neg a], \omega\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle p_{\neg a}, \omega\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle f, \sharp\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle f, \sharp\rangle.$$

Then $M_{\neg a}$ has a corresponding path: $p_{\neg a} \xrightarrow{\omega}_\delta f$, which implies that $\langle p, \omega\rangle \in L(M_{\neg a})$. Thus we have $\langle p, \omega\rangle \notin \lambda(a)$. We obtain that $(\mathcal{P}, \langle p, \omega\rangle) \models_\lambda \psi$.

Case $\psi = \psi_1 \wedge \psi_2$: $\mathcal{BP}'_\psi$ has $\langle [p, \psi], \omega\rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle [p, \psi_1], \omega\rangle \wedge \langle [p, \psi_2], \omega\rangle$. So $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi_1], \omega\rangle$ and $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi_2], \omega\rangle$.

By applying the induction hypothesis: we get $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi_1$ and $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi_2$. Thus, we get that $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi$.

Case $\psi = \psi_1 \vee \psi_2$: $\mathcal{BP}'_\psi$ has $\langle [p, \psi], \omega \rangle \Longrightarrow_{\mathcal{BP}'_\psi} \langle [p, \psi_1], \omega \rangle \vee \langle [p, \psi_2], \omega \rangle$. So $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi_1], \omega \rangle$ or $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p, \psi_2], \omega \rangle$.

By applying the induction hypothesis: we get $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi_1$ or $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi_2$. This implies that $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi$.

Case $\psi = EX\psi_1$ is similar to case $\psi = AX\psi_1$.

Case $\psi = AX\psi_1$: Let the immediate successors $\{\langle [p_1, \psi_1], \omega_1 \rangle, ..., \langle [p_n, \psi_1], \omega_n \rangle\}$ of $\langle [p, \psi], \omega \rangle$ be the children of $\langle [p, \psi], \omega \rangle$ in the accepting run. Then $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_i, \psi_1], \omega_i \rangle$, for each $1 \leq i \leq n$. By the induction hypothesis: we get that $(\mathcal{P}, \langle p_i, \omega_i \rangle) \models_\lambda \psi_1$, for each $1 \leq i \leq n$.

The immediate successors of $\langle p, \omega \rangle$ are $\langle p_i, \omega_i \rangle$ for all $1 \leq i \leq n$. Thus, we obtain that $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi$.

Case $\psi = E[\psi_1 U \psi_2]$: Let ρ be the accepting run from $\langle [p, \psi], \omega \rangle$. Each configuration $\langle [p_i, \psi], \omega_i \rangle$ in ρ have at most two children $\langle [p_i, \psi_1], \omega_i \rangle$ and $\langle [p_{i+1}, \psi], \omega_{i+1} \rangle$ or has only one child $\langle [p_i, \psi_2], \omega_i \rangle$.

Since ρ is an accepting run, there exists a configuration $\langle [p_n, \psi], \omega_n \rangle$ in ρ such that $\langle [p_n, \psi], \omega_n \rangle$ has only one child $\langle [p_n, \psi_2], \omega_n \rangle$. Let $\langle [p_0, \psi], \omega_0 \rangle, ..., \langle [p_n, \psi], \omega_n \rangle, ...$ be a path of ρ, then $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_i, \psi_1], \omega_i \rangle$ for each $1 \leq i \leq n$, and $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_n, \psi_2], \omega_n \rangle$.

By the induction hypothesis: $(\mathcal{P}, \langle p_n, \omega_n \rangle) \models_\lambda \psi_2$ and $(\mathcal{P}, \langle p_i, \omega_i \rangle) \models_\lambda \psi_1$, for each $1 \leq i < n$.

Since $\langle p, \omega \rangle \Longrightarrow_{\mathcal{P}} \langle p_i, \omega_i \rangle \Longrightarrow_{\mathcal{P}} \langle p_n, \omega_n \rangle$, we get that $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi$.

Case $\psi = A[\psi_1 U \psi_2]$: This case is similar to $\psi = E[\psi_1 U \psi_2]$.

Case $\psi = E[\psi_1 \tilde{U} \psi_2]$: Let ρ be an accepting run from $\langle [p, \psi], \omega \rangle$, then each configuration $\langle [p_i, \psi], \omega_i \rangle$ in ρ has two children 1) $\langle [p_i, \psi_2], \omega_i \rangle$ and $\langle [p_{i+1}, \psi], \omega_{i+1} \rangle$, or 2) $\langle [p_i, \psi_1], \omega_i \rangle$ and $\langle [p_i, \psi_2], \omega_i \rangle$.

- First we consider 1). Since $[p_i, \psi]$ is an accepting control location, then every configuration $\langle [p_i, \psi], \omega_i \rangle$ in ρ has two children $\langle [p_i, \psi_2], \omega_i \rangle$ and $\langle [p_{i+1}, \psi], \omega_i \rangle$. Then $\mathcal{BP}'_\psi$ has an infinite path $\langle [p_0, \psi], \omega_0 \rangle, ..., \langle [p_{i+1}, \psi], \omega_{i+1} \rangle, ...,$ in the accepting run where $\langle [p_0, \psi], \omega_0 \rangle = \langle [p, \psi], \omega \rangle$, and $\mathcal{BP}'_\psi$ has an accepting run from the configuration $\langle [p_i, \psi_2], \omega_i \rangle$ for every $i \geq 0$. By applying the induction hypothesis, we get $(\mathcal{P}, \langle p_i, \omega_i \rangle) \models_\lambda \psi_2$ for every $i \geq 0$. Thus, we get $\langle p, \omega \rangle \Longrightarrow_{\mathcal{P}} \langle p_i, \omega_i \rangle$ and $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi$.
- Let's consider 2). There exists a configuration $\langle [p_n, \psi], \omega_n \rangle$ in ρ whose children are $\langle [p_n, \psi_1], \omega_n \rangle$ and $\langle [p_n, \psi_2], \omega_n \rangle$. Then $\mathcal{BP}'_\psi$ has an infinite path $\langle [p_0, \psi], \omega_0 \rangle, ..., \langle [p_n, \psi], \omega_n \rangle, \langle [p_n, \psi_1], \omega_n \rangle ...,$ in the accepting run where $\langle [p_0, \psi], \omega_0 \rangle = \langle [p, \psi], \omega \rangle$. Each configuration $\langle [p_i, \psi], \omega_i \rangle$ has children $\langle [p_i, \psi_2], \omega_i \rangle$

and $\langle [p_{i+1}, \psi], \omega_i \rangle$. Thus $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_n, \psi_1], \omega_n \rangle$ and $\mathcal{BP}'_\psi$ has an accepting run from $\langle [p_i, \psi_2], \omega_i \rangle$, for $1 \leq i \leq n$.
By the induction hypothesis: $(\mathcal{P}, \langle p_n, \omega_n \rangle) \models_\lambda \psi_1$ and $(\mathcal{P}, \langle p_i, \omega_i \rangle) \models_\lambda \psi_2$, for each $1 \leq i \leq n$. Thus we have $\langle p, \omega \rangle \Longrightarrow_{\mathcal{P}} \langle p_i, \omega_i \rangle \Longrightarrow_{\mathcal{P}} \langle p_n, \omega_n \rangle$ and $(\mathcal{P}, \langle p, \omega \rangle) \models_\lambda \psi$.

Case $\psi = A[\psi_1 \tilde{U} \psi_2]$: This case is similar to case $\psi = E[\psi_1 \tilde{U} \psi_2]$. □