A Grammatical Representation of Visibly Pushdown Languages

Baran, Joachim and Barringer, Howard

2007

MIMS EPrint: **2010.57**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

A Grammatical Representation of Visibly Pushdown Languages

Joachim Baran and Howard Barringer

The University of Manchester, School of Computer Science, Manchester, UK
joachim.baran@cs.manchester.ac.uk, howard.barringer@cs.manchester.ac.uk

Abstract. Model-checking regular properties is well established and a powerful verification technique for regular as well as context-free program behaviours. Recently, through the use of ω-visibly pushdown languages (ωVPLs), defined by ω-visibly pushdown automata, model-checking of properties beyond regular expressiveness was made possible and shown to be still decidable even when the program's model of behaviour is an ωVPL. In this paper, we give a grammatical representation of ωVPLs and the corresponding finite word languages – VPL. From a specification viewpoint, the grammatical representation provides a more natural representation than the automata approach.

1 Introduction

In [AM04], ω-visibly pushdown languages over infinite words (ωVPLs) were introduced as a specialisation of ω-context-free languages (ωCFLs), i.e. they are strictly included in the ωCFLs but more expressive than ω-regular languages (ωRLs). The paper showed that the language inclusion problem is decidable for ωVPLs, and thus, the related model-checking problem is decidable as well. This work was presented in the context of (ω)VPLs[1] being represented as automata and a monadic second-order logic with matching relation.

In this paper, we define a grammatical representation of (ω)VPLs. We also propose that grammars allow us to write more natural specifications than (ω)-visibly pushdown automata ((ω)VPA). Section 2 introduces the formalisms used in this paper. In Section 3 our grammatical representation is presented. Finally, Section 4 concludes our work.

2 Preliminaries

For an arbitrary set X, we write 2^X to denote its power-set. Let Σ denote a finite alphabet over letters $a, b, c, \ldots$, the set of finite (infinite) words over Σ is denoted by Σ^* (Σ^ω). We use ε to denote the empty word. For an arbitrary word $w \in \Sigma^*$ we will write $|w|$ to denote its length. For the empty word ε, we set

[1] Our bracketing of ω abbreviates restating the sentence without the bracketed contents.

D. Leivant and R. de Queiroz (Eds.): WoLLIC 2007, LNCS 4576, pp. 1–11, 2007.

$|\varepsilon| = 0$. The concatenation of two words w and w' is denoted by $w \cdot w'$. The length of an infinite word equals the first infinite ordinal ω. Positions in a word w are addressed by natural numbers, where the first index starts at 1. The i-th letter of a word is referred to as $w(i)$. We use a sans-serif font for meta-variables and a (meta)-variable's context is only explicitly stated once.

2.1 Visibly Pushdown Languages

(ω)VPLs are defined over a terminal alphabet of three pairwise disjoint sets Σ_c, Σ_i and Σ_r, which we will use as properties in specifications to denote calls, internal actions and returns respectively. Any call may be matched with a subsequent return, while internal actions must not be matched at all. A formalisation of (ω)VPLs has been given in terms of automata as well as in terms of logic.

Visibly Pushdown Automata. For (ω)VPA, the current input letter determines the actions the automaton can perform.

Definition 1. *A* visibly pushdown automaton over finite words *(*VPA*) (*visibly pushdown automaton over infinite words *(*ωVPA*)) is a sextuple* $A = (Q, \Sigma_c \cup \Sigma_i \cup \Sigma_r, \Gamma, \delta, Q', F)$*, where* Q *is a finite set of states* $\{p, q, q_0, q_1, \ldots\}$*,* $\Sigma_c, \Sigma_i, \Sigma_r$ *are finite sets of terminals representing calls* $c, c_0, c_1, \ldots, c_k$*, internal actions* $i, i_0, i_1, \ldots, i_l$*, and returns* $r, r_0, r_1, \ldots, r_m$ *respectively,* Γ *is a finite set of stack symbols* $A, B, C, \ldots$*, including the stack bottom marker* $\bot$*,* δ *is a finite set of transition rules between states* $\mathsf{p}, \mathsf{q} \in Q$ *for inputs* $\mathsf{c} \in \Sigma_c$*,* $\mathsf{i} \in \Sigma_i$*, or* $\mathsf{r} \in \Sigma_r$ *and stack contents* $\mathsf{A}, \mathsf{B} \in (\Gamma \setminus \{\bot\})$ *of the form* $\mathsf{p} \xrightarrow{\mathsf{c},\kappa/\mathsf{B}\kappa} \mathsf{q}$ *for all* $\kappa \in \Gamma$*,* $\mathsf{p} \xrightarrow{\mathsf{i},\kappa/\kappa} \mathsf{q}$ *for all* $\kappa \in \Gamma$*,* $\mathsf{p} \xrightarrow{\mathsf{r},\bot/\bot} \mathsf{q}$*, or* $\mathsf{p} \xrightarrow{\mathsf{r},\mathsf{A}/\varepsilon} \mathsf{q}$*,* $Q' \subseteq Q$ *denotes a non-empty set of designated initial states,* $F \subseteq Q$ *is the set of final states.*

When reading a word w, instantaneous descriptions (q, w, α) are used to describe the current state, the current stack contents and a postfix of w that still has to be processed. The binary move relation $\vdash_A$ determines possible moves an (ω)VPA A can make. Whenever A in $\vdash_A$ is understood from the context, we write $\vdash$. In the following we use $\vdash^*$ ($\vdash^\omega$) in order to denote a finitely (infinitely) repeated application of $\vdash$ (up to the first infinite ordinal). In conjunction with $\vdash^\omega$, we use Q_{inf} to denote the set of states that appear infinitely often in the resulting sequence.

Definition 2. *The* language $L(A)$ *of a* (ω)VPA $A = (Q, \Sigma_c \cup \Sigma_i \cup \Sigma_r, \Gamma, \delta, Q', F)$ *is the set of finite (infinite) words that are derivable from any initial state in* Q'*, i.e.* $L(A) = \{w \mid (\mathsf{p}, w, \bot) \vdash^* (\mathsf{q}, \varepsilon, \gamma)$ *and* $\mathsf{p} \in Q'$ *and* $\mathsf{q} \in F\}$ *(*$L(A) = \{w \mid (\mathsf{p}, w, \bot) \vdash^\omega (\mathsf{q}, \varepsilon, \gamma)$ *and* $\mathsf{p} \in Q'$ *and* $Q_{inf} \cap F \neq \emptyset\}$*).*

In an arbitrary word of the ωVPLs, calls and returns can appear either matched or unmatched. A call automatically matches the next following return, which is not matched by a succeeding call. A call is said to be unmatched, when there are less or equally many returns than calls following it. Unmatched calls cannot

be followed by unmatched returns, but unmatched returns may be followed by unmatched calls.

A word w of the form $\mathsf{c}\alpha\mathsf{r}$ is called *minimally well-matched*, iff c and r are a matching and α contains no unmatched calls or returns. The set of all minimally well-matched words is denoted by L_{mwm} ([LMS04], p. 412, par. 8). In conjunction with a given ωVPA A, a *summary-edge* is a triple $(\mathsf{p},\mathsf{q},\mathsf{f})$, $\mathsf{f} \in \{0,1\}$, which abstracts minimally well-matched words that are recognised by A when going from p to q, where on the corresponding run a final state has to be visited ($\mathsf{f} = 1$) or not ($\mathsf{f} = 0$). The set of words represented by a summary edge is denoted by $L((\mathsf{p},\mathsf{q},\mathsf{f}))$.

Definition 3. *A* pseudo-run *of an* ωVPA $A = (Q, \Sigma_c \cup \Sigma_i \cup \Sigma_r, \Gamma, \delta, Q', F)$ *is an infinite word* $w = \alpha_1\alpha_2\alpha_3\ldots$ *with* $\alpha_i \in (\Sigma_c \cup \Sigma_i \cup \Sigma_r \cup \bigcup_{n=1}^{m}\{\Omega_n\})$, *each* Ω_n *denotes a non-empty set of* summary-edges *of the form* $(\mathsf{p},\mathsf{q},\mathsf{f})$ *with* $\mathsf{f} \in \{0,1\}$, *in case* $\alpha_i = \mathsf{c}$, *then there is no* $\alpha_j = \mathsf{r}$ *for* $i < j$, *and there is a word* $w' = \beta_1\beta_2\beta_3\ldots$, $w' \in L(A)$, *so that either* $\alpha_i = \beta_i$, *or* $\alpha_i = \Omega_k$ *and* β_i *is a minimally well-matched word that is generated due to* A *moving from state* p *to* q *and* $(\mathsf{p},\mathsf{q},\mathsf{f}) \in \Omega_k$. *In case* $\mathsf{f} = 1$ *(*$\mathsf{f} = 0$*), then a final state is (not) on the path from* p *to* q.

According to [AM04], p. 210, par. 6, a non-deterministic Büchi-automaton can be constructed that accepts all pseudo-runs of an arbitrary given ωVPA. For every pseudo-run that is represented by the Büchi-automaton, there exists a corresponding accepting run of the original ωVPA.

Monadic Second-Order Logic with Matched Calls/Returns. A logical representation, (ω)MSO$_\mu$, of (ω)VPLs was given as an extension of monadic second-order logic (MSO) with a matching relation μ, which matches calls and returns, where the call always has to appear first.

Definition 4. *A formula* φ *is a formula of* monadic second-order logic of one successor with call/return matching relation *((ω)*MSO$_\mu$*) over an alphabet* $\Sigma_c \cup \Sigma_i \cup \Sigma_r$, *iff it is of the form* $\varphi \equiv \top$, $\varphi \equiv T_{\mathsf{a}}(\mathsf{i})$, $\mathsf{a} \in \Sigma$, $\varphi \equiv \mathsf{i} \in \mathsf{X}$, $\varphi \equiv \mathsf{i} \leq \mathsf{j}$, $\varphi \equiv \mu(\mathsf{i},\mathsf{j})$, $\varphi \equiv S(\mathsf{i},\mathsf{j})$, $\varphi \equiv \neg\psi$, $\varphi \equiv \psi_1 \vee \psi_2$, $\varphi \equiv \exists\mathsf{i}\ \psi(\mathsf{i})$, *or* $\varphi \equiv \exists\mathsf{X}\psi(\mathsf{X})$, *where* ψ, ψ_1, *and* ψ_2 *are (ω)*MSO$_\mu$ *formula as well,* V *and* W *are sets of first-order and second-order variables respectively, and* $\mathsf{i},\mathsf{j} \in V$, $\mathsf{X} \in W$.

We use the standard abbreviations for the truth constant, conjunction and universal quantification. Also, $\forall\mathsf{x}(\mathsf{y} \leq \mathsf{x} \leq \mathsf{z} \Rightarrow \varphi)$ is shortened to $\forall\mathsf{x}{\in}[\mathsf{y},\mathsf{z}]\,\varphi$. In order to simplify arithmetic in conjunction with the successor function, we will omit the successor function completely in the following and write $\mathsf{i}+1$ instead of j for which $S(\mathsf{i},\mathsf{j})$ holds. Second-order quantifications $\exists X_1\exists X_2\ldots\exists X_k$ are abbreviated in vector notation as $\exists\boldsymbol{X}$.

We assume the usual semantics for (ω)MSO$_\mu$ formulae, where $\mu(\mathsf{i},\mathsf{j})$ is true when $w(\mathsf{i}) = \mathsf{c}$ and $w(\mathsf{j}) = \mathsf{r}$ are a matching call/return pair.

Definition 5. *The* language $\mathcal{L}(\varphi)$ *of an (ω)*MSO$_\mu$ *formula* φ *is the set of finite (infinite) words* w *for which there is a corresponding model of* φ.

2.2 Context-Free Grammars

Definition 6. *An* (ω)-context-free grammar *((*ω)CFG*)* G over finite words (infinite words) *is a quadruple* (V, Σ, P, S) *(quintuple* (V, Σ, P, S, F)*), where* V *is a finite set of non-terminals* $A, B, \ldots$*,* Σ *is a finite set of terminals* $a, b, \ldots$*,* V *and* Σ *are disjoint,* P *is a finite set of productions of the form* $V \times (V \cup \Sigma)^*$*, and* S *denotes a designated starting non-terminal* $S \in V$ *(and* $F \subseteq V$ *denotes the set of accepting non-terminals).*

We will use the notation $A \rightarrow_G \alpha$ for a production (A, α) in G. If G is understood from the context, we write $A \rightarrow \alpha$. We also use $\rightarrow_G$ to denote the derivation relation of G, that determines derivations of sentential forms of G. Again, we drop the sub-script when G is understood from the context. In the following we write $\overset{*}{\rightarrow}$ in order to denote a finitely repeated application of $\rightarrow$ while $\overset{\omega}{\rightarrow}$ denotes an infinite application of $\rightarrow$. Similarly to the previously used set Q_{inf}, we use V_{inf} in connection with $\overset{\omega}{\rightarrow}$ in order to denote the set of non-terminals that are infinitely often replaced among the sentential forms.

Definition 7. *The* language $\mathcal{L}(G)$ *of an* (ω)CFG $G = (V, \Sigma, P, S)$ *(*$G = (V, \Sigma, P, S, F)$*) is the set of finite (infinite) words over* Σ *that are derivable from the initial symbol, i.e.* $\mathcal{L}(G) = \{w \mid S \overset{*}{\rightarrow} w \text{ and } w \in \Sigma^*\}$ *(*$\mathcal{L}(G) = \{w \mid S \overset{\omega}{\rightarrow} w, w \in \Sigma^\omega \text{ and } V_{inf} \cap F \neq \emptyset\}$*).*

2.3 Balanced Grammars

Balanced grammars are a specialisation of context-free grammars over finite words [BB02]. Unlike the previous definition of CFGs, balanced grammars are permitted to have an infinite set of productions. This is due to regular expressions over terminals and/or non-terminals in right-hand sides of productions.

Definition 8. *A* balanced grammar *(*BG*)* G over finite words *is a quadruple* $(V, \underline{\Sigma} \cup \Sigma \cup \overline{\Sigma}, \mathcal{P}, S)$ *is a specialisation of a context-free grammar, where* $\underline{\Sigma}$ *and* $\overline{\Sigma}$ *are finite sets of terminals* $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_k$ *and co-terminals* $\overline{a}_1, \overline{a}_2, \ldots, \overline{a}_k$ *respectively, where each terminal* $\underline{a}_i$ *is associated with its unique counterpart,* $\overline{a}_i$*, its co-terminal, and vice versa,* Σ *is a finite set of intermediate terminals* $a, b, \ldots$*, the sets* $\underline{\Sigma}$*,* $\overline{\Sigma}$*, and* Σ *are mutually disjoint,* $\mathcal{P}$ *is a finite or infinite set of productions of the form* $V \times \underline{\mathsf{a}}(V \cup \Sigma)^*\overline{\mathsf{a}}$*, and* S *denotes a designated starting non-terminal* $S \in V$*.*

As already pointed out in [BB02], an infinite set of productions does not raise the grammars' expressiveness, but provides a succinct notation. The derivation relation of context-free grammars is still applicable to balanced grammars.

Definition 9. *The* language $\mathcal{L}(G)$ *of a* BG $G = (V, \underline{\Sigma} \cup \Sigma \cup \overline{\Sigma}, \mathcal{P}, S)$ *is the set of words that are derivable from the initial symbol, i.e.* $\mathcal{L}(G) = \{w \mid S \overset{*}{\rightarrow} w \text{ and } w \in (\underline{\Sigma} \cup \Sigma \cup \overline{\Sigma})^*\}$*.*

In the following, we are writing $\mathcal{R}$ to denote an arbitrary regular expression over $V \cup \Sigma$.

3 Grammars for Visibly Pushdown Languages

A grammatical representation of (ω)VPLs is presented, where we take a compositional approach that builds on pseudo-runs and minimally well-matched words. We first state our grammatical representation and then decompose it into two types of grammars. We show their resemblance of pseudo-runs and minimally well-matched words, similar to the approach for (ω)VPAs.

3.1 Quasi Balanced Grammars

In order to simplify our proofs, we give an alternative – but expressively equivalent – definition of BGs, where only a finite number of productions is admitted. We reformulate occurrences of regular expressions $\mathcal{R}$ in terms of production rules $P_\mathcal{R}$ and substitute each $\mathcal{R}$ by an initial non-terminal $S_\mathcal{R}$ that appears on a left-hand side in $P_\mathcal{R}$. Therefore, matchings $\underline{a}\mathcal{R}\overline{a}$ become $\underline{a}S_\mathcal{R}\overline{a}$, where the derivation of $S_\mathcal{R}$ resembles $L(\mathcal{R})$.

Definition 10. *Let $G = (V, \underline{\Sigma} \cup \Sigma \cup \overline{\Sigma}, \mathcal{P}, S)$ denote an arbitrary* BG*, a* quasi balanced grammar *(*qBG*) $G' = (V', \underline{\Sigma} \cup \Sigma \cup \overline{\Sigma}, P, S)$ generalises G by having a finite set of productions, where productions are either*

a) *in double Greibach normal form $\mathsf{A} \to \underline{\mathsf{a}} S_\mathcal{R} \overline{\mathsf{a}}$, or*
b) *of form $\mathsf{A} \to \mathsf{BC}$, $\mathsf{A} \to \mathsf{aC}$, or $\mathsf{A} \to \varepsilon$, where B's productions are of the form according to* a) *and C's productions are of the form according to* b)*.*

Lemma 1. *For every* BG *$G = (V, \underline{\Sigma} \cup \Sigma \cup \overline{\Sigma}, \mathcal{P}, S)$ there is a* qBG *$G' = (V', \underline{\Sigma} \cup \Sigma \cup \overline{\Sigma}, P, S)$, such that $L(G) = L(G')$.*

3.2 A Grammatical Representation of ωVPLs

Matchings in an ωVPL appear only as finite sub-words in the language, which was utilised in the characterisation of pseudo-runs. Summary-edges reflect sub-words of exactly this form, which are in L_{mwm}. Given an infinite word w, it can be split into sub-words that are either in L_{mwm} or in $\Sigma_c \cup \Sigma_i \cup \Sigma_r$, where no sub-word in Σ_r follows a sub-word in Σ_c. We abbreviate the latter constraint as Σ_c/Σ_r*-matching avoiding*. Our grammatical representation of ωVPLs utilises Σ_c/Σ_r-matching avoiding ωRGs to describe languages of pseudo-runs. Languages of summary-edges, i.e. languages with words in L_{mwm}, are separately described by qBGs under a special homomorphism. The homomorphism is required to cover matchings of calls c that can match more than one return r, which cannot be reflected as a simple terminal/co-terminal matching $\underline{\mathsf{a}}/\overline{\mathsf{a}}$. For example, the matchings c/r_1 and c/r_2 are representable as terminal/co-terminal pairs $\underline{a}/\overline{a}$ and $\underline{b}/\overline{b}$ under the mappings $h(\underline{a}) = h(\underline{b}) = c$, $h(\overline{a}) = r_1$ and $h(\overline{b}) = r_2$. Finally, the amalgamation of Σ_c/Σ_r-matching avoiding ωRGs and qBGs under the aforementioned homomorphism give us a grammatical representation of ωVPLs:

Definition 11. *A* superficial[2] *ω-regular grammar with injected balanced grammars (ωRG(qBG)+h) $G = (V, \Sigma_c \cup \Sigma_i \cup \Sigma_r \cup \bigcup_{n=1}^{m}\{g_n\}, P, S, F, \bigcup_{n=1}^{m}\{G_n\}, h)$, where*

- *Σ_c, Σ_i, Σ_r and $\bigcup_{n=1}^{m}\{g_n\}$ are mutually disjoint,*
- *G is Σ_c/Σ_r-matching avoiding,*
- *$G_n = (V_n, \Sigma_n, P_n, S_n)$ is a* qBG *for $n = 1, 2, \ldots, m$,*[3]

*is an ω*CFG *$G' = (V \cup \bigcup_{n=1}^{m}\{V_n\}, \Sigma \cup \bigcup_{n=1}^{m}\{\Sigma_n\}, P', S, F)$ with*

- *disjoint sets V and $\{V_1, V_2, \ldots, V_m\}$ as well as Σ and $\{\Sigma_1, \Sigma_2, \ldots, \Sigma_m\}$, and*
- *P' is the smallest set satisfying*
  - *$\mathsf{A} \rightarrow_{G'} \mathsf{aB}$ if $\mathsf{A} \rightarrow_G \mathsf{aB}$, where $\mathsf{a} \in (\Sigma_c \cup \Sigma_i \cup \Sigma_r)$, or*
  - *$\mathsf{A} \rightarrow_{G'} S_n\mathsf{B}$ if $\mathsf{A} \rightarrow_G g_n\mathsf{B}$, or*
  - *$\mathsf{A} \rightarrow_{G'} \alpha$ if $\mathsf{A} \rightarrow_{G_n} \alpha$,*

and h is constrained so that it preserves terminals of the injector grammar, $h(\mathsf{a}) = \mathsf{a}$ for any $\mathsf{a} \in (\Sigma_c \cup \Sigma_i \cup \Sigma_r)$, and for terminals/co-terminals of injected grammars it maps terminals $\underline{\mathsf{a}} \in \underline{\Sigma}_n$ to calls $\mathsf{c} \in \Sigma_c$, maps co-terminals $\overline{\mathsf{a}} \in \overline{\Sigma}_n$ to returns $\mathsf{r} \in \Sigma_r$, maps terminals $\mathsf{a} \in \Sigma_n$ to internal actions $\mathsf{i} \in \Sigma_i$.

In the following, we refer to the homomorphism h under the constraints which are given above as *superficial mapping h*.

Definition 12. *The* language *$\mathcal{L}(G)$ of an ω*RG(qBG)+h *$G = (V, \Sigma_c \cup \Sigma_i \cup \Sigma_r \cup \bigcup_{n=1}^{m}\{g_n\}, P, S, F, \bigcup_{n=1}^{m}\{G_n\}, h)$ denotes the set $\{h(w) \mid S \xrightarrow{\omega}_{G'} w$ and $w \in (\Sigma \cup \Sigma_1 \cup \Sigma_2 \cup \ldots \cup \Sigma_m)^{\omega}\}$, where G' is the ω-context-free grammar corresponding to G.*

Consider an arbitrary ωRG(qBG)+h $G = (V, \Sigma_c \cup \Sigma_i \cup \Sigma_r \cup \bigcup_{n=1}^{m}\{g_n\}, P, S, F, \bigcup_{n=1}^{m}\{G_n\}, h)$. We call the ωRG $G_{\uparrow} = (V, \Sigma_c \cup \Sigma_i \cup \Sigma_r \cup \bigcup_{n=1}^{m}\{g_n\}, P, S, F)$ the *injector grammar* of G, while the qBGs G_1, G_2, $\ldots$ G_m are called *injected grammars* of G.[4] When G is clear from the context, we just talk about the injector grammar $G_{\uparrow}$ and the injected grammars G_1, $\ldots$, G_m respectively. The languages associated with these grammars are referred to as injector and injected languages respectively. In fact, injector languages resemble pseudo-runs with pseudo edges g_n, $n = 1 \ldots m$, while injected language resemble matchings covered by summary-edges.

3.3 ωVPL and ωRL(qBL)+h Coincide

For the equivalence proof of ωVPLs and ωRL(qBL)+hs, we first show that minimally well-matched words, as described by summary-edges, can be expressed by

[2] Superficial – as understood as being on the surface of something.

[3] Each Σ_n is a shorthand for $\underline{\Sigma}_n \cup \Sigma_n \cup \overline{\Sigma}_n$.

[4] This should not be confused with nested words, [AM06], which describe the structure induced by matchings in finite words.

qBGs plus an appropriate superficial mapping, and vice versa. It is then straightforward to prove language equivalence, by translating an arbitrary ωRG(qBG)+h into an expressively equivalent ωMSO formula and an arbitrary ωVPA into an expressively equivalent ωRG(qBG)+h.

Let $\mathrm{VPA}_{\mathrm{mwm}}$ and $\mathrm{MSO}_{\mathrm{mwm}}$ refer to VPA and MSO-formulae whose languages are subsets of L_{mwm} respectively, i.e. restricted variants of VPA and MSO-formulae that only accept minimally well-matched words. We show that any qBG can be translated into an equivalent $\mathrm{MSO}_{\mathrm{mwm}}$ formula. Since $\mathrm{MSO}_{\mathrm{mwm}}$ defines L_{mwm}, the inclusion qBL $\subseteq L_{\mathrm{mwm}}$ is proven. Second, for an arbitrary $\mathrm{VPA}_{\mathrm{mwm}}$ a qBG is constructed so that their languages coincide, which gives us qBL $\supseteq \mathrm{VPL}_{\mathrm{mwm}}$.

In the following lemma, an $\mathrm{MSO}_{\mathrm{mwm}}$ formula is constructed from a qBG in such a way so that their languages coincide. The translation works by quantifying over each matching in the word and filling in the respective regular expressions. In order for the lemma to hold, we assume that all of the qBG's productions are uniquely identified by their terminal/co-terminal pairs. While this clearly restricts a grammar's language in general, the language under a superficial mapping is preserved.

Lemma 2. *Let $G = (V, \underline{\Sigma} \cup \Sigma \cup \overline{\Sigma}, P, S)$ denote an arbitrary* qBG *and let h be an arbitrary superficial mapping. Then the* $\mathrm{MSO}_{\mathrm{mwm}}$ *formula*

$$\varphi \equiv \exists \boldsymbol{X} \exists i \bigvee_{(S \to \underline{\mathsf{a}} S_{\mathcal{R}} \overline{\mathsf{a}}) \in P} (\Psi_{\underline{\mathsf{a}},\overline{\mathsf{a}}}(1,i) \wedge T_{\$}(i+1) \wedge \forall k \in [1,i] \Phi(k))$$

accepts the same language as G, where

$$\Psi_{\underline{\mathsf{a}},\overline{\mathsf{a}}}(\mathsf{i},\mathsf{j}) \equiv T_{\underline{\mathsf{a}}}(\mathsf{i}) \wedge T_{\overline{\mathsf{a}}}(\mathsf{j}) \wedge \mu(\mathsf{i},\mathsf{j}),$$

$$\Phi(\mathsf{k}) \equiv \bigvee_{\underline{\mathsf{a}} \in \underline{\Sigma}} T_{\underline{\mathsf{a}}}(\mathsf{k}) \Rightarrow \exists j (\mu(\mathsf{k},j) \wedge \varphi_{\underline{\mathsf{a}}}(\mathsf{k}+1,j)),$$

$$\begin{aligned} \Delta(\mathsf{s},\mathsf{t},\mathsf{k}) \equiv {} & \textstyle\bigvee_{(\mathsf{A} \to \mathsf{aB}) \in P_{\mathcal{R}}} (X_{\mathsf{A}}(\mathsf{k}) \wedge X_{\mathsf{B}}(\mathsf{k}+1) \wedge T_{\mathsf{a}}(\mathsf{k})) \vee \\ & \textstyle\bigvee_{(\mathsf{A} \to \mathsf{BC},\, \mathsf{B} \to \underline{\mathsf{b}} S_{\mathcal{R}'} \overline{\mathsf{b}}) \in P_{\mathcal{R}}} \exists j \in [\mathsf{s},\mathsf{t}] (X_{\mathsf{A}}(\mathsf{k}) \wedge X_{\mathsf{C}}(j+1) \wedge \Psi_{\underline{\mathsf{b}},\overline{\mathsf{b}}}(\mathsf{k},j)) \end{aligned}$$

$$\begin{aligned} \varphi_{\underline{\mathsf{a}}}(\mathsf{s},\mathsf{t}) \equiv {} & X_{S_{\mathcal{R}}}(s) \wedge \textstyle\bigwedge_{(\mathsf{A},\,\mathsf{B}) \,\in\, V,\, \mathsf{A} \neq \mathsf{B}} \forall k\ \neg (X_{\mathsf{A}}(k) \wedge X_{\mathsf{B}}(k)) \wedge \\ & \forall k \in [\mathsf{s},\mathsf{t}-1]\, \varphi_{\overline{\mu}}(\mathsf{s},\mathsf{t},k) \Rightarrow \Delta(\mathsf{s},\mathsf{t},k) \wedge \textstyle\bigvee_{(\mathsf{A} \to \varepsilon) \in P_{\mathcal{R}}} X_{\mathsf{A}}(\mathsf{t}), \end{aligned}$$

where $(\mathsf{A} \to \underline{\mathsf{a}} S_{\mathcal{R}} \overline{\mathsf{a}}) \in P$, *and* $\varphi_{\overline{\mu}}(\mathsf{s},\mathsf{t},\mathsf{k}) \equiv \neg \exists i,j \in [\mathsf{s},\mathsf{t}] (\mu(i,j) \wedge i < \mathsf{k} \leq j)$.

Proof. Consider an arbitrary qBG+h G and its translation to a $\mathrm{MSO}_{\mathrm{mwm}}$ formula φ. We show that every word $w\$ \in L(\varphi)$ is a word $w \in L(G)$ and vice versa.

$L(\varphi) \subseteq L(G)$*:* Let $\mathcal{M}$ be an arbitrary model of φ that represents the word w. We write $\langle T_1, X_1 \rangle \langle T_2, X_2 \rangle \ldots \langle T_{|w|}, X_{|w|} \rangle \langle T_{\$}, X_{|w|+1} \rangle$ to denote the sequence of unique predicate pairs of T and X which hold at indices 1 to $|w|+1$ in $\mathcal{M}$.

Occurrences of the form $\langle T_{\overline{\mathsf{a}}}, X_{\mathsf{B}} \rangle$ are replaced by $\langle T_{\overline{\mathsf{a}}}, \mathsf{B} \rangle$ if $(\mathsf{B} \to \varepsilon) \in P$, occurrences of the form $\langle T_{\mathsf{a}}, X_{\mathsf{A}} \rangle \langle T_{\overline{\mathsf{a}}}, \mathsf{B} \rangle$ are replaced by $\langle T_{\overline{\mathsf{a}}}, \mathsf{A} \rangle$ if $(\mathsf{A} \to \mathsf{aB}) \in P$,

and occurrences of the form $\langle T_{\underline{\mathsf{a}}}, X_{\mathsf{B}}\rangle\langle T_{\overline{\mathsf{a}}}, S_{\mathcal{R}}\rangle\langle T_{\overline{\mathsf{b}}}, \mathsf{C}\rangle$ are replaced with $\langle T_{\overline{\mathsf{b}}}, \mathsf{A}\rangle$ if $(\mathsf{A} \to \mathsf{BC}, \mathsf{B} \to \underline{\mathsf{a}} S_{\mathcal{R}} \overline{\mathsf{a}}) \in P$. Eventually, $\langle T_{\underline{\mathsf{a}}}, X_S\rangle\langle T_{\overline{\mathsf{a}}}, S_{\mathcal{R}}\rangle$ will be left, which is replaced with S iff $(S \to \underline{\mathsf{a}} S_{\mathcal{R}} \overline{\mathsf{a}}) \in P$. As a result, we have established a bottom up parse in G for an arbitrary word $w\$ \in L(\varphi)$, which implies that every word in $L(\varphi)$ is in $L(G)$.

$L(G) \subseteq L(\varphi)$*:* Let $S \to \alpha \to \beta \to \ldots \to w$ denote an arbitrary derivation in G. With each derivation step, we try to find variable assignments that satisfy φ, so that after the derivation finishes, $w\$$ is represented by the sequence $\langle T_1, X_1\rangle\langle T_2, X_2\rangle \ldots \langle T_\$, X_{|w|+1}\rangle$ of φ. The construction of $\langle T_1, X_1\rangle\langle T_2, X_2\rangle \ldots \langle T_\$, X_{|w|+1}\rangle$ follows the derivation $S \to \alpha \to \beta \to \ldots \to w$ in the sense that there is a mapping between the n-th step of the sequence constructed by the variable assignments and the n-th sentential form reached in the derivation.

We consider triples of the form $(\!|\mathsf{A}, \psi, \mathsf{B}|\!)$, where A is a non-terminal as it appears in some sentential form derived from S, ψ denotes the formula which is supposed to derive a word w, where $A \xrightarrow{*} w$, and B is a temporary stored non-terminal. When A derives α with $A \to \alpha$, we try to find variable assignments for ψ that represent terminals in α. Since terminals appear only in prefixes/postfixes of α, we remove the ground terms in ψ, add pairs $\langle T_{\mathsf{k}}, X_{\mathsf{k}}\rangle$ to the left/right of $(\!|\mathsf{A}, \psi, \mathsf{B}|\!)$ accordingly, and replace $(\!|\mathsf{A}, \psi, \mathsf{B}|\!)$ with $(\!|\mathsf{C}, \psi', \mathsf{E}|\!)$ or the sequence $(\!|\mathsf{C}, \psi', \mathsf{E}|\!)(\!|\mathsf{D}, \psi'', \mathsf{F}|\!)$, depending if α has one non-terminal C or two non-terminals CD as sub-word. The non-terminals E and F are associated with productions $\mathsf{E} \to \varepsilon$ and $\mathsf{F} \to \varepsilon$ respectively, where they denote the end of a regular expression embedded between a call and return.

Starting the rewriting process with $(\!|\mathsf{S}, \varphi, \mathsf{A}|\!)$, A is chosen arbitrarily, a sequence of tuples of the form $\langle T_{\mathsf{k}}, X_{\mathsf{k}}\rangle$ is eventually left, which indeed represents w, so that the model for $w\$$ is represented by adding $\langle T_\$, X_{\mathsf{A}}\rangle$ to the sequence. □

The reverse inclusion, i.e. qBL $\supseteq$ VPL, can be shown by a number of rewriting steps of an arbitrary VPA_{mwm} to a BG equipped with a superficial mapping. Since there is a translation from BGs to qBGs, the inclusion is then proven. The VPA_{mwm} represents hereby $L((\mathsf{p}, \mathsf{q}, \mathsf{f}))$ of some summary-edge $(\mathsf{p}, \mathsf{q}, \mathsf{f})$.

Definition 13. *Let* $G = (V, \Sigma_c \cup \Sigma_i \cup \Sigma_r, P, S)$ *denote the* CFG *with productions of the form* $S \to \mathsf{cA}$, $\mathsf{A} \to \mathsf{cBC}$, $\mathsf{A} \to \mathsf{iB}$, *and* $\mathsf{A} \to \mathsf{r}$ *that is obtained from a* VPA_{mwm} *by the standard translation [HMU01, Theorem 6.14, incl. its proof], then the* immediate matching CFG $G' = (V', \Sigma_c \cup \Sigma_i \cup \Sigma_r, P', S')$ *is obtained from* G*, so that* $S' \to_{G'} \mathsf{c}\langle \mathsf{A}, \mathsf{r}\rangle\mathsf{r}$ *iff* $S \to_G \mathsf{cA}$, $\langle \mathsf{A}, \mathsf{r}_1\rangle \to_{G'} \mathsf{c}\langle \mathsf{B}, \mathsf{r}_2\rangle\mathsf{r}_2\langle \mathsf{C}, \mathsf{r}_1\rangle$ *iff* $\mathsf{A} \to_G \mathsf{cBC}$, $\langle \mathsf{A}, \mathsf{r}\rangle \to_{G'} \mathsf{i}\langle \mathsf{B}, \mathsf{r}\rangle$ *iff* $\mathsf{A} \to_G \mathsf{iB}$, $\langle \mathsf{A}, \mathsf{r}\rangle \to_{G'} \varepsilon$ *iff* $\mathsf{A} \to_G \mathsf{r}$.

Lemma 3. *The language* $L(G)$ *of an* immediate matching CFG G *that is obtained from a* VPA_{mwm} A *is equal to* $L(A)$.

Proof (Lemma 3). The translation of Definition 13 is preserving the language equivalence of the grammars, as it is a special case of the more general translation presented in [[Eng92, Page 292]. □

In the following transformation steps, productions are rewritten so that matchings cAr appear exclusively in right-hand sides. Furthermore, we remove all productions that produce no matchings by introducing language preserving regular

expressions $\mathcal{R}$ in productions with right-hand sides of the form cAr, so that the resulting right-hand side is $\mathsf{c}\mathcal{R}\mathsf{r}$. Finally, adding a homomorphism that maps fresh terminal/co-terminal pairs to calls and returns, where the productions are modified accordingly, gives us a BG.

Definition 14. *Let* $G = (V, \Sigma_c \cup \Sigma_i \cup \Sigma_r, P, S)$ *denote an* immediate matching CFG*, a* BG $G''' = (V''', \underline{\Sigma} \cup \Sigma \cup \overline{\Sigma}, P''', S''')$ *and superficial mapping* h *are obtained from* G *in three steps as follows:*
First step: $\mathsf{A} \to_{G'} \mathsf{cBr}$ *iff* $\mathsf{A} \to_G \mathsf{cBr}$*,* $\mathsf{A} \to_{G'} \mathsf{A'C}, \mathsf{A'} \to \mathsf{cBr}$ *iff* $\mathsf{A} \to_G \mathsf{cBrC}$*,* $\mathsf{A} \to_{G'} \mathsf{iB}$ *iff* $\mathsf{A} \to_G \mathsf{iB}$*,* $\mathsf{A} \to_{G'} \varepsilon$ *iff* $\mathsf{A} \to_G \varepsilon$*.*
Second step: $\mathsf{A} \to_{G''} \mathsf{c}\mathcal{R}_B\mathsf{r}$ *iff* $\mathsf{A} \to_{G'} \mathsf{cBr}$*, where* $\mathcal{R}_B$ *describes the language* $L(B)$ *over* $\Sigma_i \cup V_{c\alpha r}$*,* $V_{c\alpha r} = \{A \mid A \to_{G'} \mathsf{cBr}\}$*.*
Third step: $\mathsf{A} \to_{G'''} \underline{\mathsf{a}}\mathcal{R}_B\overline{\mathsf{a}}$*,* $h(\underline{\mathsf{a}}) = \mathsf{c}$*,* $h(\overline{\mathsf{a}}) = \mathsf{r}$ *iff* $\mathsf{A} \to_{G''} \mathsf{c}\mathcal{R}_B\mathsf{r}$*.*

Lemma 4. *For any immediate matching* CFG G *and its corresponding* BG G' *plus superficial mapping* h *as of Definition 14, their languages coincide.*

Proof. In the first step, we only split up some productions into two separate productions $\mathsf{A} \to \mathsf{A'C}$ and $\mathsf{A'} \to \mathsf{cBr}$, which preserves language equivalence. In the second step, every non-terminal B in right-hand sides of the form cBr is substituted with it regular language over $\Sigma_i \cup V$. This is clearly just a syntactical abbreviation, and hence, does not modify the language either. Finally, in the third step, every call is replaced by a terminal and every return is replaced by a co-terminal, with an appropriate h respectively. □

Equivalence of ωRL(qBL) and ωVPL is now shown by translating an arbitrary ωRG(qBG) into an $\omega\mathrm{MSO}_\mu$ formula and an arbitrary ωVPA into an ωRG(qBG), where each time the languages of the characterisations coincide.

Theorem 1. *The language classes* ωRL(qBL)$+h$ *and* ωVPL *coincide.*

Proof. ωRL(qBL)$+h \subseteq \omega$VPL*:* Let $G = (V, \Sigma_c \cup \Sigma_i \cup \Sigma_r \cup \bigcup_{n=1}^m \{g_n\}, P, S, F, \bigcup_{n=1}^m \{G_n\}, h)$ denote an arbitrary ωRG(qBG)$+h$. Its injector language is regular, and hence, is representable as an ωMSO formula by the standard translation.

Each of the injected languages $L(G_n)$ is representable as $\mathrm{MSO}_{\mathrm{mwm}}$ formula φ_n respectively. Let φ'_n denote a variation of φ_n, where the formula presented in Lemma 2 is modified to $(\exists \boldsymbol{X} \bigvee_{(S \to \underline{\mathsf{a}} S_{\mathcal{R}} \overline{\mathsf{a}}) \in P} (\Psi_{\underline{\mathsf{a}},\overline{\mathsf{a}}}(\mathsf{i},\mathsf{j}) \wedge \forall k \in [\mathsf{i},\mathsf{j}] \Phi(k)))(\mathsf{i},\mathsf{j})$ but left unchanged otherwise. With appropriate renaming of variables, each terminal g_n can then be substituted by the corresponding formula φ'_n in the injector grammar, so that we get an ωMSO formula

$$\varphi \equiv \quad X_S(1) \ \wedge \forall k \left(\begin{array}{l} \bigvee_{(\mathsf{A} \to \mathsf{aB}) \in P} (X_\mathsf{A}(k) \wedge X_\mathsf{B}(k+1) \wedge T_\mathsf{a}(k)) \ \vee \\ \bigvee_{(\mathsf{A} \to g_n \mathsf{B}) \in P} \exists j (X_\mathsf{A}(k) \wedge X_\mathsf{B}(j+1) \wedge \varphi'_n(k,j)) \end{array} \right) \ \wedge \\ \bigvee_{\mathsf{A} \in F} \forall k \exists j (k < j \wedge X_\mathsf{A}(j)).$$

Language inclusion follows from the fact that every $\omega\mathrm{MSO}_\mu$ formula can be translated into an ωVPA.

ωRL(qBL)+$h \supseteq \omega$VPL*:* Consider an ωVPA A and let $A' = (Q', \Sigma_c \cup \Sigma_i \cup \Sigma_r \cup \bigcup_{n=1}^{m}\{\Omega_n\}, \delta, q_i, F')$ denote the Büchi-automaton accepting all pseudo-runs of A. A' can be represented as right-linear injector grammar $G_\uparrow$ with productions of the form $\mathsf{A} \to \mathsf{cB}$, $\mathsf{A} \to \mathsf{rB}$, and $\mathsf{A} \to (\mathsf{p},\mathsf{q},\mathsf{f})_{n'}\mathsf{B}$ for representing sets of summary-edges Ω_n with $(\mathsf{p},\mathsf{q},\mathsf{f})_{n'} \in \Omega_n$. Since summary-edges $(\mathsf{p},\mathsf{q},\mathsf{f})_{n'}$ are treated as terminals in A', their f component does not contribute to the acceptance of a pseudo-run. Hence, for every production $\mathsf{A} \to (\mathsf{p},\mathsf{q},1)_{n'}\mathsf{B}$, it is w.l.o.g. required that $\mathsf{B} \in F'$.

All summary-edges stand for languages in $\mathrm{VPL}_{\mathrm{mwm}}$, and hence, are representable as $\mathrm{VPA}_{\mathrm{mwm}}$s respectively. Each $\mathrm{VPA}_{\mathrm{mwm}}$ representing a summary-edge $(\mathsf{p},\mathsf{q},\mathsf{f})_{n'}$ can be transformed into a qBG $G_{n'}$ plus additional superficial mapping h. By combining $G_\uparrow$ and the various $G_{n'}$ to a superficial ωRG(qBG), we get the language inclusion. □

The use of qBGs is counter-productive. BGs are more accessible as well as succinct due to the use of regular expressions in their productions. Injecting BGs instead of qBGs into ωRGs does not change the expressiveness, which is trivially true as every BG can be translated into a qBG.

Corollary 1. *The language classes (*ω*)*RL(BL)+h *and (*ω*)*VPL *coincide.*

4 Conclusion

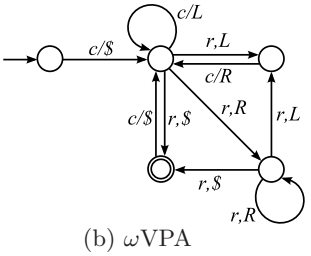
In this paper, a grammatical representation of (ω)VPLs was given. We introduced an amalgamation of ωRGs and qBGs equipped with a specific homomorphisms and showed that the resulting language class defined by the new grammar coincides with the ωVPLs.

Our grammatical approach towards (ω)VPLs provides a more natural representation of language specifications. As a small example, consider the following. Figure1(a) on the next page shows the code for traversing infinitely many finite binary trees. Let c denote a call to `traverse(...)` and let r denote the return in `traverse(...)`, then the ωVPA in Figure1(b) and the ωRG(BG)+h in Figure1(c) represent all possible traces that can be generated by `main`, i.e. they are behavioural specifications of the code. It is apparent that the ωRG(BG)+h resembles the pseudo code in greater detail than the corresponding ωVPA.

```
main :=
  do forever
    traverse(getTree())

function traverse(node n) :=
  if 'n is not a leaf' then
    traverse(n's left child)
    traverse(n's right child)
  return
```

(a) Pseudo code



(b) ωVPA

$S \to g_1 S$
$S_1 \to \underline{a} S_1 S_1 \overline{a} \mid \underline{a}\overline{a}$

$h(\underline{a}) = c, h(\overline{a}) = r$
$F = \{S\}$

(c) ωRG(BG)+h

Fig. 1. Representations of `traverse(...)`'s calls and returns

Acknowledgements. Joachim Baran thanks the EPSRC and the School of Computer Science for the research training awards enabling this work to be undertaken, as well as Juan Antonio Navarro-Pérez for helpful and valuable discussions regarding second-order logic.

References

[AM04] Alur, R., Madhusudan, P.: Visibly pushdown languages. In: Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, pp. 202–211. ACM Press, New York (2004)

[AM06] Alur, R., Madhusudan, P.: Adding nesting structure to words. In: Ibarra, O.H., Dang, Z. (eds.) DLT 2006. LNCS, vol. 4036, pp. 1–13. Springer, Heidelberg (2006)

[BB02] Berstel, J., Boasson, L.: Balanced grammars and their languages. In: Brauer, W., Ehrig, H., Karhumäki, J., Salomaa, A. (eds.) Formal and Natural Computing. LNCS, pp. 3–25. Springer, Heidelberg (2002)

[[Eng92] Engelfriet, J.: An elementary proof of double Greibach normal form. Information Processing Letters 44(6), 291–293 (1992)

[HMU01] Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation, 2nd edn. Addison-Wesley, Reading (2001)

[LMS04] Löding, C., Madhusudan, P., Serre, O.: Visibly pushdown games. In: Lodaya, K., Mahajan, M. (eds.) FSTTCS 2004. LNCS, vol. 3328, pp. 408–420. Springer, Heidelberg (2004)