On the Expressive Power of 2-Stack Visibly Pushdown Automata

Benedikt Bollig

Research Report LSV-07-27

September 2007

Laboratoire Spécification et Vérification

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

ON THE EXPRESSIVE POWER OF 2-STACK VISIBLY PUSHDOWN AUTOMATA

BENEDIKT BOLLIG

LSV, ENS Cachan, CNRS — 61, avenue du Président Wilson, 94235 Cachan Cedex, France
e-mail address: bollig@lsv.ens-cachan.fr

Abstract. Visibly pushdown automata are input-driven pushdown automata that recognize some non-regular context-free languages while preserving the nice closure and decidability properties of finite automata. Visibly pushdown automata with multiple stacks have been considered recently by La Torre, Madhusudan, and Parlato, who exploit the concept of visibility further to obtain a rich pushdown-automata class that can even express properties beyond the class of context-free languages. At the same time, their automata are closed under boolean operations, come up with a decidable emptiness and inclusion problem, and enjoy a logical characterization in terms of monadic second-order logic over nested words, which add a nesting structure to ordinary words. These results require a restricted version of visibly pushdown automata with multiple stacks whose behavior can be split up into a fixed number of phases.

In this paper, we consider 2-stack visibly pushdown automata (i.e., visibly pushdown automata with two stacks) in their unrestricted form. Our main results in this regard read as follows:

(1) 2-stack visibly pushdown automata are expressively equivalent to the existential fragment of monadic second-order logic.
(2) Over nested words, monadic second-order quantifier alternation forms an infinite hierarchy (unlike in the restricted framework by La Torre et al., where full monadic second-order logic is only as expressive as its existential fragment).
(3) 2-stack visibly pushdown automata are not closed under complementation.

Finally, we discuss the expressive power of Büchi 2-stack visibly pushdown automata running over infinite (nested) words. Extending the logic by an infinity quantifier, we can likewise establish equivalence to existential monadic second-order logic.

1. Introduction

The notion of a regular word language has ever played an important rôle in computer science, as it constitutes a robust concept that comes up with manifold representations in terms of finite automata, regular expressions, monadic second-order logic, etc. Generalizing regular languages towards richer classes and more expressive formalisms is often accompanied by the loss of robustness and decidability properties. It is, for example, well-known that the class of context-free languages, represented by pushdown automata, is no longer closed under complementation and that universality, equivalence, and inclusion are undecidable problems [10].

Key words and phrases: visibly pushdown automata, nested words, monadic second-order logic.

Visibly pushdown languages have been introduced by Alur and Madhusudan to overcome this deficiency while subsuming many interesting and useful context-free properties [1]. Visibly pushdown languages are represented by special pushdown automata whose stack operations are driven by the input. More precisely, the underlying alphabet of possible actions is partitioned into (1) call, (2) return, and (3) internal actions, which, when reading an action, indicates if (1) a stack symbol is pushed on the stack, (2) a stack symbol is read and popped from the stack, or (3) the stack is not touched at all, respectively. Such a partition gives rise to a *call-return alphabet*. Though this limits the expressive power of pushdown automata, the such defined class of visibly pushdown languages is rich enough to model various interesting non-regular properties for program analysis. Even more, this class preserves some important closure properties of regular languages, such as the closure under boolean operations, and it comes up with decidable problems, such as inclusion, that are undecidable in the context of general pushdown automata. Last but not least, the visibly pushdown languages are captured by a monadic second-order logic that makes use of a binary nesting predicate. Such a logic is suitable in the context of visibility, as the nesting structure of a word is uniquely determined, regardless of a particular run of the pushdown automaton. The logical characterization smoothly extends the classical theory of regular languages [5, 8].

Visibly pushdown automata with multiple stacks have been considered recently and independently by La Torre, Madhusudan, and Parlato [16], as well as Carotenuto, Murano, and Peron [6]. The aim of these papers is to exploit the concept of visibility further to obtain even richer classes of non-regular languages while preserving important closure properties and decidability of verification-related problems such as emptiness and inclusion.

In [16], the authors consider visibly pushdown automata with arbitrarily many stacks. To retain the nice properties of visibly pushdown automata with only one stack, the idea is to restrict the domain, i.e., the possible inputs, to those words that can be divided into at most k phases for a predefined k. In every phase, pop actions correspond to one and the same stack. These restricted visibly pushdown automata come up with a decidable emptiness problem, which is shown by a reduction to the emptiness problem for finite tree automata, and are closed under union, intersection, and complementation (wrt. the domain of k-phase words). Moreover, a word language is recognizable if and only if it can be defined in monadic second-order logic where the usual logic over words is expanded by a matching predicate that matches a push with its corresponding pop event. As mentioned above, such a matching is unique wrt. the underlying call-return alphabet. The only negative result in this regard is that multi-stack visibly pushdown automata cannot be determinized.

The paper [6] considers visibly pushdown automata with two stacks and call-return alphabets that appear more general than those of [16]: Any stack is associated with a partition of one and the same alphabet into call, return, and local transitions so that an action might be both a call action for the first stack and, at the same time, a return action for the second. In this way, both stacks can be worked on simultaneously. Note that, if we restrict to the alphabets of [16] where the stack alphabets are disjoint, the models from [6] and [16] coincide. Carotenuto et al. show that the emptiness problem for their model is undecidable (with the call-return alphabet belonging to the input). Their approach to gain decidability and, moreover, interesting closure properties is to exclude simultaneous pop operations by introducing an ordering constraint on stacks. More precisely, a pop operation on the second stack is only possible if the first stack is empty. Under these restrictions, the emptiness problem turns out to be decidable in polynomial time, and one obtains a language class that is closed under union, intersection, and complementation. Unlike the model of [16], any restricted automaton can be transformed into an equivalent deterministic one.

In this paper, we consider 2-stack visibly pushdown automata (i.e., visibly pushdown automata with two stacks) where any action is exclusive to one of the stacks, unless we deal with an internal action, which does not affect the stacks at all. Thus, we adopt the model of [16], though we have to restrict to two stacks for our main results. One of these results states that the corresponding language class is precisely characterized by the existential fragment of monadic second-order logic where a first-order kernel is preceded by a block of existentially quantified second-order variables. In a second step, we show that the full monadic second-order logic is strictly more expressive than its existential fragment so that we conclude that 2-stack visibly pushdown automata are not closed under complementation.

The key technique in our proofs is to consider words over call-return alphabets as relational structures, called *nested words* [2]. Nested words augment ordinary words with a nesting relation that, as the logical atomic predicate mentioned above, relates push with corresponding pop events. More precisely, we consider a nested word to be a graph whose nodes are labeled with actions and are related in terms of a matching and an immediate-predecessor relation. We thus deal with structures of bounded degree: any node has at most two incoming edges (one from the immediate successor and one from a push event if we deal with a pop event operating on the non-empty stack) and, similarly, at most two outgoing edges. As there is a one-to-one correspondence between words and their nested counterpart, we may consider nested-word automata [2], which are equivalent to visibly pushdown automata but operate on the enriched word structures. There have been several notions of automata on graphs and partial orders [13, 14] that are similar to nested-word automata and have one idea in common: the state that is taken after executing some event depends on the states that have been visited in neighboring events. Such defined automata may likewise operate on models for concurrent-systems executions such as Mazurkiewicz traces [7] and message sequence charts [4]. In the framework of nested-word automata, to determine the state after executing a pop operation, we therefore have to consider both the state of the immediate-predecessor position and the state that had been reached after the execution of the corresponding push event. To obtain a logical characterization of nested-word automata over two stacks, we adopt a technique from [4]: for a natural number r, we compute a nested-word automaton $\mathcal{B}_r$ that computes the *sphere* of radius r around any event i, i.e., the restriction of the input word to those events that have distance at most r from i. Once we have this automaton, we can apply Hanf's Theorem, which states that satisfaction of a given first-order formula depends on the number of these local spheres counted up to a threshold that depends on the quantifier-nesting depth of the formula [9]. This finally leads us to a logical characterization of 2-stack visibly pushdown automata in terms of existential monadic second-order logic. Note that our construction of $\mathcal{B}_r$ is close to the nontrivial technique applied in [4]. In the context of nested words, however, the correctness proof is more complicated. The fact that we deal with two stacks only is crucial, and the construction fails as soon as a third stack comes into play.

Then, we exploit the concept of nested words to show that full monadic second-order logic is more expressive than its existential fragment. This is done by a first-order interpretation of nested words over two stacks into grids, for which the analogous result has been known [12].

An extension of Hanf's Theorem has recently been established to cope with infinite structures [3]. This allows us to apply the automaton $\mathcal{B}_r$ to also obtain a logical characterization of the canonical extension of 2-stack visibly pushdown automata towards Büchi automata running over infinite words.

Outline of the paper. In Section 2, we introduce multi-stack visibly pushdown automata, running over words, as well as multi-stack nested-word automata, which operate on nested words. We establish expressive equivalence of these two models. Section 3 recalls monadic second-order logic over relational structures and, in particular, nested words. There, we also state Hanf's Theorem, which provides a normal form of first-order definable properties in terms of spheres. The construction of the sphere automaton $\mathcal{B}_r$, which is, to some extent, the core contribution of this paper, is the subject of Section 4. By means of this automaton, we can show expressive equivalence of 2-stack visibly pushdown automata and existential monadic second-order logic (Section 5). Section 6 establishes the gap between this fragment and the full logic, from which we conclude that 2-stack visibly pushdown automata cannot be complemented in general. By slightly modifying our logic, we obtain, in Section 7, a characterization of Büchi 2-stack visibly pushdown automata, running on infinite words. We conclude with Section 8 stating some related open problems.

2. Multi-Stack Visibly Pushdown Automata

The set $\{0, 1, 2, \ldots\}$ of natural numbers is denoted by $\mathbb{N}$, the set $\{1, 2, \ldots\}$ of positive natural numbers by $\mathbb{N}_+$. We call any finite set an *alphabet*. For a set Σ, we denote by Σ^*, Σ^+, and Σ^ω the sets of finite, nonempty finite, and infinite strings over Σ, respectively.[1] The empty word is denoted by ε. For a natural number $n \in \mathbb{N}$, we let $[n]$ stand for the set $\{1, \ldots, n\}$ (i.e., $[0]$ is the empty set). In this paper, we will identify isomorphic structures and we use both $=$ and $\cong$ to denote isomorphism.

Let $K \geq 1$. A (K-stack) *call-return alphabet* is a collection $\langle\{(\Sigma_c^s, \Sigma_r^s)\}_{s\in[K]}, \Sigma_{int}\rangle$ of pairwise disjoint alphabets. We allow Σ_{int} to be empty whereas the other alphabets are supposed to be nonempty. Intuitively, Σ_c^s contains the actions that call the stack s, Σ_r^s is the set of returns of stack s, and Σ_{int} is a set of internal actions, which do not involve any stack operation.

We fix $K \geq 1$ and a K-stack call-return alphabet $\widetilde{\Sigma} = \langle\{(\Sigma_c^s, \Sigma_r^s)\}_{s\in[K]}, \Sigma_{int}\rangle$. Moreover, we set $\Sigma_c = \bigcup_{s\in[K]} \Sigma_c^s$, $\Sigma_r = \bigcup_{s\in[K]} \Sigma_r^s$, and $\Sigma = \Sigma_c \cup \Sigma_r \cup \Sigma_{int}$.

2.1. **Multi-Stack Visibly Pushdown Automata.**

Definition 2.1. A *multi-stack visibly pushdown automaton* (Mvpa) over $\widetilde{\Sigma}$ is a tuple $\mathcal{A} = (Q, \Gamma, \delta, Q_I, F)$ where

- Q is its finite set of *states*,
- $Q_I \subseteq Q$ is the set of *initial states*,
- $F \subseteq Q$ is the set of *final states*,
- Γ is the finite *stack alphabet* containing a special symbol $\perp$, and

[1] From now on, to avoid confusion with nested words, we use the term "string" rather than "word" if we deal with elements from Σ^*.

- δ provides the *transitions* in terms of a triple $\langle \delta_c, \delta_r, \delta_{int} \rangle$ with

$$\begin{aligned} \delta_c &\subseteq Q \times \Sigma_c \times (\Gamma \setminus \{\perp\}) \times Q, \\ \delta_r &\subseteq Q \times \Sigma_r \times \Gamma \times Q, \text{ and} \\ \delta_{int} &\subseteq Q \times \Sigma_{int} \times Q \;. \end{aligned}$$

A 2-*stack visibly pushdown automaton* (2VPA) is an MVPA that is defined over a 2-stack alphabet (i.e., $K = 2$).

A transition $(q, a, A, q') \in \delta_c$, say with $a \in \Sigma_c^s$, is a push-transition meaning that, being in state q, the automaton can read a, push the symbol $A \in \Gamma \setminus \{\perp\}$ onto the s-th stack, and go over to state q'. A transition $(q, a, A, q') \in \delta_r$, say with $a \in \Sigma_r^s$, allows us to pop $A \neq \perp$ from the s-th stack when reading a, while the control changes from state q to state q'. If, however, $A = \perp$, then the stack is not touched, i.e., $\perp$ is never popped. Finally, a transition $(q, a, q') \in \delta_{int}$ is applied when reading internal actions $a \in \Sigma_{int}$. They do not involve any stack operation and, actually, do not even allow us to read from the stack.

Let us formalize the behavior of the MVPA $\mathcal{A}$. A *stack contents* is a nonempty finite sequence from $Cont = (\Gamma \setminus \{\perp\})^* \cdot \{\perp\}$. The leftmost symbol is thus the top symbol of the stack contents. A configuration of $\mathcal{A}$ consists of a state and a stack contents for any stack. Hence, it is an element of $Q \times Cont^{[K]}$. Consider a string $w = a_1 \dots a_n \in \Sigma^+$. A *run* of the MNWA $\mathcal{A}$ on w is a sequence $\rho = (q_0, \sigma_0^1, \dots \sigma_0^K) \dots (q_n, \sigma_n^1, \dots \sigma_n^K) \in (Q \times Cont^{[K]})^+$ such that $q_0 \in Q_I$, $\sigma_0^s = \perp$ for any stack $s \in [K]$, and, for any $i \in \{1, \dots, n\}$, the following hold:

[Push]: If $a_i \in \Sigma_c^s$ for $s \in [K]$, then there is a stack symbol $A \in \Gamma \setminus \{\perp\}$ such that $(q_{i-1}, a_i, A, q_i) \in \delta_c$, $\sigma_i^s = A \cdot \sigma_{i-1}^s$, and $\sigma_i^{s'} = \sigma_{i-1}^{s'}$ for any $s' \in [K] \setminus \{s\}$.

[Pop]: If $a_i \in \Sigma_r^s$ for $s \in [K]$, then there is a stack symbol $A \in \Gamma$ such that $(q_{i-1}, a_i, A, q_i) \in \delta_r$, $\sigma_i^{s'} = \sigma_{i-1}^{s'}$ for any $s' \in [K] \setminus \{s\}$, and either $A \neq \perp$ and $\sigma_{i-1}^s = A \cdot \sigma_i^s$, or $A = \perp$ and $\sigma_{i-1}^s = \sigma_i^s = \perp$.

[Internal]: If $a_i \in \Sigma_{int}$, then $(q_{i-1}, a_i, q_i) \in \delta_{int}$, and $\sigma_i^s = \sigma_{i-1}^s$ for any $s \in [K]$.

The run ρ is accepting if $q_n \in F$. A string $w \in \Sigma^+$ is accepted by $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on w. The set of accepted strings forms the (string) language of $\mathcal{A}$, which is denoted by $L(\mathcal{A})$.[2]

Example 2.2. There is no MVPA that recognizes the context-sensitive language $\{a^n b^n c^n \mid n \geq 1\}$, no matter which call-return alphabet we chose. Note that, however, with the more general notion of a call-return alphabet from [6], it is possible to recognize this language by means of two stacks. Now consider the 2-stack call-return alphabet $\widetilde{\Sigma}$ given by $\Sigma_c^1 = \{a_1\}$, $\Sigma_r^1 = \{b_1\}$, $\Sigma_c^2 = \{a_2\}$, $\Sigma_r^2 = \{b_2\}$, and $\Sigma_{int} = \emptyset$. The language $L = \{(a_1 a_2)^n b_1^{n+1} b_2^{n+1} \mid n \geq 1\}$ can be recognized by some 2VPA over $\widetilde{\Sigma}$, even by the restricted model of 2-phase 2VPA from [16], as any word from L can be split into at most two return phases. In the following, we define a 2VPA $\mathcal{A} = (\{q_0, \dots, q_4\}, \{\$, \perp\}, \delta, \{q_0\}, \{q_0\})$ over $\widetilde{\Sigma}$ such that $L(\mathcal{A}) = L^+$, which is no longer divisible into a bounded number of return phases. The transition relation δ is

[2]To simplify the presentation, the empty word ε is excluded from the domain.

given as follows:

$$\begin{array}{llll} \delta_c: & (q_0, a_1, \$, q_2) & \delta_r: & (q_3, b_1, \$, q_3) \\ & (q_2, a_2, \$, q_1) & & (q_3, b_1, \bot, q_4) \\ & (q_1, a_1, \$, q_2) & & (q_4, b_2, \$, q_4) \\ & (q_2, a_2, \$, q_3) & & (q_4, b_2, \bot, q_0) \end{array}$$

The idea is that the finite-state control ensures that an input word matches the regular expression $((a_1a_2)^+b_1^+b_2^+)^+$. To guarantee that, in any iteration, the number of a_k is by one less than the number of b_k, any push action a_k stores a stack symbol \$ in stack k, which can then be removed by the corresponding pop actions unless the symbol $\bot$ is discovered.

2.2. **Nested Words and Multi-Stack Nested-Word Automata.** We will now see how strings over symbols from the call-return alphabet $\widetilde{\Sigma}$ can be represented by relational structures. Basically, to a string, we add a binary predicate that combines push with corresponding pop events. Let $s \in [K]$. A string $w \in \Sigma^*$ is called s-well formed if it is generated by the following context-free grammar:

$$A ::= aAb \mid AA \mid \varepsilon \mid c$$

where $a \in \Sigma_c^s$, $b \in \Sigma_r^s$, and $c \in \Sigma \setminus (\Sigma_c^s \cup \Sigma_r^s)$.

Definition 2.3. A *nested word* over $\widetilde{\Sigma}$ is a structure $([n], \lessdot, \mu, \lambda)$ where $n \geq 1$ (we call the elements from $[n]$ *positions*, *nodes*, or *events*), $\lessdot = \{(i, i+1) \mid i \in [n-1]\}$, $\lambda : [n] \to \Sigma$, and $\mu = \bigcup_{s \in [K]} \mu^s \subseteq [n] \times [n]$ where, for any $s \in [K]$ and $(i,j) \in [n] \times [n]$, $(i,j) \in \mu^s$ iff $i < j$, $\lambda(i) \in \Sigma_c^s$, $\lambda(j) \in \Sigma_r^s$, and $\lambda(i+1) \ldots \lambda(j-1)$ is s-well formed.

The set of nested words over $\widetilde{\Sigma}$ is denoted by $\mathbb{NW}(\widetilde{\Sigma})$.

Note that μ and its inverse μ^{-1} can be seen as partial maps $[n] \dashrightarrow [n]$ in the obvious manner. Moreover, observe that, given nested words $W = ([n], \lessdot, \mu, \lambda)$ and $W' = ([n'], \lessdot', \mu', \lambda')$, $n = n' \wedge \lambda = \lambda'$ implies $W = W'$. It is therefore justified to represent W as the string $\mathrm{string}(W) := \lambda(1) \ldots \lambda(n) \in \Sigma^+$. This naturally extends to sets $\mathcal{L}$ of nested words and we set $\mathrm{string}(\mathcal{L}) := \{\mathrm{string}(W) \mid W \in \mathcal{L}\}$. Vice versa, given a string $w \in \Sigma^+$, there is precisely one nested word W over $\widetilde{\Sigma}$ such that $\mathrm{string}(W) = w$. This unique nested word is denoted $\mathrm{nested}(w)$. For $L \subseteq \Sigma^+$, we let $\mathrm{nested}(L) := \{\mathrm{nested}(w) \mid w \in L\}$.

Example 2.4. Consider the 2-stack call-return alphabet $\widetilde{\Sigma}$ from Example 2.2, which was given by $\Sigma_c^1 = \{a_1\}$, $\Sigma_r^1 = \{b_1\}$, $\Sigma_c^2 = \{a_2\}$, $\Sigma_r^2 = \{b_2\}$, and $\Sigma_{int} = \emptyset$. Figure 1 depicts a nested word $W = ([n], \lessdot, \mu, \lambda)$ over $\widetilde{\Sigma}$ with $n = 10$. The straight arrows represent $\lessdot$, the curved arrows capture μ (those above the horizontal correspond to the first stack). For example, $(2, 9) \in \mu$. Thus, $\mu(2)$ and $\mu^{-1}(9)$ are defined, whereas both $\mu^{-1}(7)$ and $\mu^{-1}(10)$ are not. In terms of visibly pushdown automata, this means that positions 7 and 10 are employed when the first/second stack is empty, respectively. Observe that $W = \mathrm{nested}(a_1a_2a_1a_2b_1b_1b_1b_2b_2b_2)$ and $\mathrm{string}(W) = a_1a_2a_1a_2b_1b_1b_1b_2b_2b_2$.

We now turn to an automata model that is suited to nested words and, to some extent, is equivalent to Mvpa. This model has been considered in [2] for nested words over 2-stack call-return alphabets.

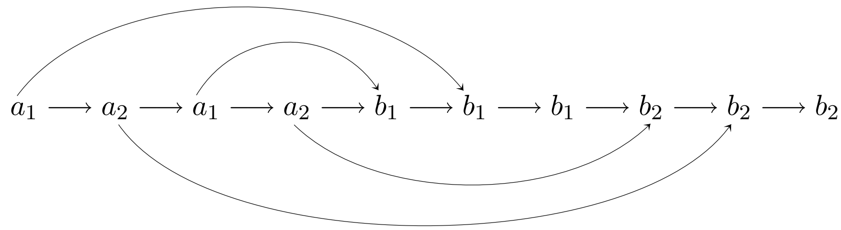

Figure 1: A nested word

Definition 2.5. A *generalized multi-stack nested-word automaton* (generalized MNWA) over $\widetilde{\Sigma}$ is a tuple $\mathcal{B} = (Q, \delta, Q_I, F, C)$ where

- Q is the finite set of *states*,
- $Q_I \in Q$ is the set of *initial states*,
- $F \subseteq Q$ is the set of *final states*,
- $C \subseteq Q$ is a set of *calling states*, and
- δ is a pair $\langle \delta_1, \delta_2 \rangle$ with $\delta_1 \subseteq Q \times \Sigma \times Q$ and $\delta_2 \subseteq Q \times Q \times \Sigma_r \times Q$, which contain the *transitions*.

We call $\mathcal{B}$ a *multi-stack nested-word automaton* (MNWA) if $C = \emptyset$.

A (*generalized*) 2-stack nested-word automaton (2NWA) is a (generalized, respectively) MNWA that is defined over a 2-stack alphabet (i.e., $K = 2$).

Intuitively, δ_1 contains all the local and push transitions, as well as all the pop transitions that act on an empty stack (i.e., in terms of nested words and nested-word automata, those transitions that perform an action from Σ_r that is not matched by a corresponding calling action). A run of $\mathcal{B}$ on a nested word $W = ([n], \lessdot, \mu, \lambda)$ over $\widetilde{\Sigma}$ is a mapping $\rho : [n] \to Q$ such that $(q, \lambda(1), \rho(1)) \in \delta_1$ for some $q \in Q_I$, and, for any $i \in \{2, \ldots, n\}$, we have

$$\begin{cases} (\rho(\mu^{-1}(i)), \rho(i-1), \lambda(i), \rho(i)) \in \delta_2 & \text{if } \lambda(i) \in \Sigma_r \text{ and } \mu^{-1}(i) \text{ is defined} \\ (\rho(i-1), \lambda(i), \rho(i)) \in \delta_1 & \text{otherwise} \end{cases}$$

The run ρ is accepting if $\rho(n) \in F$ and, for any $i \in [n]$ with $\rho(i) \in C$, we have both $\lambda(i) \in \Sigma_c$ and $\mu(i)$ is defined. The language of $\mathcal{B}$, denoted by $\mathcal{L}(\mathcal{B})$, is the set of nested words that allow for an accepting run of $\mathcal{B}$.

Recall that there is a one-to-one correspondence between strings and nested words. We let therefore $\mathcal{L}(\mathcal{A})$ with $\mathcal{A}$ an MVPA stand for the set nested$(L(\mathcal{A}))$.

Example 2.6. Consider again the 2-stack call-return alphabet $\widetilde{\Sigma}$ given by $\Sigma_c^1 = \{a_1\}$, $\Sigma_r^1 = \{b_1\}$, $\Sigma_c^2 = \{a_2\}$, $\Sigma_r^2 = \{b_2\}$, and $\Sigma_{int} = \emptyset$. In Example 2.2, we have seen that, for $L = \{(a_1 a_2)^n b_1^{n+1} b_2^{n+1} \mid n \geq 1\}$, the iteration L^+ is the language of some 2VPA over $\widetilde{\Sigma}$. We can also specify a 2NWA $\mathcal{B} = (\{q_0, \ldots, q_4\}, \delta, \{q_0\}, \{q_0\}, \emptyset)$ over $\widetilde{\Sigma}$ such that $\mathcal{L}(\mathcal{B}) = \text{nested}(L^+)$. Note that $\mathcal{L}(\mathcal{B})$ will contain, for example, the nested word that is depicted in Figure 1. The transition relation δ is given as follows:

$$\begin{array}{llll} \delta_1: & (q_0, a_1, q_2) & \delta_2: & (q_2, q_3, b_1, q_3) \\ & (q_2, a_2, q_1) & & (q_3, q_4, b_2, q_4) \\ & (q_1, a_1, q_2) & & (q_1, q_4, b_2, q_4) \\ & (q_2, a_2, q_3) & & \\ & (q_3, b_1, q_4) & & \\ & (q_4, b_2, q_0) & & \end{array}$$

Similarly to Example 2.2, the finite-state control will ensure the general regular structure of a word without explicit counting. This "counting" is then implicitly done by the relation δ_2, which requires a matching a_k for any b_k. A general technique for a reduction from Mvpa to Mnwa and vice versa can be found below (Lemma 2.8).

We can show that the use of calling states does not increase the expressiveness of Mnwa. Note that, however, the concept of calling states will turn out to be helpful when building the sphere automaton in Section 4.

Lemma 2.7. For any generalized Mnwa $\mathcal{B}$ over $\widetilde{\Sigma}$, there is an Mnwa $\mathcal{B}'$ over $\widetilde{\Sigma}$ such that $\mathcal{L}(\mathcal{B}') = \mathcal{L}(\mathcal{B})$.

Proof. In the construction of an Mnwa, we exploit the following property of a nested word $W = ([n], \lessdot, \mu, \lambda)$: given $(i, j) \in \mu$, say, with $i \in \Sigma_c^s$, $\mu(i')$ is defined for all $i' \in \{i+1, \ldots, j-1\}$ satisfying $i' \in \Sigma_c^s$. Basically, $\mathcal{B}'$ will simulate $\mathcal{B}$. In addition, whenever a calling state is assigned to a position labeled with an element from Σ_c^s, we will set a flag $\overline{b}[s] = 1$, which can only be resolved and turn into a final state ($\overline{b}[s] = 0$) when a matching return position has been found. As any interim call position that concerns stack s is matched anyway, the flags $\overline{b}[s]$ in that interval are set to 2. Thus, while a flag is 1 or 2, there is still some unmatched calling position. Hence, a final state requires any flag to equal 0, which also designates the initial state.

Let us become more precise and let $\mathcal{B} = (Q, \delta, Q_I, F, C)$ be a generalized Mnwa. We determine the Mnwa $\mathcal{B}' = (Q', \delta', Q_I', F', \emptyset)$ by $Q' = Q \times \{0, 1, 2\}^{[K]}$, $Q_I' = Q_I \times \{(0)_{s \in [K]}\}$, $F' = F \times \{(0)_{s \in [K]}\}$, and $\delta' = \langle \delta_1', \delta_2' \rangle$ where

- δ_1' is the set of triples $((q, \overline{b}), a, (q', \overline{b}')) \in Q' \times \Sigma \times Q'$ such that $(q, a, q') \in \delta_1$, $q' \in C$ implies $a \in \Sigma_c$, and, for any $s \in [K]$,

$$\overline{b}'[s] = \begin{cases} 2 & \text{if } \overline{b}[s] \in \{1, 2\} \\ 1 & \text{if } \overline{b}[s] = 0 \text{ and } a \in \Sigma_c^s \text{ and } q' \in C \\ 0 & \text{otherwise} \end{cases}$$

- δ_2' is the set of quadruples $((p, \overline{c}), (q, \overline{b}), a, (q', \overline{b}')) \in Q' \times Q' \times \Sigma_r \times Q'$ such that $(p, q, a, q') \in \delta_2$ and, for any $s \in [K]$,

$$\overline{b}'[s] = \begin{cases} 0 & \text{if } \overline{c}[s] = 1 \\ \overline{b}[s] & \text{otherwise} \end{cases}$$

In fact, we have $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{B}')$. ∎

Lemma 2.8. *Let $\mathcal{L} \subseteq \mathbb{NW}(\widetilde{\Sigma})$ be a set of nested words over $\widetilde{\Sigma}$. The following are equivalent:*

(1) *There is an* Mvpa *$\mathcal{A}$ over $\widetilde{\Sigma}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.*
(2) *There is an* Mnwa *$\mathcal{B}$ over $\widetilde{\Sigma}$ such that $\mathcal{L}(\mathcal{B}) = \mathcal{L}$.*

Proof. Given an Mvpa $\mathcal{A} = (Q, \Gamma, \delta, Q_I, F)$, we define an Mnwa $\mathcal{B} = (Q', \delta', Q'_I, F', \emptyset)$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$ as follows: $Q' = Q \times \Gamma$, $Q'_I = Q_I \times \{\bot\}$, $F' = F \times \Gamma$, and $\delta' = \langle \delta'_1, \delta'_2 \rangle$ where

- δ'_1 is the set of triples $((q, A), a, (q', A') \in Q' \times \Sigma \times Q'$ such that $(q, a, A', q') \in \delta_c$, $(q, a, q') \in \delta_{int}$, or $(q, a, \bot, q') \in \delta_r$, and
- δ'_2 is the set of quadruples $((p, B), (q, A), a, (q', A') \in Q' \times Q' \times \Sigma \times Q'$ such that $(q, a, B, q') \in \delta_r$.

For the converse direction, consider an Mnwa $\mathcal{B} = (Q, \delta, Q_I, F, \emptyset)$. Consider the Mvpa $\mathcal{A} = (Q, Q \cup \{\bot\}, \delta', Q_I, F)$ where $\delta' = \langle \delta'_c, \delta'_r, \delta'_{int} \rangle$ is given by

- $\delta'_c = \{(q, a, q', q') \mid (q, a, q') \in \delta_1 \cap (Q \times \Sigma_c \times Q)\}$,
- $\delta'_{int} = \delta_1 \cap (Q \times \Sigma_{int} \times Q)$, and
- δ'_r is the set of tuples $(q, a, A, q') \in Q \times \Sigma_r \times \Gamma \times Q$ such that either $(q, a, q') \in \delta_1$ and $A = \bot$, or $(A, q, a, q') \in \delta_1$.

We have $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$. □

3. Monadic Second-Order Logic and Hanf's Theorem

3.1. **Monadic Second-Order Logic over Relational Structures.** We fix supplies of first-order variables $x, y, \ldots$ and second-order variables $X, Y, \ldots$. Let τ be a function-free signature. The set MSO(τ) of *monadic second-order* (MSO) formulas over τ is given by the following grammar:

$$\varphi ::= P(x_1, \ldots, x_m) \mid x_1 = x_2 \mid x \in X \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x\varphi \mid \exists X\varphi$$

Hereby, $m \in \mathbb{N}$, $P \in \tau$ is an m-ary predicate symbol, the x_k and x are a first-order variables, and X is a second-order variable. Moreover, we will make use of the usual abbreviations such as $\varphi_1 \wedge \varphi_2$ for $\neg(\neg\varphi_1 \vee \neg\varphi_2)$, $\varphi_1 \rightarrow \varphi_2$ for $\neg\varphi_1 \vee \varphi_2$, etc. Given a τ-structure $\mathfrak{A}$ with universe A, a formula $\varphi(x_1, \ldots, x_m, X_1, \ldots, X_n) \in \mathrm{MSO}(\tau)$ with free variables in $\{x_1, \ldots, x_m, X_1, \ldots, X_n\}$, $(u_1, \ldots, u_m) \in A^m$, and $(U_1, \ldots, U_n) \in (2^A)^n$, we write, as usual, $\mathfrak{A} \models \varphi[u_1, \ldots, u_m, U_1, \ldots, U_n]$ if $\mathfrak{A}$ satisfies φ when assigning $(u_1, \ldots, u_m)$ to $(x_1, \ldots, x_m)$ and $(U_1, \ldots, U_n)$ to $(X_1, \ldots, X_n)$.

Let us identify some important fragments of MSO(τ). The set FO(τ) of *first order* (FO) formulas over τ comprises those formulas from MSO(τ) that do not contain any second-order quantifier. Furthermore, an *existential* MSO (EMSO) formula is of the form $\exists X_1 \ldots \exists X_n \varphi$ with $\varphi \in \mathrm{FO}(\tau)$. The corresponding class of formulas is denoted EMSO(τ). More generally, given $m \geq 1$, we denote by $\mathbf{\Sigma}_m(\tau)$ the set of formulas of the form $\exists\overline{X_1}\forall\overline{X_2}\ldots\exists/\forall\overline{X_m}\varphi$ where $\varphi \in \mathrm{FO}(\tau)$ and the $\overline{X_k}$ are blocks of second-order variables, possibly empty or of different length.

We will later make use of the notion of *definability* relative to a class of structures. Let $\mathcal{F} \subseteq \mathrm{MSO}(\tau)$ be a class of formulas and $\mathcal{L}, \mathcal{C}$ be sets of τ-structures. We say that $\mathcal{L}$ is *$\mathcal{F}$-definable relative to $\mathcal{C}$* if there is a sentence (i.e., a formula without any free variables) $\varphi \in \mathcal{F}$ such that $\mathcal{L}$ is the set of τ-structures $\mathfrak{A} \in \mathcal{C}$ such that $\mathfrak{A} \models \varphi$.

Now let $\widetilde{\Sigma}$ be a call-return alphabet. We define $\tau_{\widetilde{\Sigma}}$ to be the signature $\{\lambda_a \mid a \in \Sigma\} \cup \{\lessdot, \mu\}$ with λ_a a unary and $\lessdot$ and μ binary predicate symbols. We write the MSO formula $\lambda_a(x)$ as $\lambda(x) = a$ and the formula $\lessdot(x_1, x_2)$ as $x_1 \lessdot x_2$. MSO formulas over $\tau_{\widetilde{\Sigma}}$ can be canonically interpreted over nested words $([n], \lessdot, \mu, \lambda) \in \mathbb{NW}(\widetilde{\Sigma})$, as λ can be seen as a collection of unary relations $\lambda_a = \{i \in [n] \mid \lambda(i) = a\}$ where $a \in \Sigma$. A sample MSO formula over $\tau_{\widetilde{\Sigma}}$ such that $\Sigma = \{a, b\}$ is $\forall x \forall y\, (\lambda(x) = a \wedge \mu(x, y) \rightarrow \lambda(y) = b)$. It expresses that any calling a is matched by a b. Given a sentence $\varphi \in \mathrm{MSO}(\tau_{\widetilde{\Sigma}})$, we denote by $\mathcal{L}(\varphi)$ the set of nested words that satisfy φ. We might likewise assign to φ the set $L(\varphi)$ of strings $w \in \Sigma^+$ such that $\mathrm{nested}(w) \models \varphi$.

The quantifier rank $\mathrm{rank}(\varphi)$ of a first-order formula $\varphi \in \mathrm{FO}(\tau_{\widetilde{\Sigma}})$ is the maximal number of quantifier nestings in φ. More precisely, we define $\mathrm{rank}(\lambda(x) = a) = \mathrm{rank}(x \lessdot y) = \mathrm{rank}(\mu(x, y)) = \mathrm{rank}(x = y) = \mathrm{rank}(x \in X) = 0$, $\mathrm{rank}(\neg\varphi) = \mathrm{rank}(\varphi)$, $\mathrm{rank}(\varphi_1 \vee \varphi_2) = \max\{\mathrm{rank}(\varphi_1), \mathrm{rank}(\varphi_2)\}$, and $\mathrm{rank}(\exists x \varphi) = \mathrm{rank}(\varphi) + 1$.

Definition 3.1. Let $k \in \mathbb{N}$. For nested words $U, V \in \mathbb{NW}(\widetilde{\Sigma})$, we write $U \equiv_{k,\widetilde{\Sigma}} V$ if, for any first-order sentence $\varphi \in \mathrm{FO}(\tau_{\widetilde{\Sigma}})$ with $\mathrm{rank}(\varphi) \leq k$, we have $U \models \varphi$ iff $V \models \varphi$.

Note that $\equiv_{k,\widetilde{\Sigma}}$ is an equivalence relation of finite index.

3.2. **Spheres, Threshold Equivalence, and Hanf's Theorem.** The notion of a sphere will play a central role in this paper. Let $\widetilde{\Sigma}$ be a call-return alphabet and let $\mathfrak{A} = (N, \lessdot, \mu, \lambda, \ldots)$ be a structure over a signature that subsumes $\tau_{\widetilde{\Sigma}}$. Given $i, j \in N$, The *distance* $d_{\mathfrak{A}}(i, j)$ of i and j in $\mathfrak{A}$ is the minimal length of a path from i to j in the graph $(N, \lessdot \cup \mu \cup (\lessdot \cup \mu)^{-1})$. If $d_{\mathfrak{A}}(i, j) = 1$, we also write $i \leftrightarrow_{\mathfrak{A}} j$. We write $i \rightarrow_{\mathfrak{A}} j$ if $(i, j) \in \lessdot \cup \mu$. Let r be a natural number and let $i \in N$. The *r-sphere* of $\mathfrak{A}$ around i, which we denote by $r\text{-}\mathrm{Sph}(\mathfrak{A}, i)$, is basically the substructure of $\mathfrak{A}$ induced by the new universe $\{j \in N \mid d_{\mathfrak{A}}(i, j) \leq r\}$, but extended by the constant i as a distinguished element, called the *sphere center*. For an example, consider the nested word W from Figure 3. The 2-sphere of W around $i = 10$ is shown in Figure 2 where the sphere center is marked as a rectangle. Note that $2\text{-}\mathrm{Sph}(W, 10) = 2\text{-}\mathrm{Sph}(W, 14)$. Given an isomorphism type S of an r-sphere, we let $|\mathfrak{A}|_S := |\{i \in N \mid S \text{ and } r\text{-}\mathrm{Sph}(\mathfrak{A}, i) \text{ are isomorphic}\}|$ denote the number of points in $\mathfrak{A}$ that *realize* S. Let $\mathfrak{A}' = (N', \lessdot', \mu', \lambda', \ldots)$ be another structure with a signature that subsumes $\tau_{\widetilde{\Sigma}}$. For $i, j \in N$ and $i', j' \in N'$, we write $(i, j) \sqsubseteq^{\mathfrak{A}}_{\mathfrak{A}'} (i', j')$ if $\lambda(i) = \lambda'(i')$, $\lambda(j) = \lambda'(j')$, $(i, j) \in \lessdot$ implies $(i', j') \in \lessdot'$, and $(i, j) \in \mu$ implies $(i', j') \in \mu'$.

For $r \in \mathbb{N}$, we denote by $\mathit{Spheres}_r(\widetilde{\Sigma})$ the set of r-spheres that arise from nested words over $\widetilde{\Sigma}$, i.e., $\mathit{Spheres}_r(\widetilde{\Sigma}) := \{r\text{-}\mathrm{Sph}(W, i) \mid W \in \mathbb{NW}(\widetilde{\Sigma}) \text{ and } i \text{ is a node of } W\}$.

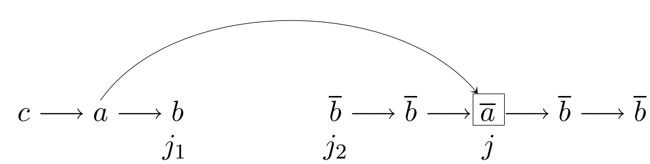

Figure 2: A 2-sphere ...

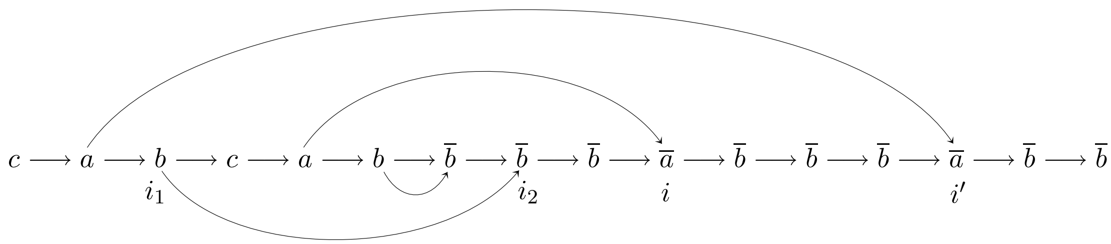

Figure 3: ... embedded into a nested word

Definition 3.2 (Threshold equivalence)**.** Let $r, t \in \mathbb{N}$ and let $U, V \in \mathbb{NW}(\widetilde{\Sigma})$ be nested words. We write $U \leftrightarrows_{r,t} V$ if, for any isomorphism type $S \in \mathit{Spheres}_r(\widetilde{\Sigma})$ of an r-sphere, we have $|U|_S = |V|_S$ or both $t < |U|_S$ and $t < |V|_S$.

Next, we state Hanf's locality theorem. For a comprehensive proof thereof, see [11, 15].

Theorem 3.3 (Hanf [9])**.** *Let $k \in \mathbb{N}$. There exist $r, t \in \mathbb{N}$ such that, for any two nested words $U, V \in \mathbb{NW}(\widetilde{\Sigma})$, $U \leftrightarrows_{r,t} V$ implies $U \equiv_{k,\widetilde{\Sigma}} V$. Hereby, r and t can be computed effectively such that $r \leq 3^k$ and $t \leq k3^k$.*

Thus, for any first-order sentence $\varphi \in \mathrm{FO}(\tau_{\widetilde{\Sigma}})$, there are $r \leq 3^{|\varphi|}$ and $t \leq |\varphi|3^{|\varphi|}$ such that $\mathcal{L}(\varphi)$ is the finite union of equivalence classes of $\leftrightarrows_{r,t}$. Observe that $\leftrightarrows_{r,t}$ is an equivalence relation of finite index.

4. The Sphere Automaton

For this section, we fix a 2-stack call-return alphabet $\widetilde{\Sigma} = \langle \{(\Sigma_c^1, \Sigma_r^1), (\Sigma_c^2, \Sigma_r^2)\}, \Sigma_{int} \rangle$. The key connection between first-order logic and 2VPA/2NWA is provided by the following proposition, which states the existence of an automaton that computes the sphere around any node of a nested word.

Proposition 4.1. Let r be any natural number. There are a generalized 2NWA $\mathcal{B}_r = (Q, \delta, Q_I, F, C)$ over $\widetilde{\Sigma}$ and a mapping $\eta : Q \to \mathit{Spheres}_r(\widetilde{\Sigma})$ such that

- $\mathcal{L}(\mathcal{B}_r) = \mathbb{NW}(\widetilde{\Sigma})$ (i.e., $\mathcal{L}(\mathcal{B}_r)$ is the set of all nested words), and
- for any nested word $W \in \mathbb{NW}(\widetilde{\Sigma})$, for any accepting run ρ of $\mathcal{B}_r$ on W, and for any node i of W, we have $\eta(\rho(i)) \cong r\text{-Sph}(W, i)$.

4.1. The Naïve Approach.

We first propose a rather obvious approach to constructing $\mathcal{B}_r$, which works in simpler settings, e.g., in the domain of strings, but will fail when considering nested words over at least two stacks. Namely, a first attempt to construct such a generalized 2NWA $\mathcal{B}_r$ would be to make each position i guess its sphere and then to show that these guesses can be verified by relating the guessed spheres of neighboring positions (i.e., positions $i-1$, i, and $\mu^{-1}(i)$ if i performs a pop action on a nonempty stack). We might thus proceed as follows: We set $Q = \mathit{Spheres}_r(\widetilde{\Sigma}) \cup \{\iota\}$ and $Q_I = \{\iota\}$. The set of final states is given by $F = \{r\text{-Sph}(W, n) \mid W = ([n], \lessdot, \mu, \lambda) \in \mathbb{NW}(\widetilde{\Sigma})\}$, the set of calling states by $C = \{(N, \lessdot, \mu, \lambda, i) \in \mathit{Spheres}_r(\widetilde{\Sigma}) \mid \mu(i) \text{ is defined}\}$. Finally, $\delta = \langle \delta_1, \delta_2 \rangle$ is given as follows:

- For $S, S' \in Q$ and $a \in \Sigma$, we let $(S, a, S') \in \delta_1$ if one of the following holds:
  - $S = \iota$ and there exists $W = ([n], \lessdot, \mu, \lambda) \in \mathbb{NW}(\widetilde{\Sigma})$ such that both $\lambda(1) = a$ and $S' \cong r\text{-Sph}(W, 1)$, or
  - there exist $W = ([n], \lessdot, \mu, \lambda) \in \mathbb{NW}(\widetilde{\Sigma})$ and $i, i' \in [n]$ such that
    (1) $\lambda(i') = a$
    (2) $(i, i') \in \lessdot \setminus \mu$
    (3) $S \cong r\text{-Sph}(W, i)$
    (4) $S' \cong r\text{-Sph}(W, i')$
- For $S_c, S, S' \in Q$ and $a \in \Sigma_r$, we let $(S_c, S, a, S') \in \delta_2$ if there exist a nested word $W = ([n], \lessdot, \mu, \lambda) \in \mathbb{NW}(\widetilde{\Sigma})$ and $i_c, i, i' \in [n]$ such that
  (1) $\lambda(i') = a$
  (2) $(i_c, i') \in \mu$
  (3) $(i, i') \in \lessdot$
  (4) $S_c \cong r\text{-Sph}(W, i_c)$
  (5) $S \cong r\text{-Sph}(W, i)$
  (6) $S' \cong r\text{-Sph}(W, i')$

However, this straightforward approach does not work. Consider W to be the nested word over the 2-stack call-return alphabet $\langle \{(\{a\}, \{\overline{a}\}), (\{b\}, \{\overline{b}\})\}, \{c\} \rangle$ that is illustrated in Figure 4. For $r = 2$, it is possible to define an accepting run of the above automaton on W that assigns the 2-sphere S from Figure 5 to the first position of W, though S is not isomorphic to 2-Sph$(W, 1)$. Actually, the sphere 2-Sph$(W, 1)$ exhibits an edge between the a- and the $\overline{a}$-labeled position, which is missing in S. However, a run of our automaton can "pass" the missing edge through W resulting in an accepting run. The whole run is specified in the appendix.

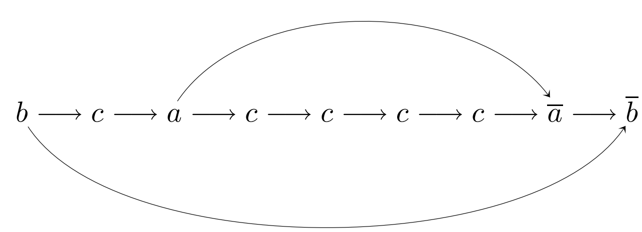

Figure 4: Why the naïve approach fails (1)

4.2. **The Solution.** We now turn to the correct solution. In any state, the generalized 2NWA $\mathcal{B}_r$ will guess the current sphere as well as spheres of nodes nearby and the current position in these additional spheres. Adding some global information allows us to locally check whether all the guesses are correct. The rest of this section is devoted to the construction of $\mathcal{B}_r$ and a corresponding mapping η to prove Proposition 4.1.

$$\begin{tikzcd} \boxed{b} \arrow[r] \arrow[rrrr, bend right] & c \arrow[r] & a & \overline{a} \arrow[r] & \overline{b} \end{tikzcd}$$

Figure 5: Why the naïve approach fails (2)

4.2.1. *The Construction.* Recall that $\mathit{Spheres}_r(\widetilde{\Sigma})$ denotes the set of all the r-spheres that arise from nested words, i.e., $\mathit{Spheres}_r(\widetilde{\Sigma}) = \{r\text{-Sph}(W,i) \mid W$ is a nested word and i is a position in $W\}$. An *extended* r-sphere over $\widetilde{\Sigma}$ is a structure $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col})$ where $\mathit{core}(E) := (N, \lessdot, \mu, \lambda, \gamma) \in \mathit{Spheres}_r(\widetilde{\Sigma})$ (in particular, $\gamma \in N$), $\alpha \in N$, and $\mathit{col} \in [\#\mathit{Col}]$ with $\#\mathit{Col} = 4 \cdot \mathit{maxN}^2 + 1$ where maxN is the maximal size of an r-sphere, i.e., $\mathit{maxN} = \max\{|N| \mid (N, \lessdot, \mu, \lambda, i) \in \mathit{Spheres}_r(\widetilde{\Sigma})\}$. We say that α is the *active node* of E and col is its *color*. Let $\mathit{eSpheres}_r(\widetilde{\Sigma})$ denote the set of all the extended spheres over $\widetilde{\Sigma}$. For an extended sphere $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col})$ and an element $i \in N$, we denote by $E[i]$ the extended sphere $(N, \lessdot, \mu, \lambda, \gamma, i, \mathit{col})$, i.e., the extended sphere that we obtain by replacing the active node α with i.

The idea of the construction of the generalized 2NWA $\mathcal{B}_r$ is the following: Any state q of $\mathcal{B}_r$ is a set of extended spheres, which reflect the "environment" of a node that q is assigned to. Now suppose that, in a run of $\mathcal{B}_r$ on a nested word $\widetilde{W} = ([\widetilde{n}], \widetilde{\lessdot}, \widetilde{\mu}, \widetilde{\lambda})$, q is assigned to a position $i \in [\widetilde{n}]$ and contains $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col})$. If the run is accepting, this will mean that the environment of i in $\widetilde{W}$ looks like the environment of α in E. In particular, q will contain exactly one extended sphere $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col})$ such that γ and α coincide, meaning that $r\text{-Sph}(\widetilde{W}, i) \cong (N, \lessdot, \mu, \lambda, \gamma)$. Of course, $\mathcal{B}_r$ has to locally guess the environment of a position. But how can we ensure that a guess is correct? Obviously, we have to pass a local guess to any neighboring position in $\widetilde{W}$. So suppose again that a state q containing $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col})$ is assigned to a node i of $\widetilde{W}$. As α shall correspond to i, we need to ensure that $\lambda(\alpha) = \widetilde{\lambda}(i)$ (this will be taken care of by item (2) in the definition of the transition relation below). Now suppose that α has a $\lessdot$-successor $j \in N$, i.e., $\alpha \lessdot j$. Then, we have to guarantee that $i < \widetilde{n}$. This is done by simply excluding q from the set of final states. Moreover, j should correspond to $i+1$, which is ensured by passing $E[j]$ to the state that will be assigned to $i+1$ (see item (7)). On the other hand, if i comes up with a $\widetilde{\lessdot}$-successor, then α must have a $\lessdot$-successor j as well such that $E[j]$ belongs to the state that will be assigned to $i+1$. Observe that this rule applies unless $d_E(\gamma, \alpha) = r$, as then $i+1$ lies out of the area of responsibility of E (see item (5)). Similar requirements have to be considered wrt. potential $\lessdot$-/$\widetilde{\lessdot}$-predecessors (see (3), (4), and (6)), as well as wrt. the relations μ and $\widetilde{\mu}$ (see (3')–(7')). One difficulty in our construction, however, is to guarantee the lack of an edge. So assume the extended sphere E is the one given by Figure 2 with j_1 as the active node. Let us neglect colors for the moment. Suppose furthermore that $\widetilde{W}$ is the nested word from Figure 3. Then, an accepting run ρ of $\mathcal{B}_r$ on $\widetilde{W}$ will assign to i_1 a state that contains E (modulo some coloring). Moreover, the state assigned to i will contain $E[j]$, where the sphere center and the active node coincide. We observe that, in E, the node j_1 is maximal. In particular, there is no μ-edge between j_1 and j_2. This should be reflected in $\widetilde{W}$. A first idea to guarantee this might be to just prevent $\rho(i_2)$ from containing the extended sphere $E[j_2]$ (note that $(i_1, i_2) \in \widetilde{\mu}$). This is, however, too restrictive. Actually, $(r\text{-Sph}(\widetilde{W}, i), i_2)$ and $E[j_2]$ are isomorphic (neglecting the coloring of E) so that $\rho(i_2)$ must contain $E[j_2]$. The solution is already present in terms of the coloring of extended spheres. More precisely, $\rho(i_2)$ is allowed to carry $E[j_2]$ as soon as it has a color that is different from the color of the extended sphere $E[j_1]$ assigned to i_1. Roughly speaking, there might be isomorphic spheres in $\widetilde{W}$ that are overlapping. To consider them simultaneously, they are thus equipped with distinct colors.

The construction we obtain following the above ideas indeed allows us to infer, from an accepting run assigning a state q to a node i, the r-sphere around i. As mentioned above, we simply consider the (unique) extended sphere $(N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col})$ contained in q such that $\gamma = \alpha$. Then, $(N, \lessdot, \mu, \lambda, \gamma)$ is indeed the sphere of interest. It is not obvious that the above ideas really do work, all the less as the construction will apply to nested words over two stacks, but no longer to nested words over more than two stacks. After all, the key argument will be provided by Proposition 4.3, stating an important property of nested words over two stacks. Intuitively, it states the following: Suppose that, in a nested word, there is an acyclic path from a node i to another node i', and suppose this path is of a certain type w (recording the labelings and edges seen in the path). Then, applying the same path several times will never lead back to i. This is finally the reason why a cycle in an extended sphere that occurs in a run on a nested word $\widetilde{W}$ is in fact simulated by $\widetilde{W}$.

Let us formally construct the generalized 2NWA $\mathcal{B}_r = (Q, \delta, Q_I, F, C)$. An element of Q is a subset $\mathcal{E}$ of $\mathit{eSpheres}_r(\widetilde{\Sigma})$ such that either $\mathcal{E} = \emptyset$, which will be the only initial state, or the following conditions are satisfied:

(a) there is a unique extended sphere $(N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}$ such that $\gamma = \alpha$ (we set $\mathit{core}(\mathcal{E}) := (N, \lessdot, \mu, \lambda, \gamma)$),
(b) there is $a \in \Sigma$ such that, for any $(N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}$, $\lambda(\alpha) = a$ (so that we can assign a unique label a to $\mathcal{E}$, denoted by $\mathit{label}(\mathcal{E})$),
(c) for any two elements $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col})$ and $E' = (N', \lessdot', \mu', \lambda', \gamma', \alpha', \mathit{col}')$ from $\mathcal{E}$, if $\mathit{core}(E) = \mathit{core}(E')$ and $\mathit{col} = \mathit{col}'$, then $\alpha = \alpha'$.

So let us turn to the transition relation $\delta = \langle \delta_1, \delta_2 \rangle$:

- For $\mathcal{E}, \mathcal{E}' \in Q$ and $a \in \Sigma$, we let $(\mathcal{E}, a, \mathcal{E}') \in \delta_1$ if $\mathcal{E}' \neq \emptyset$ and the following hold:
  (1) for any $(N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}'$, $\alpha \notin \mathrm{dom}(\mu^{-1})$ (i.e., $\mu^{-1}(\alpha)$ is not defined)
  (2) $\mathit{label}(\mathcal{E}') = a$,
  (3) for any $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}$ and $i \in N$,
  $$E[i] \in \mathcal{E}' \implies (\alpha, i) \in \lessdot$$
  (4) for any $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}'$,
  $$\mathcal{E} \neq \emptyset \wedge \neg\exists i : (i, \alpha) \in \lessdot \implies d_E(\gamma, \alpha) = r$$
  (5) for any $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}$,
  $$\neg\exists i : (\alpha, i) \in \lessdot \implies d_E(\gamma, \alpha) = r$$
  (6) for any $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}'$ and $i \in N$,
  $$(i, \alpha) \in \lessdot \implies E[i] \in \mathcal{E}$$
  (7) for any $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}$ and $i \in N$,
  $$(\alpha, i) \in \lessdot \implies E[i] \in \mathcal{E}'$$

- For $\mathcal{E}_c, \mathcal{E}, \mathcal{E}' \in Q$ and $a \in \Sigma_r$, we let $(\mathcal{E}_c, \mathcal{E}, a, \mathcal{E}') \in \delta_2$ if $\mathcal{E}_c, \mathcal{E}, \mathcal{E}' \neq \emptyset$ and (2)–(7) as above hold as well as the following:
  (3') for any $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}_c$ and $i \in N$,
  $$E[i] \in \mathcal{E}' \implies (\alpha, i) \in \mu$$
  (4') for any $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}'$,
  $$\alpha \notin \mathrm{dom}(\mu^{-1}) \implies d_E(\gamma, \alpha) = r$$
  (5') for any $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}_c$,
  $$\alpha \notin \mathrm{dom}(\mu) \implies d_E(\gamma, \alpha) = r$$
  (6') for any $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}'$,
  $$\alpha \in \mathrm{dom}(\mu^{-1}) \implies E[\mu^{-1}(\alpha)] \in \mathcal{E}_c$$
  (7') for any $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}_c$ and $i \in N$,
  $$\alpha \in \mathrm{dom}(\mu) \implies E[\mu(\alpha)] \in \mathcal{E}'$$

As already mentioned, the only initial state of $\mathcal{B}_r$ is the empty set, i.e., $Q_I = \{\emptyset\}$. Moreover, $\mathcal{E} \in Q$ is a final state if, for any extended sphere $(N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}$, both $\alpha \notin \mathrm{dom}(\mu)$ and there is no $i \in N$ such that $(\alpha, i) \in \lessdot$. Finally, $\mathcal{E}$ is contained in C if there is $(N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \mathcal{E}$ such that $\alpha \in \mathrm{dom}(\mu)$.

The mapping $\eta : Q \to \mathit{Spheres}_r(\widetilde{\Sigma})$ as required in Proposition 4.1 is provided by *core*. More precisely, we set $\eta(\emptyset)$ to be some arbitrary sphere and $\eta(\mathcal{E}) = \mathit{core}(\mathcal{E})$ if $\mathcal{E} \neq \emptyset$.

4.2.2. *Any Nested Word Is Accepted.* Let $\widetilde{W} = ([\widetilde{n}], \widetilde{\lessdot}, \widetilde{\mu}, \widetilde{\lambda})$ be an arbitrary nested word over $\widetilde{\Sigma}$. We show that $\widetilde{W} \in \mathcal{L}(\mathcal{B}_r)$. Let us first distribute colors to each of the involved spheres. For this, we define the notion of an overlap: for any $i, i' \in [\widetilde{n}]$, i and i' are said to have an *r-overlap* in $\widetilde{W}$ if $r\text{-Sph}(\widetilde{W}, i) \cong r\text{-Sph}(\widetilde{W}, i')$ and $d_{\widetilde{W}}(i, i') \leq 2r + 1$. For example, in Figure 3, i and i' have a 2-overlap.

Claim 4.2. There is a mapping $\chi : [\widetilde{n}] \to [\#\mathit{Col}]$ such that, for any $i, i' \in [\widetilde{n}]$ with $i \neq i'$, the following holds: if i and i' have an r-overlap in $\widetilde{W}$, then $\chi(i) \neq \chi(i')$.

Proof. The mapping is obtained as a graph coloring. Consider the graph $([\widetilde{n}], \mathit{Arcs})$, $\mathit{Arcs} \subseteq [\widetilde{n}] \times [\widetilde{n}]$, where, for $i, i' \in [\widetilde{n}]$, we have $(i, i') \in \mathit{Arcs}$ iff $i \neq i'$ and i and i' have an r-overlap in $\widetilde{W}$. Clearly, $([\widetilde{n}], \mathit{Arcs})$ cannot be of degree greater than $4 \cdot \mathit{maxN}^2$ (for each $i \in [n]$, there are at most four distinct events i' such that $d_{\widetilde{W}}(i, i') \leq 1$). Hence, it can be *$\#\mathit{Col}$-colored* by a mapping $\chi : [\widetilde{n}] \to [\#\mathit{Col}]$ (i.e., $\chi(i) \neq \chi(i')$ for any $(i, i') \in \mathit{Arcs}$), which concludes the proof of Claim 4.2. □

We now specify $\rho : [\widetilde{n}] \to Q$: for $i \in [\widetilde{n}]$, we set $\rho(i) = \{(r\text{-Sph}(\widetilde{W}, i'), i, \chi(i')) \mid i' \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i, i') \leq r\}$. With this definition, we can check that, for any $i \in [\widetilde{n}]$, $\rho(i)$ is a valid state of $\mathcal{B}_r$, and that ρ is indeed an accepting run of $\mathcal{B}_r$ on $\widetilde{W}$. So let $i \in [\widetilde{n}]$ and let $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col})$ and $E' = (N', \lessdot', \mu', \lambda', \gamma', \alpha', \mathit{col}')$ be contained in $\rho(i)$.

(a) Assume that $\gamma = \alpha$ and $\gamma' = \alpha'$. Then, $(N, \lessdot, \mu, \lambda, \gamma, \gamma) \cong (r\text{-Sph}(\widetilde{W}, i), i)$ and $(N', \lessdot', \mu', \lambda', \gamma', \gamma') \cong (r\text{-Sph}(\widetilde{W}, i), i)$. Consequently, we have $(N, \lessdot, \mu, \lambda, \gamma, \gamma) \cong (N', \lessdot', \mu', \lambda', \gamma', \gamma')$. Moreover, $\mathit{col} = \mathit{col}' = \chi(i)$.
(b) Of course, $\lambda(\alpha) = \lambda'(\alpha')$.

(c) Assume $(N, \lessdot, \mu, \lambda, \gamma) \cong (N', \lessdot', \mu', \lambda', \gamma')$ and $col = col'$. There are $i_1, i_2 \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i, i_1) \leq r$, $d_{\widetilde{W}}(i, i_2) \leq r$, $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1), i)$, $(N, \lessdot, \mu, \lambda, \gamma, \alpha') \cong (r\text{-Sph}(\widetilde{W}, i_2), i)$, and $col = \chi(i_1) = \chi(i_2)$. Clearly, we have $r\text{-Sph}(\widetilde{W}, i_1) \cong r\text{-Sph}(\widetilde{W}, i_2)$. Furthermore, $i_1 = i_2$ and, therefore, $\alpha = \alpha'$. This is because i_1 and i_2 have an r-overlap in $\widetilde{W}$ so that, according to Claim 4.2, $i_1 \neq i_2$ would imply $\chi(i_1) \neq \chi(i_2)$, which contradicts the premise.

Now, for $i \in \{0, \ldots, \widetilde{n}\}$ and $i' = i + 1$ with $i' \notin \text{dom}(\widetilde{\mu}^{-1})$, we check that the triple $(\rho(i), \lambda(i'), \rho(i'))$ is contained in δ_1, where we let $\rho(0) = \emptyset$. Note first that, of course, $\rho(i') \neq \emptyset$.

(1) Suppose $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, col) \in \rho(i')$. We have $E \cong (r\text{-Sph}(\widetilde{W}, i''), i', \chi(i''))$ for some $i'' \in [\widetilde{n}]$ with $d_{\widetilde{W}}(i', i'') \leq r$. As $i' \notin \text{dom}(\widetilde{\mu}^{-1})$, we deduce $\alpha \notin \text{dom}(\mu^{-1})$.
(2) Obviously, we have $label(\rho(i')) = \widetilde{\lambda}(i')$,
(3) Suppose $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, col) \in \rho(i)$ (we thus have $i \geq 1$) and $j \in N$ such that $E[j] \in \rho(i')$. Recall that we have to show that, then, $(\alpha, j) \in \lessdot$. There are $i_1, i_1' \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i_1, i) \leq r$, $d_{\widetilde{W}}(i_1', i') \leq r$, $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1), i)$, $(N, \lessdot, \mu, \lambda, \gamma, j) \cong (r\text{-Sph}(\widetilde{W}, i_1'), i')$, and $col = \chi(i_1) = \chi(i_1')$. We easily see that i_1 and i_1' have an r-overlap in $\widetilde{W}$. We deduce, according to Claim 4.2, $i_1 = i_1'$. As, then, $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1), i)$, $(N, \lessdot, \mu, \lambda, \gamma, j) \cong (r\text{-Sph}(\widetilde{W}, i_1), i')$, and $(i, i') \in \widetilde{\lessdot}$, we can infer $(\alpha, j) \in \lessdot$.
(4) Let $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, col) \in \rho(i')$, suppose $i' \geq 2$, and suppose that there is no $j \in N$ such that $(j, \alpha) \in \lessdot$. Recall that we have to show that $d_E(\gamma, \alpha) = r$. There is $i_1' \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i_1', i') \leq r$ and $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1'), i')$. But if $d_E(\gamma, \alpha) < r$, then $d_{\widetilde{W}}(i_1', i') < r$, and there must be a $\lessdot$-predecessor of α, which is a contradiction. We therefore deduce that $d_E(\gamma, \alpha) = r$.
(5) Let $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, col) \in \rho(i)$ and suppose that there is no $j \in N$ such that $(\alpha, j) \in \lessdot$. Similarly to the case (4), we show that $d_E(\gamma, \alpha) = r$. In fact, there is $i_1 \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i_1, i) \leq r$ and $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1), i)$. Again, if $d_E(\gamma, \alpha) < r$, then $d_{\widetilde{W}}(i_1, i) < r$ so that there must be a $\lessdot$-successor of α, which is a contradiction. We conclude that $d_E(\gamma, \alpha) = r$.
(6) Let $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, col) \in \rho(i')$ and $j \in N$ such that $(j, \alpha) \in \lessdot$. We show that, then, $E[j] \in \rho(i)$. There is $i_1' \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i_1', i') \leq r$, $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1'), i')$, and $col = \chi(i_1')$. As $(j, \alpha) \in \lessdot$, α is not minimal so that we have $i \geq 1$. Since, furthermore, $d_E(\gamma, j) \leq r$ implies $d_{\widetilde{W}}(i_1', i) \leq r$, and since we have $(N, \lessdot, \mu, \lambda, \gamma, j) \cong (r\text{-Sph}(\widetilde{W}, i_1'), i)$ and $col = \chi(i_1')$, we deduce $E[j] = (N, \lessdot, \mu, \lambda, \gamma, j, col) \in \rho(i)$.
(7) Let $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, col) \in \rho(i)$ and $j \in N$ such that $(\alpha, j) \in \lessdot$. We have to show that $E[j] \in \rho(i')$. There is $i_1 \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i_1, i) \leq r$, $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1), i)$, and $col = \chi(i_1)$. Since $d_E(\gamma, j) \leq r$ implies $d_{\widetilde{W}}(i_1, i') \leq r$, and since we have $(N, \lessdot, \mu, \lambda, \gamma, j) \cong (r\text{-Sph}(\widetilde{W}, i_1), i')$ and $col = \chi(i_1)$, we deduce $E[j] = (N, \lessdot, \mu, \lambda, \gamma, j, col) \in \rho(i')$.

Next, for $i_c, i, i' \in [\widetilde{n}]$ with $i' = i + 1$ and $(i_c, i') \in \widetilde{\mu}$, we check that the quadruple $(\rho(i_c), \rho(i), \lambda(i'), \rho(i'))$ is contained in δ_2. Checking (2)–(7) proceeds as in the above cases.

For completeness, we present the cases (3')–(7'), which are shown analogously. First observe that, indeed, $\rho(i_c)$, $\rho(i)$, and $\rho(i')$ are all nonempty.

(3') Suppose $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \rho(i_c)$ and $j \in N$ such that $E[j] \in \rho(i')$. We show that $(\alpha, j) \in \mu$. There are $i_1, i_1' \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i_1, i_c) \leq r$, $d_{\widetilde{W}}(i_1', i') \leq r$, $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1), i_c)$, $(N, \lessdot, \mu, \lambda, \gamma, j) \cong (r\text{-Sph}(\widetilde{W}, i_1'), i')$, and $\mathit{col} = \chi(i_1) = \chi(i_1')$. Again, i_1 and i_1' have an r-overlap in $\widetilde{W}$. According to Claim 4.2, $i_1 = i_1'$. Then, $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1), i_c)$, $(N, \lessdot, \mu, \lambda, \gamma, j) \cong (r\text{-Sph}(\widetilde{W}, i_1), i')$, and $(i_c, i') \in \widetilde{\mu}$, so that we can deduce $(\alpha, j) \in \mu$.

(4') Let $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \rho(i')$ and suppose that there is no $j \in N$ such that $(j, \alpha) \in \mu$. We have to show that $d_E(\gamma, \alpha) = r$. There is $i_1' \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i_1', i') \leq r$ and $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1'), i')$. But if $d_E(\gamma, \alpha) < r$, then $d_{\widetilde{W}}(i_1', i') < r$, so there must be a μ-predecessor of α, which is a contradiction. We deduce $d_E(\gamma, \alpha) = r$.

(5') Let $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \rho(i_c)$ and suppose that there is no $j \in N$ such that $(\alpha, j) \in \mu$. We show that, then, $d_E(\gamma, \alpha) = r$. There is $i_1 \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i_1, i_c) \leq r$ and $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1), i_c)$. If $d_E(\gamma, \alpha) < r$, then $d_{\widetilde{W}}(i_1, i_c) < r$, so there must be a μ-successor of α, which is a contradiction. We conclude that $d_E(\gamma, \alpha) = r$.

(6') Let $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \rho(i')$ and $j \in N$ such that $(j, \alpha) \in \mu$. We show $E[j] \in \rho(i_c)$. There is $i_1' \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i_1', i') \leq r$, $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1'), i')$, and $\mathit{col} = \chi(i_1')$. Due to $d_E(\gamma, j) \leq r$, we also have $d_{\widetilde{W}}(i_1', i_c) \leq r$, and since $(N, \lessdot, \mu, \lambda, \gamma, j) \cong (r\text{-Sph}(\widetilde{W}, i_1'), i_c)$ and $\mathit{col} = \chi(i_1')$, we deduce $E[j] \in \rho(i_c)$.

(7') Let $E = (N, \lessdot, \mu, \lambda, \gamma, \alpha, \mathit{col}) \in \rho(i_c)$ and $j \in N$ such that $(\alpha, j) \in \mu$. We have to show $E[j] \in \rho(i')$. There is $i_1 \in [\widetilde{n}]$ such that $d_{\widetilde{W}}(i_1, i_c) \leq r$, $(N, \lessdot, \mu, \lambda, \gamma, \alpha) \cong (r\text{-Sph}(\widetilde{W}, i_1), i_c)$, and $\mathit{col} = \chi(i_1)$. From $d_E(\gamma, j) \leq r$, it follows $d_{\widetilde{W}}(i_1, i') \leq r$. As, moreover, $(N, \lessdot, \mu, \lambda, \gamma, j) \cong (r\text{-Sph}(\widetilde{W}, i_1), i')$ and $\mathit{col} = \chi(i_1)$, we deduce $E[j] = (N, \lessdot, \mu, \lambda, \gamma, j, \mathit{col}) \in \rho(i')$.

4.2.3. *Any Run Keeps Track Of Spheres.* We will now show that an accepting run reveals the sphere around any node. This constitutes the more difficult part of the correctness proof.

We introduce some useful notation: By Δ, we denote the set $\{\rightarrow, \leftarrow, \overset{1}{\curvearrowright}, \overset{1}{\curvearrowleft}, \overset{2}{\curvearrowright}, \overset{2}{\curvearrowleft}\}$ of *directions*. Now let $W = ([n], \lessdot, \mu, \lambda) \in \mathbb{NW}(\widetilde{\Sigma})$ be a nested word, $i, j \in [n]$, and let $w = e_1 \dots e_m \in \Delta^*$ (where $e_k \in \Delta$ for any $k \in \{1, \dots, m\}$). We write $i \overset{w}{\Longrightarrow}_W j$ if there are $i_0, i_1, \dots, i_m \in [n]$ such that $i_0 = i$, $i_m = j$, and, for any $k \in \{0, \dots, m-1\}$, one of the following holds:

(a) $e_{k+1} = \rightarrow$ and $i_{k+1} = i_k + 1$
(b) $e_{k+1} = \leftarrow$ and $i_{k+1} = i_k - 1$
(c) $e_{k+1} = \overset{s}{\curvearrowright}$ and $i_k \in \mathrm{dom}(\mu)$ and $\lambda(i_k) \in \Sigma_c^s$ and $i_{k+1} = \mu(i_k)$ (for some $s \in \{1, 2\}$)
(d) $e_{k+1} = \overset{s}{\curvearrowleft}$ and $i_k \in \mathrm{dom}(\mu^{-1})$ and $\lambda(i_k) \in \Sigma_r^s$, and $i_{k+1} = \mu^{-1}(i_k)$ (for some $s \in \{1, 2\}$)

Moreover, we write $i \xhookrightarrow{w}_W j$ if there are pairwise distinct $i_0, i_1, \ldots, i_{m-1} \in [n]$ and $i_m \in [n] \setminus \{i_1, \ldots, i_{m-1}\}$ such that $i_0 = i$, $i_m = j$, and, for any $k \in \{0, \ldots, m-1\}$, (a)–(d) as above hold.

The following proposition is crucial for our project, and it fails when considering nested words over more than two stacks.

Proposition 4.3. Let $W = ([n], \lessdot, \mu, \lambda) \in \mathbb{NW}(\widetilde{\Sigma})$, $w \in \Delta^+$, and $i, i' \in [n]$ such that $i \xhookrightarrow{w}_W i'$. If there is $k \geq 1$ such that $i \xhookrightarrow{w^k}_W i$, then $i = i'$.

Before we prove Proposition 4.3, note that it does not hold as soon as a third stack comes into play. To see this, consider Figure 6, describing part of a nested word W over the 3-stack call-return alphabet $\langle\{(\{a\},\{\overline{a}\}),(\{b\},\{\overline{b}\}),(\{c\},\{\overline{c}\})\},\emptyset\rangle$. For $w = \overset{1}{\curvearrowright}\leftarrow\overset{2}{\curvearrowleft}\leftarrow\overset{3}{\curvearrowleft}\leftarrow$, we have $i \xhookrightarrow{w}_W i'$ and $i \xhookrightarrow{ww}_W i$ (where the meaning of $\overset{3}{\curvearrowleft}$ is the expected one). However, $i \neq i'$.

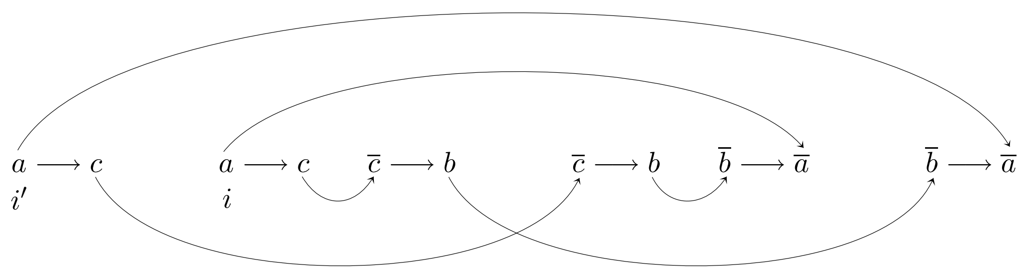

Figure 6: Proposition 4.3 fails when considering three stacks

Moreover, it is crucial to require, in the above definition of $i \xhookrightarrow{w}_W j$, the elements $i_0, i_1, \ldots, i_{m-1} \in [n]$ to be pairwise distinct. This can be seen considering a part of the nested word W over the 2-stack call-return alphabet $\langle\{(\{a\},\{\overline{a}\}),(\{b\},\{\overline{b}\})\},\emptyset\rangle$ that is depicted in Figure 7. Let $w = \overset{1}{\curvearrowright}\leftarrow\leftarrow\overset{1}{\curvearrowleft}\leftarrow\overset{2}{\curvearrowleft}\leftarrow$. We have $i \overset{ww}{\Longrightarrow}_W i$, i.e., starting from i, we can follow the sequence of directions w twice, arriving at i again. However, apart from i, we have to visit j_1 and j_2 twice. Indeed, $i \xhookrightarrow{w}_W i'$, but also $i \overset{ww}{\not\hookrightarrow}_W i$.

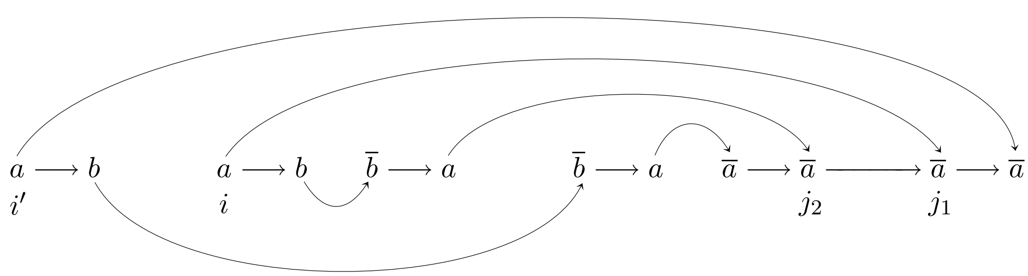

Figure 7: Intermediate positions need to be pairwise distinct

Proof (of Proposition 4.3). Let $W = ([n], \lessdot, \mu, \lambda) \in \mathbb{NW}(\widetilde{\Sigma})$, $w \in \Delta^+$, and $i \in [n]$. We have to show that, if $i \overset{w}{\hookrightarrow}_W i$, then w cannot be decomposed nontrivially into identical factors, i.e., there is no $u \in \Delta^+$ such that $w = u^k$ for some $k \geq 2$. To see this easily, we observe that a situation such as $i \overset{w}{\hookrightarrow}_W i$ corresponds to a topological circle, as depicted in Figure 8. A topological circle is a closed line in the two-dimensional plane that never crosses over itself. Let us construct topological circles according to the following procedure: We assume a straight (horizontal) line of the plane. Assume further a point i on this line. Starting from i, we choose another two points as follows: Pick a symbol γ from the alphabet $\{\overset{1}{\curvearrowright}, \overset{1}{\curvearrowleft}\} \cdot \{\overset{2}{\curvearrowright}, \overset{2}{\curvearrowleft}\}$. According to this choice, we first draw a semicircle above the straight line ending somewhere on the line, and then, without interruption, a semicircle below the line, again resulting in a point on the line. Each semicircle is drawn in the direction indicated by γ, e.g., $\overset{1}{\curvearrowright}\overset{2}{\curvearrowleft}$ requires to draw the upper semicircle from left to right and the lower one from right to left. This procedure is continued until we reach the original point i. Observe that, in Figure 8, we follow the sequence $w = (\overset{1}{\curvearrowright}\overset{2}{\curvearrowright})(\overset{1}{\curvearrowright}\overset{2}{\curvearrowright})(\overset{1}{\curvearrowleft}\overset{2}{\curvearrowleft})(\overset{1}{\curvearrowright}\overset{2}{\curvearrowleft})(\overset{1}{\curvearrowleft}\overset{2}{\curvearrowleft})(\overset{1}{\curvearrowleft}\overset{2}{\curvearrowleft})$, starting in the left outermost point of intersection on the horizontal line. Also note that we have $w \neq u^k$ for any $u \in (\{\overset{1}{\curvearrowright}, \overset{1}{\curvearrowleft}\} \cdot \{\overset{2}{\curvearrowright}, \overset{2}{\curvearrowleft}\})^+$ and $k \geq 2$. It is not hard to see that topological circles behave aperiodically in general.

So suppose that $w \in (\{\overset{1}{\curvearrowright}, \overset{1}{\curvearrowleft}\} \cdot \{\overset{2}{\curvearrowright}, \overset{2}{\curvearrowleft}\})^+$ is of the form $u_1(\overset{1}{\curvearrowright}\overset{2}{\curvearrowleft})u_2$ (in the other cases, we follow similar arguments). We have to show that following a sequence w^k such that $k \geq 2$ will never produce a topological circle. As in any repetition of $u_1(\overset{1}{\curvearrowright}\overset{2}{\curvearrowleft})u_2$, the segments belonging to u_1 must not intersect and those belonging to u_2 cannot intersect either, this reduces to the problem if the repetition of the atomic sequence $\overset{1}{\curvearrowright}\overset{2}{\curvearrowleft}$ may yield a topological circle. But obviously, for $k \geq 2$, $(\overset{1}{\curvearrowright}\overset{2}{\curvearrowleft})^k$ gives rise to a "spiral", and going back to the starting point would require to intersect the line that has been drawn hitherto.

To summarize, following a sequence from the set $(\{\overset{1}{\curvearrowright}, \overset{1}{\curvearrowleft}\} \cdot \{\overset{2}{\curvearrowright}, \overset{2}{\curvearrowleft}\})^+$ several times (i.e., at least twice) can never produce a topological circle. It is easy to translate this fact into the nested-word setting over two stacks meaning that $i \overset{w}{\hookrightarrow}_W i$ implies that w cannot be decomposed nontrivially into equal factors. $\square$

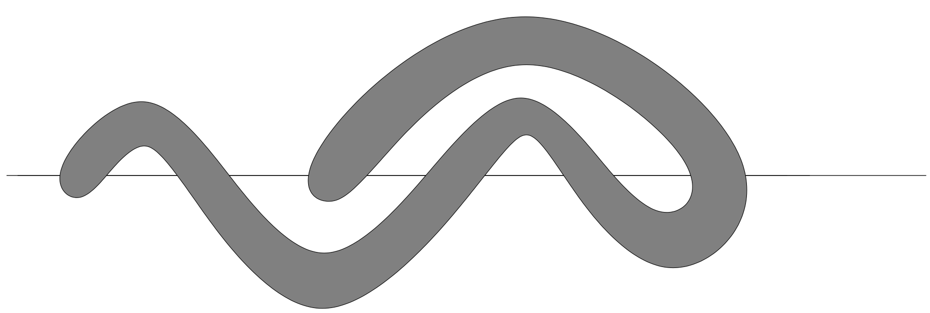

Figure 8: Proof of Proposition 4.3

We will now show that, indeed, $\mathcal{B}_r$ discovers the r-sphere around any node of an input nested word.

Let $W = ([n], \lessdot, \mu, \lambda) \in \mathbb{NW}(\widetilde{\Sigma})$ be a nested word and ρ be a run of $\mathcal{B}_r$ on W. Consider any $i \in [n]$, let $(N_i, \lessdot_i, \mu_i, \lambda_i, \gamma_i)$ refer to $\mathit{core}(\rho(i))$, and let col_i be the unique element from $[\#\mathit{Col}]$ satisfying $E_i := (N_i, \lessdot_i, \mu_i, \lambda_i, \gamma_i, \gamma_i, \mathit{col}_i) \in \rho(i)$.

The following statement claims that an arbitrarily long path in E_i is simulated by a corresponding path in W.

Claim 4.4. Let $d \geq 0$ and suppose there are $j_0, \ldots, j_d \in N_i$ such that $\gamma_i = j_0 \leftrightarrow_{E_i} j_1 \leftrightarrow_{E_i} \ldots \leftrightarrow_{E_i} j_d$. Then, there is a (unique) sequence of nodes $i_0, \ldots, i_d \in [n]$ such that

- $i_0 = i$,
- for each $k \in \{0, \ldots, d\}$, $E_i[j_k] \in \rho(i_k)$ (in particular, $\lambda(i_k) = \lambda_i(j_k)$), and
- for any $k \in \{0, \ldots, d-1\}$, $(j_k, j_{k+1}) \sqsubseteq_W^{E_i} (i_k, i_{k+1})$.

Proof. The proof is by induction. Obviously, the statement holds for $d = 0$. So assume $d \geq 0$ and suppose there are a sequence $j_0, \ldots, j_d, j_{d+1} \in N_i$ such that $\gamma_i = j_0 \leftrightarrow_{E_i} j_1 \leftrightarrow_{E_i} \ldots \leftrightarrow_{E_i} j_d \leftrightarrow_{E_i} j_{d+1}$ and a unique sequence $i_0, i_1, \ldots, i_d \in [n]$ such that $i_0 = i$, $E_i[j_k] \in \rho(i)$ for each $k \in \{0, \ldots, d\}$, and $(j_k, j_{k+1}) \sqsubseteq_W^{E_i} (i_k, i_{k+1})$ for any $k \in \{0, \ldots, d-1\}$. We consider four cases:

- Assume $(j_d, j_{d+1}) \in \lessdot_i$. Then, $\rho(i_d)$ is not a final state so that $i_d < n$. We set $i_{d+1} = i_d + 1$. Due to (7), we have $E_i[j_{d+1}] \in \rho(i_{d+1})$.
- Assume $(j_{d+1}, j_d) \in \lessdot_i$. Then, according to (6), $i_d \geq 2$. We set $i_{d+1} = i_d - 1$. Due to (6), we also have $E_i[j_{d+1}] \in \rho(i_{d+1})$.
- Assume $(j_d, j_{d+1}) \in \mu_i$. Clearly, $\rho(i_d)$ is a calling state so that $\mu(i_d)$ is defined. Setting $i_{d+1} = \mu(i_d)$, we have, due to (7'), $E_i[j_{d+1}] \in \rho(i_{d+1})$.
- Assume $(j_{d+1}, j_d) \in \mu_i$. According to (1), $i_d \in \mathrm{dom}(\mu^{-1})$. With (6'), letting $i_{d+1} = \mu^{-1}(i_d)$, we have $E_i[j_{d+1}] \in \rho(i_{d+1})$.

This concludes the proof of Claim 4.4. □

Claim 4.5. There is a homomorphism $h : r\text{-Sph}(W, i) \to \mathit{core}(\rho(i))$.

Proof. We show by induction the following statement:

> For any $d \in \{0, \ldots, r\}$, there is a homomorphism $h : d\text{-Sph}(W, i) \to d\text{-Sph}((N_i, \lessdot_i, \mu_i, \lambda_i), \gamma_i)$ such that, for any $i' \in [n]$ with $d_W(i, i') \leq d$, we have $E_i[h(i')] \in \rho(i')$. (*)

Of course, (*) holds for $d = 0$. So assume that (*) holds true for some natural number $d \in \{0, \ldots, r-1\}$, i.e., there is a homomorphism $h : d\text{-Sph}(W, i) \to d\text{-Sph}((N_i, \lessdot_i, \mu_i, \lambda_i), \gamma_i)$ such that $E_i[h(i')] \in \rho(i')$ for any $i' \in [n]$ with $d_W(i, i') \leq d$. We show that then (*) holds for $d + 1$ as well. For this, let $i_1, i_2 \in [n]$ such that $d_W(i, i_1) = d$ and $d_W(i, i_2) = d + 1$.

- Suppose $i_1 \lessdot i_2$. Since $d_W(i, i_1) < r$, we also have $d_{E_i}(\gamma_i, h(i_1)) < r$. Due to (5), there is $j_2 \in N_i$ such that $h(i_1) \lessdot_i j_2$. Since $E_i[h(i_1)] \in \rho(i_1)$, we obtain, by (7) and (2), that $\lambda_i(j_2) = \lambda(i_2)$ and $E_i[j_2] \in \rho(i_2)$.
- Similarly, we proceed if $i_2 \lessdot i_1$. By $d_{E_i}(\gamma_i, h(i_1)) < r$ and (4), there is $j_2 \in N_i$ such that $j_2 \lessdot_i h(i_1)$. Since $E_i[h(i_1)] \in \rho(i_1)$, we obtain, by (6) and (2), that $\lambda_i(j_2) = \lambda(i_2)$ and $E_i[j_2] \in \rho(i_2)$.
- If $(i_1, i_2) \in \mu$, then there exists, exploiting (5') and (7'), $j_2 \in N_i$ such that $(h(i_1), j_2) \in \mu_i$, $\lambda_i(j_2) = \lambda(i_2)$, and $E_i[j_2] \in \rho(i_2)$.
- If $(i_2, i_1) \in \mu$, then we can find, due to (4') and (6'), $j_2 \in N_i$ such that $(j_2, h(i_1)) \in \mu_i$, $\lambda_i(j_2) = \lambda(i_2)$, and $E_i[j_2] \in \rho(i_2)$.

Observe that j_2 is uniquely determined by i_2 and does not depend on the choice of i_1 or on the relation between i_1 and i_2: If we obtained distinct elements j_2 and j_2', then the constraints $E_i[j_2] \in \rho(i_2)$ and $E_i[j_2'] \in \rho(i_2)$ would imply that $\rho(i_2)$ is not a valid state.

The above procedure extends the domain of the homomorphism h by those elements whose distance to i is $d+1$. I.e., for $i_1, i_2 \in [n]$ with $d_W(i, i_1) = d_W(i, i_2) = d+1$, we determined two *unique* elements $h(i_1), h(i_2) \in N_i$, respectively. Let us show that $(i_1, i_2) \sqsubseteq^W_{core(\rho(i))} (h(i_1), h(i_2))$. Suppose $i_1 \lessdot i_2$ (the case $i_2 \lessdot i_1$ is symmetric). As $E_i[h(i_1)] \in \rho(i_1)$ and $E_i[h(i_2)] \in \rho(i_2)$, we have, by (3), $h(i_1) \lessdot_i h(i_2)$. Similarly, with (3'), $(i_1, i_2) \in \mu$ implies $(h(i_1), h(i_2)) \in \mu_i$. ☐

Claim 4.6. There is a homomorphism $h' : core(\rho(i)) \to r\text{-Sph}(W, i)$.

Proof. We show, again by induction, the following statement:

> For any natural number $d \in \{0, \ldots, r\}$, there is a homomorphism h' : $d\text{-Sph}((N_i, \lessdot_i, \mu_i, \lambda_i), \gamma_i) \to d\text{-Sph}(W, i)$ such that, for any $j \in N_i$ with $d_{E_i}(\gamma_i, j) \leq d$, we have $E_i[j] \in \rho(h'(j))$. (**)

Clearly, (**) holds for $d = 0$. Assume that (**) holds for some natural number $d \in \{0, \ldots, r-1\}$ and let $h' : d\text{-Sph}((N_i, \lessdot_i, \mu_i, \lambda_i), \gamma_i) \to d\text{-Sph}(W, i)$ be a corresponding homomorphism. Let $j_1, j_2 \in N_i$ such that $d_{E_i}(\gamma_i, j_1) = d$ and $d_{E_i}(\gamma_i, j_2) = d+1$.

Suppose that $j_1 \lessdot_i j_2$. As $E_i[j_1] \in \rho(h'(j_1))$, $\rho(h'(j_1))$ cannot be a final state of $\mathcal{B}_r$ so that there is $i_2 \in [n]$ such that $h'(j_1) \lessdot i_2$. Clearly, we have $E_i[j_2] \in \rho(i_2)$. Analogously, we proceed in the cases $j_2 \lessdot_i j_1$, $(j_1, j_2) \in \mu_i$, and $(j_2, j_1) \in \mu_i$ to obtain such an element i_2. Note that i_2 is uniquely determined by j_2 and does not depend on the choice of j_1 or on the specific relation between j_1 and j_2. This is less obvious than the corresponding fact in the proof of Claim 4.5 but can be shown along the lines of the following procedure, proving that the extension of the domain of h' by elements $j \in N_i$ with $d_{E_i}(\gamma_i, j) = d+1$ is a homomorphism:

We show that, for $j, j' \in N_i$ with $d_{E_i}(\gamma_i, j) = d_{E_i}(\gamma_i, j') = d+1$, we have $(j, j') \sqsubseteq^{E_i}_W (h'(j), h'(j'))$ (where the elements $h'(j)$ and $h'(j')$ are obtained as indicated above). So suppose $j \leftrightarrow_{E_i} j'$. There are $\ell \in \{0, \ldots, d\}$ and pairwise distinct $j_0, \ldots, j_{2(d+1)-\ell} \in N_i$, such that

$$\begin{tikzcd}
 & & & j_{\ell+1} \arrow[r, leftrightarrow, "E_i"] & \cdots \arrow[r, leftrightarrow, "E_i"] & j_{d+1} \arrow[d, leftrightarrow, "E_i"] & = j \\
\gamma_i = j_0 \arrow[r, leftrightarrow, "E_i"] & \cdots \arrow[r, leftrightarrow, "E_i"] & j_\ell \arrow[ur, leftrightarrow, "E_i"] \arrow[dr, leftrightarrow, "E_i"'] & & & & \\
 & & & j_{2(d+1)-\ell} \arrow[r, leftrightarrow, "E_i"] & \cdots \arrow[r, leftrightarrow, "E_i"] & j_{d+2} & = j'
\end{tikzcd}$$

For ease of notation, set $D = 2(d+1) - \ell$ and let, for $k \in \mathbb{N}$,

$$mod(k) = \begin{cases} k & \text{if } k \leq D \\ ((k-\ell) \bmod (D - \ell + 1)) + \ell & \text{if } k > D \end{cases}$$

I.e., the mapping *mod* counts until D and afterwards modulo $D - \ell + 1$. According to Claim 4.4, there is a unique infinite sequence $i_0, i_1, \ldots \in [n]$ such that

- $i_0 = i$,
- for any $k \in \mathbb{N}$, $E_i[j_{mod(k)}] \in \rho(i_k)$, and
- for any $k \in \mathbb{N}$, $(j_{mod(k)}, j_{mod(k+1)}) \sqsubseteq^{E_i}_W (i_k, i_{k+1})$.

In what follows, we show that $i_{D+1} = i_\ell$, which implies $(j_{d+1}, j_{d+2}) \sqsubseteq_W^{E_i} (i_{d+1}, i_{d+2})$ so that $(j_{d+1}, j_{d+2}) \sqsubseteq_W^{E_i} (h'(j_{d+1}), h'(j_{d+2}))$. There is $w = e_\ell \dots e_D \in \Delta^+$ such that

- $j_\ell \xRightarrow{w}_{E_i} j_\ell$,
- $j_\ell \xRightarrow{e_\ell \dots e_{\ell+k-1}}_{E_i} j_{\ell+k}$ for any $k \in \{1, \dots, D-\ell\}$, and
- $i_\ell \xRightarrow{w^k}_W i_{\ell+k(D-\ell+1)}$ for any $k \geq 1$.

We can obtain such a w by setting, for any $k \in \{\ell, \dots, D\}$,

$$e_k = \begin{cases} \rightarrow & \text{if } j_k \lessdot_i j_{mod(k+1)} \\ \leftarrow & \text{if } j_{mod(k+1)} \lessdot_i j_k \\ \overset{1}{\curvearrowright} & \text{if } \lambda_i(j_k) \in \Sigma_c^1 \text{ and } (j_k, j_{mod(k+1)}) \in \mu_i \text{ and } j_k \not\lessdot_i j_{mod(k+1)} \\ \overset{1}{\curvearrowleft} & \text{if } \lambda_i(j_k) \in \Sigma_r^1 \text{ and } (j_{mod(k+1)}, j_k) \in \mu_i \text{ and } j_{mod(k+1)} \not\lessdot_i j_k \\ \overset{2}{\curvearrowright} & \text{if } \lambda_i(j_k) \in \Sigma_c^2 \text{ and } (j_k, j_{mod(k+1)}) \in \mu_i \text{ and } j_k \not\lessdot_i j_{mod(k+1)} \\ \overset{2}{\curvearrowleft} & \text{if } \lambda_i(j_k) \in \Sigma_r^2 \text{ and } (j_{mod(k+1)}, j_k) \in \mu_i \text{ and } j_{mod(k+1)} \not\lessdot_i j_k \end{cases}$$

As $[n]$ is a finite set[3], there are $p, q \in \mathbb{N}$ such that $\ell \leq p < q$ and $i_p = i_q$. We choose p and q such that $i_\ell, \dots, i_{q-1}$ are pairwise distinct. We have both $E_i[j_{mod(p)}] \in \rho(i_p)$ and $E_i[j_{mod(q)}] \in \rho(i_p)$. According to the definition of the set of states of $\mathcal{B}_r$, this implies $j_{mod(p)} = j_{mod(q)}$. Let us distinguish three cases:

Case 1: $p = \ell$ and $q = \ell + k(D - \ell + 1)$ for some $k \geq 1$. Then, $i_\ell \overset{w^k}{\hookrightarrow}_W i_{\ell+k(D-\ell+1)}$ so that, according to Proposition 4.3, we have $i_\ell = i_{D+1}$, and we are done.

Case 2: $p > \ell$ and $q = p + k(D - \ell + 1)$ for some $k \geq 1$. Setting $e = e_{mod(p-1)}$, we have both $i_{p-1} \overset{e}{\hookrightarrow}_W i_p$ and $i_{q-1} \overset{e}{\hookrightarrow}_W i_p$, which is a contradiction, as $i_{p-1} \neq i_{q-1}$.

Case 3: $p \geq \ell$ and $q \neq p+k(D-\ell+1)$ for any $k \geq 1$. But this implies $mod(p) \neq mod(q)$ and, as the $j_\ell, \dots, j_D$ are pairwise distinct, $j_{mod(p)} \neq j_{mod(q)}$, a contradiction.

This concludes the proof of Claim 4.6. □

So let $h : r\text{-Sph}(W, i) \rightarrow core(\rho(i))$ and $h' : core(\rho(i)) \rightarrow r\text{-Sph}(W, i)$ be the unique homomorphisms that we obtain following the constructive proofs of Claims 4.5 and 4.6, respectively. It is now immediate that h is injective, $h^{-1} = h'$, and $h : r\text{-Sph}(W, i) \rightarrow core(\rho(i))$ is an isomorphism.

Recall that $\eta : Q \rightarrow Spheres_r(\widetilde{\Sigma})$ shall map the empty set to an arbitrary sphere and a nonempty set $\mathcal{E} \in Q$ onto $core(\mathcal{E})$. Indeed, we constructed a generalized 2NWA $\mathcal{B}_r = (Q, \delta, Q_I, F, C)$ together with a mapping $\eta : Q \rightarrow Spheres_r(\widetilde{\Sigma})$ such that

- $\mathcal{L}(\mathcal{B}_r)$ is the set of all nested words over $\widetilde{\Sigma}$ (cf. Section 4.2.2), and
- for any nested word $W \in \mathbb{NW}(\widetilde{\Sigma})$, for any accepting run ρ of $\mathcal{B}_r$ on W, and for any node i of W, we have $\eta(\rho(i)) \cong r\text{-Sph}(W, i)$ (cf. Section 4.2.3).

This shows Proposition 4.1.

[3]In the context of infinite nested words, this argument can be replaced with the fact that, starting in i, there is no infinite sequence of *pairwise distinct* nodes that follows the infinite sequence of directions w^ω, i.e., the infinite repetition of w (see Section 7).

5. 2-Stack Visibly Pushdown Automata vs. Logic

Proposition 4.1, which constitutes the key result from Section 4, can now be used to establish expressive equivalence of 2VPA and EMSO logic. In this section, we again fix a 2-stack call-return alphabet $\widetilde{\Sigma} = \langle \{(\Sigma_c^1, \Sigma_r^1), (\Sigma_c^2, \Sigma_r^2)\}, \Sigma_{int} \rangle$.

Lemma 5.1. Let $r, t \in \mathbb{N}$ and let $S \in \mathit{Spheres}_r(\widetilde{\Sigma})$ be an r-sphere in some nested word over $\widetilde{\Sigma}$. There are generalized 2NWA $\mathcal{B}^1$ and $\mathcal{B}^2$ over $\widetilde{\Sigma}$ such that $\mathcal{L}(\mathcal{B}^1) = \{W \in \mathbb{NW}(\widetilde{\Sigma}) \mid |W|_S = t\}$ and $\mathcal{L}(\mathcal{B}^2) = \{W \in \mathbb{NW}(\widetilde{\Sigma}) \mid |W|_S > t\}$.

Proof. In both cases, we start from the generalized 2NWA $\mathcal{B}_r = (Q, \delta, Q_I, F, C)$ and the mapping $\eta : Q \to \mathit{Spheres}_r(\widetilde{\Sigma})$ from Proposition 4.1. For $k = 1, 2$, we obtain $\mathcal{B}^k$ by extending the state space with a counter that, using η, counts the number of realizations of S up to $t + 1$. The new set of initial states is thus in both cases $Q_I \times \{0\}$. However, the set of final states of $\mathcal{B}^1$ is $F \times \{t\}$, the one of $\mathcal{B}^2$ is $F \times \{t + 1\}$. $\square$

We are now prepared to state the first main result of this paper.

Theorem 5.2. *Let $\mathcal{L}$ be a set of nested words over the 2-stack call-return alphabet $\widetilde{\Sigma}$. Then, the following are equivalent:*

(1) *There is a 2VPA $\mathcal{A}$ over $\widetilde{\Sigma}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.*
(2) *There is a sentence $\varphi \in \mathrm{EMSO}(\tau_{\widetilde{\Sigma}})$ such that $\mathcal{L}(\varphi) = \mathcal{L}$.*

Proof. To prove (1) $\to$ (2), one can perform a standard construction of an EMSO formula from a 2NWA, where the latter can be extracted from the given 2VPA according to Lemma 2.8. Basically, the formula "guesses" a possible run on the input word in terms of existentially quantified second-order variables and then verifies, in its first-order fragment, that we actually deal with a run that is accepting.

So let us directly prove (2) $\to$ (1) and let $\varphi = \exists X_1 \ldots \exists X_m \psi \in \mathrm{EMSO}(\tau_{\widetilde{\Sigma}})$ be a sentence with $\psi \in \mathrm{FO}(\tau_{\widetilde{\Sigma}})$ (we suppose $m \geq 1$). We define a new 2-stack call-return alphabet

$$\widehat{\Sigma} = \langle \{(\Sigma_c^1 \times 2^m, \Sigma_r^1 \times 2^m), (\Sigma_c^2 \times 2^m, \Sigma_r^2 \times 2^m)\}, \Sigma_{int} \times 2^m \rangle$$

where 2^m shall denote the powerset of $[m]$. From ψ, we obtain an FO formula ψ' over $\tau_{\widehat{\Sigma}}$ by replacing any occurrence of $\lambda(x) = a$ with $\bigvee_{M \in 2^m} \lambda(x) = (a, M)$ and any occurrence of $x \in X_k$ with $\bigvee_{a \in \Sigma,\ M \in 2^m} \lambda(x) = (a, M \cup \{k\})$. We set $\mathcal{L} \subseteq \mathbb{NW}(\widehat{\Sigma})$ to be the set of nested words that satisfy ψ'. From Hanf's Theorem (Theorem 3.3), we know that $\mathcal{L}$ is a finite union of $\leftrightarrows_{r,t}$ equivalence classes for suitable r and t. It is easy to see that the class of nested-word languages that are recognized by generalized 2NWA is closed under union and intersection. Thus, by Lemma 5.1, there is a generalized 2NWA $\mathcal{B}'$ recognizing $\mathcal{L}$.

Now, to check whether some nested word from $\mathbb{NW}(\widetilde{\Sigma})$ satisfies φ, a generalized 2NWA $\mathcal{B}$ with $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\varphi)$ will guess an additional labeling for any node in terms of an element from 2^m and then simulate $\mathcal{B}'$. By Lemma 2.7 and Lemma 2.8, we finally obtain a 2VPA $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\varphi)$. $\square$

Observe that the number of states of the 2VPA that we construct for a given EMSO sentence is elementary in the size of the formula. However, we do not know if it can be computed in elementary time or if it can be computed effectively at all. For this, one has to check if a representative of an equivalence class of $\leftrightarrows_{r,t}$ can be constructed algorithmically.

6. Grids and Monadic Second-Order Quantifier Alternation

In this section, we show that the monadic second-order quantifier-alternation hierarchy over nested words is infinite. In other words, the more alternation of second-order quantification we allow, the more expressive formulas become. From this, we can finally deduce that 2-stack visibly pushdown automata cannot be complemented in general. In the proof, we use results that have been gained in the setting of grids. By means of first-order reductions from grids into nested words, we can indeed transfer expressiveness results for grids to the nested-word setting. Let us first state a general result from [12], starting with the formal definition of a strong first-order reduction:

Definition 6.1 ([12], Definition 32)**.** Let $\mathcal{C}$ and $\mathcal{C}'$ be classes of structures over relational signatures τ and τ', respectively. A *strong first-order reduction* from $\mathcal{C}$ to $\mathcal{C}'$ with rank $m \geq 1$ is an injective mapping $\Phi : \mathcal{C} \to \mathcal{C}'$ such that the following hold:

(1) For any $G \in \mathcal{C}$, the universe of $\Phi(G)$ is $\bigcup_{k\in\{1,\ldots,m\}}(\{k\} \times \mathrm{dom}(G))$, i.e., the disjoint union of m copies of $\mathrm{dom}(G)$, where $\mathrm{dom}(G)$ shall denote the universe of G.
(2) There is $\psi(x_1, \ldots, x_m) \in \mathrm{FO}(\tau')$ such that, for any structure $G \in \mathcal{C}$, any $u_1, \ldots, u_m \in \mathrm{dom}(G)$, and any $k_1, \ldots, k_m \in [m]$, we have $\Phi(G) \models \psi[(k_1, u_1), \ldots, (k_m, u_m)]$ iff $((k_1, u_1), \ldots, (k_m, u_m)) = ((1, u_1), \ldots, (m, u_1))$. (The intuition is that a model $((1, u), \ldots, (m, u))$ of ψ represents $u \in \mathrm{dom}(G)$.)
(3) For any relation symbol r' from τ', say with arity l, and any $\kappa : [l] \to [m]$, there is $\varphi_\kappa^{r'}(x_1, \ldots, x_l) \in \mathrm{FO}(\tau)$ such that, for any $G \in \mathcal{C}$ and any $u_1, \ldots, u_l \in \mathrm{dom}(G)$, $G \models \varphi_\kappa^{r'}[u_1, \ldots, u_l]$ iff $\Phi(G) \models r'[(\kappa(1), u_1), \ldots, (\kappa(l), u_l)]$.
(4) For any relation symbol r from τ, say with arity l, there is $\varphi^r(x_1, \ldots, x_l) \in \mathrm{FO}(\tau')$ such that, for any $G \in \mathcal{C}$ and any $u_1, \ldots, u_l \in \mathrm{dom}(G)$, $G \models r[u_1, \ldots, u_l]$ iff $\Phi(G) \models \varphi^r[(1, u_1), \ldots, (1, u_l)]$.

Once we have a strong first-order reduction from $\mathcal{C}$ to $\mathcal{C}'$, logical definability carries over from $\mathcal{C}$ to $\mathcal{C}'$:

Theorem 6.2 ([12], Theorem 33)**.** *Let $\mathcal{C}$ and $\mathcal{C}'$ be classes of structures over relational signatures τ and τ', respectively. Let $\Phi : \mathcal{C} \to \mathcal{C}'$ be a strong first-order reduction such that $\Phi(\mathcal{C})$ is $\mathbf{\Sigma}_1(\tau')$-definable relative to $\mathcal{C}'$. Then, for any $\mathcal{L} \subseteq \mathcal{C}$ and $k \geq 1$, $\mathcal{L}$ is $\mathbf{\Sigma}_k(\tau)$-definable relative to $\mathcal{C}$ iff $\Phi(\mathcal{L})$ is $\mathbf{\Sigma}_k(\tau')$-definable relative to $\mathcal{C}'$.*

We proceed as follows. We first recall the notion of the class of grids, of which we know that the monadic second-order quantifier-alternation hierarchy is infinite. Then, we give a strong first-order reduction from the class of grids to the class of nested words over a simple 2-stack visibly pushdown alphabet so that we can deduce that the monadic second-order quantifier-alternation hierarchy over nested words is infinite, too. Note that we will add to ordinary grids some particular labeling in terms of a and b, which will simplify the upcoming constructions. It is, however, easy to see that well-known results concerning ordinary grids extend to these extended grids (cf. Theorem 6.3 below).

We fix a signature $\tau_{Grids} = \{P_a, P_b, \mathrm{succ}_1, \mathrm{succ}_2\}$ with P_a, P_b unary and $\mathrm{succ}_1, \mathrm{succ}_2$ binary relation symbols. Let $n, m \geq 1$ be natural numbers. The (n, m)-*grid* is the τ_{Grids}-structure $G(n, m) = ([n] \times [m], \mathrm{succ}_1, \mathrm{succ}_2, P_a, P_b)$ such that $\mathrm{succ}_1 = \{((i, j), (i+1, j)) \mid i \in [n-1], j \in [m]\}$, $\mathrm{succ}_2 = \{((i, j), (i, j+1)) \mid i \in [n], j \in [m-1]\}$, $P_a = \{(i, j) \in [n] \times [m] \mid j$ is odd$\}$, and $P_b = \{(i, j) \in [n] \times [m] \mid j$ is even$\}$. The $(3, 4)$-grid is illustrated in Figure 9. By $\mathbb{G}$, we denote the set of all the grids.

$$\begin{tikzcd}
a \arrow[r] \arrow[d] & b \arrow[r] \arrow[d] & a \arrow[r] \arrow[d] & b \arrow[d] \\
a \arrow[r] \arrow[d] & b \arrow[r] \arrow[d] & a \arrow[r] \arrow[d] & b \arrow[d] \\
a \arrow[r] & b \arrow[r] & a \arrow[r] & b
\end{tikzcd}$$

Figure 9: The (3,4)-grid

Theorem 6.3 ([12]). *The monadic second-order quantifier-alternation hierarchy over grids is infinite. I.e., for any $k \geq 1$, there is a set of grids that is $\mathbf{\Sigma}_{k+1}(\tau_{Grids})$-definable relative to $\mathbb{G}$ but not $\mathbf{\Sigma}_k(\tau_{Grids})$-definable relative to $\mathbb{G}$.*

For the rest of this section, we suppose that $\widetilde{\Sigma}$ is the 2-stack call-return alphabet given by $\Sigma_c^1 = \{a\}$, $\Sigma_r^1 = \{\overline{a}\}$, $\Sigma_c^2 = \{b\}$, $\Sigma_r^2 = \{\overline{b}\}$, and $\Sigma_{int} = \emptyset$.

We now describe an encoding $\Phi : \mathbb{G} \to \mathbb{NW}(\widetilde{\Sigma})$ of grids into nested words over $\widetilde{\Sigma}$. Given $n, m \geq 1$, we let

$$\Phi(G(n,m)) := \begin{cases} \text{nested}\Big(a^n \big[(\overline{a}b)^n(\overline{b}a)^n\big]^{(m-1)/2}\ \overline{a}^n\Big) & \text{if } m \text{ is odd} \\ \text{nested}\Big(a^n \big[(\overline{a}b)^n(\overline{b}a)^n\big]^{m/2-1}(\overline{a}b)^n\overline{b}^n\Big) & \text{if } m \text{ is even} \end{cases}$$

The idea is that the first n a's (and, as explained below, the corresponding return events) in a nested word represent the first column of $G(n,m)$ seen from top to bottom; the first n b's represent the second column, where the column is seen from bottom to top; the second n a's stand for the third column, again considered from top to bottom, and so on. The encoding $\Phi(G(3,4))$ of the (3,4)-grid as a nested word is depicted in Figure 10. We claim that Φ is indeed a strong first-order reduction from the set of grids to the set $\mathbb{NW}(\widetilde{\Sigma})$ of nested words over $\widetilde{\Sigma}$.

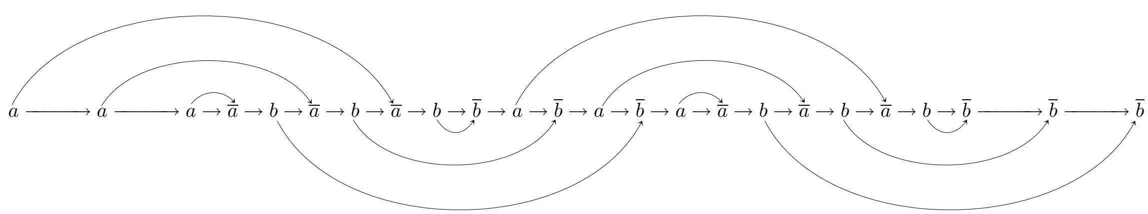

Figure 10: The encoding $\Phi(G(3,4))$ of the (3,4)-grid as a nested word

Proposition 6.4. We have that $\Phi : \mathbb{G} \to \mathbb{NW}(\widetilde{\Sigma})$ is a strong first-order reduction with rank 2. Moreover, $\Phi(\mathbb{G})$ is $\mathbf{\Sigma}_1(\tau_{\widetilde{\Sigma}})$-definable relative to $\mathbb{NW}(\widetilde{\Sigma})$.

Proof. Let us first introduce a useful notation. Given a nested word $W = ([n], \lessdot, \mu, \lambda)$ and $c \in \Sigma$ such that W contains at least k positions labeled with c, we let $pos_c(W,k)$ denote the least position i in W such that $|\{j \in [i] \mid \lambda(j) = c\}| = k$ (i.e., $pos_c(W,k)$ denotes the position of the k-th c in W).

Let $n, m \geq 1$ and let $([2 \cdot n \cdot m], \lessdot, \mu, \lambda)$ refer to $\Phi(G(n,m))$. Recall that λ can be seen as the collection of unary relations $\lambda_c = \{i \in [2 \cdot n \cdot m] \mid \lambda(i) = c\}$ for $c \in \Sigma$. Let us map any node in the (n,m)-grid (i.e., any element from $[n] \times [m]$) to a position of $\Phi(G(n,m))$ by defining a function $\chi_{n,m} : [n] \times [m] \to [2 \cdot n \cdot m]$ as follows:

$$\chi_{n,m}(i,j) = \begin{cases} pos_a(\Phi(G(n,m)), n \cdot [(j+1)/2 - 1] + i) & \text{if } j \text{ is odd} \\ pos_b(\Phi(G(n,m)), n \cdot [j/2 - 1] + (n+1-i)) & \text{if } j \text{ is even} \end{cases}$$

for any $(i,j) \in [n] \times [m]$. Intuitively, $\chi_{n,m}(i,j) \in [2 \cdot n \cdot m]$ represents the node (i,j) in the (n,m)-grid. This mapping is further extended towards a bijection $\overline{\chi}_{n,m} : \{1,2\} \times ([n] \times [m]) \to [2 \cdot n \cdot m]$ as required by Definition 6.1 (item (1)). Namely, we map $\overline{\chi}_{n,m}(1,(i,j))$ onto $\chi_{n,m}(i,j)$ and $\overline{\chi}_{n,m}(2,(i,j))$ onto $\mu(\chi_{n,m}(i,j))$.

We are prepared to specify the first-order formulas as supposed in Definition 6.1: Let

$$\psi(x_1, x_2) = \mu(x_1, x_2) \ . \tag{2}$$

Indeed, for any $n, m \geq 1$, $k_1, k_2 \in \{1,2\}$, and $u_1, u_2 \in [n] \times [m]$, we have

$$\Phi(G(n,m)) \models \psi[\overline{\chi}_{n,m}(k_1,u_1), \overline{\chi}_{n,m}(k_2,u_2)] \text{ iff } ((k_1,u_1),(k_2,u_2)) = ((1,u_1),(2,u_1)) \ .$$

We will identify a map $\kappa : [l] \to \{1,2\}$ with $(\kappa(1), \ldots, \kappa(l))$. Let, for $c \in \Sigma$ and $\kappa \in \{1,2\}$,

$$\varphi_\kappa^{\lambda_c}(x) = \begin{cases} P_c(x) & \text{if } c \in \{a,b\} \text{ and } \kappa = 1 \\ P_{\overline{c}}(x) & \text{if } c \in \{\overline{a}, \overline{b}\} \text{ and } \kappa = 2 \\ \mathit{false} & \text{otherwise} \end{cases} \tag{3}$$

where we let $\overline{\overline{a}} = a$ and $\overline{\overline{b}} = b$. For any $n, m \geq 1$, $\kappa \in \{1,2\}$, and $u \in [n] \times [m]$, we have

$$G(n,m) \models \varphi_\kappa^{\lambda_c}(x)[u] \text{ iff } \Phi(G(n,m)) \models (\lambda(x) = c)[\overline{\chi}_{n,m}(\kappa,u)] \ .$$

Further, let, for $\kappa \in \{1,2\} \times \{1,2\}$,

$$\varphi_\kappa^{\lessdot}(x_1,x_2) = \begin{cases} \mathrm{succ}_1(x_1,x_2) \wedge \neg\exists z\ \mathrm{succ}_2(z,x_1) & \text{if } \kappa = (1,1) \\ \begin{pmatrix} P_a(x_1) \wedge \mathrm{succ}_1(x_2,x_1) \wedge \neg\exists z\ \mathrm{succ}_2(x_1,z) \\ \vee\ P_b(x_1) \wedge \mathrm{succ}_1(x_1,x_2) \wedge \neg\exists z\ \mathrm{succ}_2(x_1,z) \end{pmatrix} & \text{if } \kappa = (2,2) \\ \begin{pmatrix} (x_1 = x_2) \wedge P_a(x_1) \wedge \neg\exists z\ \mathrm{succ}_1(x_1,z) \\ \vee\ (x_1 = x_2) \wedge P_b(x_1) \wedge \neg\exists z\ \mathrm{succ}_1(z,x_1) \\ \vee\ P_a(x_1) \wedge P_b(x_2) \wedge \exists z\ (\mathrm{succ}_1(z,x_1) \wedge \mathrm{succ}_2(z,x_2)) \\ \vee\ P_b(x_1) \wedge P_a(x_2) \wedge \exists z\ (\mathrm{succ}_1(z,x_1) \wedge \mathrm{succ}_2(x_2,z)) \end{pmatrix} & \text{if } \kappa = (1,2) \\ \begin{pmatrix} P_a(x_1) \wedge P_b(x_2) \wedge \mathrm{succ}_2(x_1,x_2) \\ \vee\ P_b(x_1) \wedge P_a(x_2) \wedge \mathrm{succ}_2(x_1,x_2) \end{pmatrix} & \text{if } \kappa = (2,1) \end{cases}$$

For any $n, m \geq 1$, $\kappa \in \{1,2\} \times \{1,2\}$, and $u_1, u_2 \in [n] \times [m]$, we have

$$G(n,m) \models \varphi_\kappa^{\lessdot}(x)[u_1,u_2] \text{ iff } \Phi(G(n,m)) \models (x_1 \lessdot x_2)[\overline{\chi}_{n,m}(\kappa(1),u_1), \overline{\chi}_{n,m}(\kappa(2),u_2)] \ .$$

Finally, to complete step (3), let, for $\kappa \in \{1,2\} \times \{1,2\}$,

$$\varphi_\kappa^{\mu}(x_1,x_2) = \begin{cases} x_1 = x_2 & \text{if } \kappa = (1,2) \\ \mathit{false} & \text{otherwise} \end{cases}$$

Then, for any $n, m \geq 1$, $\kappa \in \{1,2\} \times \{1,2\}$ and $u_1, u_2 \in [n] \times [m]$,

$$G(n,m) \models \varphi^\mu_\kappa(x)[u_1,u_2] \text{ iff } \Phi(G(n,m)) \models (\mu(x_1,x_2))[\overline{\chi}_{n,m}(\kappa(1),u_1), \overline{\chi}_{n,m}(\kappa(2),u_2)] \ .$$

Let

$$\varphi^{P_a}(x) = (\lambda(x) = a) \text{ and } \varphi^{P_b}(x) = (\lambda(x) = b) \ . \tag{4}$$

Of course, we have, for any $n, m \geq 1$, $c \in \{a, b\}$, and $u \in [n] \times [m]$,

$$G(n,m) \models P_c(x)[u] \text{ iff } \Phi(G(n,m)) \models (\varphi^{P_c})[\overline{\chi}_{n,m}(1,u)] \ .$$

Let

$$\varphi^{\mathrm{succ}_1}(x_1,x_2) = \left(\begin{array}{ll} & \lambda(x_1) = a \wedge \lambda(x_2) = a \wedge (x_1 \lessdot x_2 \vee \exists z\ (x_1 \lessdot z \wedge z \lessdot x_2)) \\ \vee & \lambda(x_1) = b \wedge \lambda(x_2) = b \wedge \exists z\ (x_2 \lessdot z \wedge z \lessdot x_1) \end{array} \right)$$

and let furthermore

$$\varphi^{\mathrm{succ}_2}(x_1,x_2) = \exists z\ (\mu(x_1,z) \wedge z \lessdot x_2) \ .$$

Then, for any $n, m \geq 1$, $u_1, u_2 \in [n] \times [m]$, and $k \in \{1,2\}$, it holds

$$G(n,m) \models \mathrm{succ}_k(x_1,x_2)[u_1,u_2] \text{ iff } \Phi(G(n,m)) \models (\varphi^{\mathrm{succ}_k})[\overline{\chi}_{n,m}(1,u_1), \overline{\chi}_{n,m}(1,u_2)] \ .$$

With the above formulas, it is now immediate to verify that Φ is indeed a strong first-order reduction.

Now observe that $\Phi(\mathbb{G})$ is the "conjunction" of

- the regular expression $\Big(a^+\big[(\overline{a}b)^+(\overline{b}a)^+\big]^*\ \overline{a}^+\Big) + \Big(a^+\big[(\overline{a}b)^+(\overline{b}a)^+\big]^*(\overline{a}b)^+\overline{b}^+\Big)$,
- the first-order formula $\forall x \exists y\ \big(\mu(x,y) \vee \mu(y,x)\big)$, and
- the first-order property (written in shorthand)

$$\begin{array}{lll} \forall x_1,x_2,y_1,y_2 \Big(& & \lambda(x_1) = \lambda(x_2) \wedge \mu(x_1,y_1) \wedge \mu(x_2,y_2) \\ & \rightarrow & \quad \big(\lambda(x_1) = a \wedge x_2 - x_1 = 1 \ \rightarrow\ y_1 - y_2 \in \{1,2\}\big) \\ & \wedge & \quad \big(\lambda(y_1) = \overline{a} \wedge y_1 - y_2 = 1 \ \rightarrow\ x_2 - x_1 \in \{1,2\}\big) \\ & \wedge & \quad \big(\lambda(y_1) = \overline{b} \wedge y_1 - y_2 = 1 \ \rightarrow\ x_2 - x_1 = 2\big) \\ & \wedge & \quad \big(x_2 - x_1 = 2 \wedge \lambda(x_1+1) \neq \lambda(x_1) \ \rightarrow\ y_1 - y_2 \in \{1,2\}\big) \\ & \wedge & \quad \big(y_1 - y_2 = 2 \wedge \lambda(y_2+1) \neq \lambda(y_2) \ \rightarrow\ x_2 - x_1 \in \{1,2\}\big)\Big) \end{array}$$

As the regular expression represents a $\mathbf{\Sigma}_1(\tau_{\widetilde{\Sigma}})$-definable property, $\Phi(\mathbb{G})$ is $\mathbf{\Sigma}_1(\tau_{\widetilde{\Sigma}})$-definable relative to $\mathbb{NW}(\widetilde{\Sigma})$, which concludes the proof of Proposition 6.4. $\square$

Combining Theorem 6.2, Theorem 6.3, and Proposition 6.4, we obtain the following:

Theorem 6.5. *The monadic second-order quantifier-alternation hierarchy over nested words is infinite. I.e., for any $k \geq 1$, there is a set of nested words over $\widetilde{\Sigma}$ that is $\mathbf{\Sigma}_{k+1}(\tau_{\widetilde{\Sigma}})$-definable relative to $\mathbb{NW}(\widetilde{\Sigma})$ but not $\mathbf{\Sigma}_k(\tau_{\widetilde{\Sigma}})$-definable relative to $\mathbb{NW}(\widetilde{\Sigma})$.*

Note that Theorem 6.5 relies neither on a particular call-return alphabet nor on a certain number of stacks (unless there is only one stack), as its proof is based on the simplest possible 2-stack call-return alphabet $\widetilde{\Sigma}$, which is given by $\Sigma^1_c = \{a\}$, $\Sigma^1_r = \{\overline{a}\}$, $\Sigma^2_c = \{b\}$, $\Sigma^2_r = \{\overline{b}\}$, and $\Sigma_{int} = \emptyset$.

Finally, Theorems 5.2 and 6.5 give rise to the following theorem:

Theorem 6.6. *The class of nested-word languages that are recognized by* 2VPA *is not closed under complementation. More precisely, for any* 2*-stack call-return alphabet* $\widetilde{\Sigma}$*, there is a set* $\mathcal{L}$ *of nested words over* $\widetilde{\Sigma}$ *such that the following hold:*

- *There is a* 2VPA $\mathcal{A}$ *over* $\widetilde{\Sigma}$ *such that* $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.
- *There is no* 2VPA $\mathcal{A}$ *over* $\widetilde{\Sigma}$ *such that* $\mathcal{L}(\mathcal{A}) = \mathbb{NW}(\widetilde{\Sigma}) \setminus \mathcal{L}$.

This implies that the deterministic model of a 2VPA (see [16] for its formal definition) is strictly weaker than the general model. This fact was, however, already shown in [16]: Consider the language $L = \{(ab)^m c^n d^{m-n} x^n y^{m-n} \mid m \in \mathbb{N}, n \in [m]\}$ and the 2-stack call-return alphabet $\widetilde{\Sigma}$ given by $\Sigma_c^1 = \{a\}$, $\Sigma_r^1 = \{c, d\}$, $\Sigma_c^2 = \{b\}$, $\Sigma_r^2 = \{x, y\}$, and $\Sigma_{int} = \emptyset$. Then, L is accepted by some 2VPA over $\widetilde{\Sigma}$ but not by any deterministic 2VPA over $\widetilde{\Sigma}$.

7. Büchi Multi-Stack Visibly Pushdown Automata

We now transfer some fundamental notions and results from the finite case into the setting of infinite (nested) words.

7.1. Büchi Multi-Stack Visibly Pushdown Automata.

Let $K \geq 1$, and let $\widetilde{\Sigma} = \langle\{(\Sigma_c^s, \Sigma_r^s)\}_{s\in[K]}, \Sigma_{int}\rangle$ be a K-stack call-return alphabet.

Definition 7.1. A *Büchi multi-stack visibly pushdown automaton* (Büchi MVPA) over $\widetilde{\Sigma}$ is a tuple $\mathcal{A} = (Q, \Gamma, \delta, Q_I, F)$ whose components agree with those of an ordinary MVPA, i.e., Q is its finite set of *states*, $Q_I \subseteq Q$ is the set of *initial states*, $F \subseteq Q$ is the set of *final states*, Γ is the finite *stack alphabet* containing the special symbol $\perp$, and δ is a triple $\langle\delta_c, \delta_r, \delta_{int}\rangle$ with $\delta_c \subseteq Q \times \Sigma_c \times (\Gamma \setminus \{\perp\}) \times Q$, $\delta_r \subseteq Q \times \Sigma_r \times \Gamma \times Q$, and $\delta_{int} \subseteq Q \times \Sigma_{int} \times Q$.

A *Büchi* 2*-stack visibly pushdown automaton* (Büchi 2VPA) is a Büchi MVPA that is defined over a 2-stack alphabet.

Consider an infinite string $w = a_1a_2\ldots \in \Sigma^\omega$. A run of the Büchi MVPA $\mathcal{A}$ on w is a sequence $\rho = (q_0, \sigma_0^1, \ldots \sigma_0^K)(q_1, \sigma_1^1, \ldots \sigma_1^K)\ldots \in (Q \times Cont^{[K]})^\omega$ (recall that $Cont = (\Gamma \setminus \{\perp\})^* \cdot \{\perp\}$) such that $q_0 \in Q_I$, $\sigma_0^s = \perp$ for any stack $s \in [K]$, and **[Push]**, **[Pop]**, and **[Internal]** as specified in the finite case hold for any $i \in \mathbb{N}_+$. We call the run accepting if $\{q \mid q = q_i \text{ for infinitely many } i \in \mathbb{N}\} \cap F \neq \emptyset$. A string $w \in \Sigma^\omega$ is accepted by $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on w. The such defined (string) language of $\mathcal{A}$ is denoted by $L^\omega(\mathcal{A})$.

For the infinite case, we can likewise establish a relational structure of *infinite* nested words:

Definition 7.2. An *infinite nested word* over $\widetilde{\Sigma}$ is a structure $(\mathbb{N}_+, \lessdot, \mu, \lambda)$ where $\lessdot = \{(i, i+1) \mid i \in \mathbb{N}_+\}$, $\lambda : \mathbb{N}_+ \to \Sigma$, and $\mu = \bigcup_{s\in[K]} \mu^s \subseteq \mathbb{N}_+ \times \mathbb{N}_+$ where, for any $s \in [K]$ and $(i, j) \in \mathbb{N}_+ \times \mathbb{N}_+$, $(i, j) \in \mu^s$ iff $i < j$, $\lambda(i) \in \Sigma_c^s$, $\lambda(j) \in \Sigma_r^s$, and $\lambda(i+1)\ldots\lambda(j-1)$ is s-well formed.

The set of infinite nested words over $\widetilde{\Sigma}$ is denoted by $\mathbb{NW}^\omega(\widetilde{\Sigma})$. Again, given infinite nested words $W = (\mathbb{N}_+, \lessdot, \mu, \lambda)$ and $W' = (\mathbb{N}_+, \lessdot', \mu', \lambda')$, $\lambda = \lambda'$ implies $W = W'$ so that we can represent W as $\text{string}(W) := \lambda(1)\lambda(2)\ldots \in \Sigma^\omega$. Vice versa, given a string $w \in \Sigma^\omega$, there is exactly one infinite nested word W over $\widetilde{\Sigma}$ such that $\text{string}(W) = w$, which we denote $\text{nested}(w)$.

Definition 7.3. A *generalized Büchi multi-stack nested-word automaton* (generalized Büchi MNWA) over $\widetilde{\Sigma}$ is a tuple $\mathcal{B} = (Q, \delta, Q_I, F, C)$ where Q, δ, Q_I, F, and C are as in a generalized MNWA. Recall that, in particular, δ is a pair $\langle \delta_1, \delta_2 \rangle$ with $\delta_1 \subseteq Q \times \Sigma \times Q$ and $\delta_2 \subseteq Q \times Q \times \Sigma_r \times Q$.

We call $\mathcal{B}$ a *generalized* Büchi 2-stack nested-word automaton (generalized Büchi 2NWA) if it is defined over a 2-stack alphabet.

If $C = \emptyset$, then we may call $\mathcal{B}$ a Büchi MNWA (Büchi 2NWA, if $K = 2$).

A run of $\mathcal{B}$ on an infinite nested word $W = (\mathbb{N}_+, \lessdot, \mu, \lambda) \in \mathbb{NW}^\omega(\widetilde{\Sigma})$ is a mapping $\rho : \mathbb{N}_+ \to Q$ such that $(q, \lambda(1), \rho(1)) \in \delta_1$ for some $q \in Q_I$, and, for any $i \geq 2$, we have

$$\begin{cases} (\rho(\mu^{-1}(i)), \rho(i-1), \lambda(i), \rho(i)) \in \delta_2 & \text{if } \lambda(i) \in \Sigma_r \text{ and } \mu^{-1}(i) \text{ is defined} \\ (\rho(i-1), \lambda(i), \rho(i)) \in \delta_1 & \text{otherwise} \end{cases}$$

The run ρ is accepting if $\rho(i) \in F$ for infinitely many $i \in \mathbb{N}_+$ and, for any $i \in \mathbb{N}_+$ with $\rho(i) \in C$, both $\lambda(i) \in \Sigma_c$ and $\mu(i)$ is defined. The language of $\mathcal{B}$, denoted by $\mathcal{L}^\omega(\mathcal{B})$, is the set of infinite nested words over $\widetilde{\Sigma}$ that allow for an accepting run of $\mathcal{B}$.

As we still have a one-to-one correspondence between strings and nested words, we may let $\mathcal{L}^\omega(\mathcal{A})$ with $\mathcal{A}$ a Büchi MVPA stand for the set $\{\text{nested}(w) \mid w \in L^\omega(\mathcal{A})\}$.

It is now straightforward to adapt Lemma 2.7 and Lemma 2.8 to the infinite setting:

Lemma 7.4. For any generalized Büchi MNWA $\mathcal{B}$, there is a Büchi MNWA $\mathcal{B}'$ such that $\mathcal{L}^\omega(\mathcal{B}') = \mathcal{L}^\omega(\mathcal{B})$.

Lemma 7.5. Let $\mathcal{L} \subseteq \mathbb{NW}^\omega(\widetilde{\Sigma})$. The following are equivalent:

(1) There is a Büchi MVPA $\mathcal{A}$ such that $\mathcal{L}^\omega(\mathcal{A}) = \mathcal{L}$.
(2) There is a Büchi MNWA $\mathcal{B}$ such that $\mathcal{L}^\omega(\mathcal{B}) = \mathcal{L}$.

7.2. **Büchi 2-Stack Visibly Pushdown Automata vs. Logic.** In this section, we will again restrict to two stacks. Unfortunately, EMSO logic over nested words turns out to be too weak to capture all the behaviors of Büchi 2VPA. In this logic, considered over infinite words, one cannot even express that one particular action occurs infinitely often. To overcome this deficiency, one can introduce a first-order quantifier $\exists^\infty x\varphi$ meaning that there are infinitely many positions x to satisfy the property φ [3].

So let us fix a 2-stack call-return alphabet $\widetilde{\Sigma} = \langle \{(\Sigma_c^1, \Sigma_r^1), (\Sigma_c^2, \Sigma_r^2)\}, \Sigma_{int} \rangle$ for the rest of the paper. We introduce the logic $\mathrm{MSO}^\infty(\tau_{\widetilde{\Sigma}})$, which is given by the following grammar:

$$\begin{aligned} \varphi ::= \lambda(x) = a \mid x \lessdot y \mid \mu(x,y) \mid x = y \mid x \in X \mid \\ \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x\varphi \mid \exists^\infty x\varphi \mid \exists X\varphi \end{aligned}$$

where $a \in \Sigma$. The fragments $\mathrm{EMSO}^\infty(\tau_{\widetilde{\Sigma}})$ and $\mathrm{FO}^\infty(\tau_{\widetilde{\Sigma}})$ are defined as one would expect. The satisfaction relation is as usual concerning the familiar fragment $\mathrm{MSO}(\tau_{\widetilde{\Sigma}})$. Moreover, given a formula $\varphi(y, x_1, \ldots, x_m, X_1, \ldots, X_n) \in \mathrm{MSO}^\infty(\tau_{\widetilde{\Sigma}})$, an infinite nested word W, $(i_1, \ldots, i_m) \in (\mathbb{N}_+)^m$, and $(I_1, \ldots, I_n) \in (2^{\mathbb{N}_+})^n$, we set $W \models (\exists^\infty y\varphi)[i_1, \ldots, i_m, I_1, \ldots, I_n]$ iff $W \models \varphi[i, i_1, \ldots, i_m, I_1, \ldots, I_n]$ for infinitely many $i \in \mathbb{N}_+$. Given a sentence $\varphi \in \mathrm{MSO}^\infty(\tau_{\widetilde{\Sigma}})$, we denote by $\mathcal{L}^\omega(\varphi)$ the set of infinite nested words over $\widetilde{\Sigma}$ that satisfy φ.

We extend Definition 3.1 towards another equivalence relation of finite index: For $k \in \mathbb{N}$ and infinite nested words $U, V \in \mathbb{NW}^\omega(\widetilde{\Sigma})$, we write $U \equiv^\infty_{k,\widetilde{\Sigma}} V$ if, for any first-order sentence $\varphi \in \mathrm{FO}^\infty(\tau_{\widetilde{\Sigma}})$ with $\mathrm{rank}(\varphi) \leq k$, we have $U \models \varphi$ iff $V \models \varphi$. Hereby, the rank of a formula carries over to the extended logic in a straightforward manner, i.e., $\mathrm{rank}(\exists^\infty x\varphi) = \mathrm{rank}(\varphi) + 1$.

Definition 7.6 ([3])**.** Let $r, t \in \mathbb{N}$ and let $U, V \in \mathbb{NW}^\omega(\widetilde{\Sigma})$ be infinite nested words. We write $U \leftrightarrows^\infty_{r,t} V$ if, for any isomorphism type $S \in \mathit{Spheres}_r(\widetilde{\Sigma})$ of an r-sphere, we have $|U|_S = |V|_S$ or both $t < |U|_S < \infty$ and $t < |V|_S < \infty$.

To establish a connection between the extended logic and our Büchi automata models, we have to provide an extension of Hanf's Theorem. Indeed, the extended threshold equivalence is a refinement of first-order definability in the context of the new quantifier:

Theorem 7.7 ([3])**.** *Let $k \in \mathbb{N}$. There exist $r, t \in \mathbb{N}$ such that, for any two infinite nested words $U, V \in \mathbb{NW}^\omega(\widetilde{\Sigma})$, $U \leftrightarrows^\infty_{r,t} V$ implies $U \equiv^\infty_{k,\widetilde{\Sigma}} V$.*

We observe that the 2NWA $\mathcal{B}_r$ constructed in the proof of Proposition 4.1 can be easily adapted to obtain its counterpart for infinite nested words:

Proposition 7.8. Let $r \in \mathbb{N}$ be any natural number. There are a generalized Büchi 2NWA $\mathcal{B}^\omega_r = (Q, \delta, Q_I, F, C)$ over $\widetilde{\Sigma}$ and a mapping $\eta : Q \to \mathit{Spheres}_r(\widetilde{\Sigma})$ such that

- $\mathcal{L}^\omega(\mathcal{B}^\omega_r) = \mathbb{NW}^\omega(\widetilde{\Sigma})$ and
- for any $W \in \mathbb{NW}^\omega(\widetilde{\Sigma})$, for any accepting run ρ of $\mathcal{B}^\omega_r$ on W, and for any node $i \in \mathbb{N}_+$ of W, we have $\eta(\rho(i)) \cong r\text{-Sph}(W, i)$.

Proof. First, we observe that Proposition 4.3 and the crucial argument stated in the proof of Claim 4.6 (see Footnote 3) hold for infinite nested words just as well. Now, look at the generalized 2NWA $\mathcal{B}_r = (Q, \delta, Q_I, F, C)$ as constructed in the proof of Proposition 4.1. As the only purpose of the set F of final states is to ensure progress in some states where progress is required in terms of spheres with a non-maximal active node, we can set $\mathcal{B}^\omega_r$ to be (Q, δ, Q_I, Q, C), and we are done. □

With this, we can easily extend Lemma 5.1 and determine a Büchi 2NWA to detect if a particular sphere occurs infinitely often in an infinite nested word:

Lemma 7.9. Let $r, t \in \mathbb{N}$ and let $S \in \mathit{Spheres}_r(\widetilde{\Sigma})$. There is a generalized Büchi 2NWA $\mathcal{B}$ over $\widetilde{\Sigma}$ such that $\mathcal{L}^\omega(\mathcal{B}) = \{W \in \mathbb{NW}^\omega(\widetilde{\Sigma}) \mid$ there are infinitely many $i \in \mathbb{N}_+$ such that $r\text{-Sph}(W, i) \cong S\}$.

Proof. We start from the generalized Büchi 2NWA $\mathcal{B}^\omega_r = (Q, \delta, Q_I, Q, C)$ and the mapping $\eta : Q \to \mathit{Spheres}_r(\widetilde{\Sigma})$ from Proposition 7.8. To obtain $\mathcal{B}$ as required in the proposition, we simply set the set of final states to be $\{q \in Q \mid \eta(q) \cong S\}$. □

Theorem 7.10. *Let $\mathcal{L} \subseteq \mathbb{NW}^\omega(\widetilde{\Sigma})$ be a set of infinite nested words over the 2-stack call-return alphabet $\widetilde{\Sigma}$. Then, the following are equivalent:*

1. *There is a Büchi 2VPA $\mathcal{A}$ over $\widetilde{\Sigma}$ such that $\mathcal{L}^\omega(\mathcal{A}) = \mathcal{L}$.*
2. *There is a sentence $\varphi \in \mathrm{EMSO}^\infty(\tau_{\widetilde{\Sigma}})$ such that $\mathcal{L}^\omega(\varphi) = \mathcal{L}$.*

Proof. To prove (1) → (2), one again uses standard methods. Basically, second-order variables X_q for $q \in Q$ encode an assignment of states to positions in a nested word. Then, the first-order part of the formula expresses that this assignment is actually an accepting run. To take care of the acceptance condition, we add the disjunction of formulas $\exists^\infty x\ (x \in X_q)$ with q a final state.

For the direction (2) → (1), we make use of Lemmas 7.4, 7.5, 7.9, (a simple variation of) Lemma 5.1, and the easy fact that the class of languages of infinite nested words that are recognized by generalized Büchi 2NWA is closed under union and intersection. With this, the proof proceeds exactly as in the finite case. □

8. Open Problems

We leave open if visibly pushdown automata still admit a logical characterization in terms of EMSO logic once they are equipped with more than two stacks.

We conjecture that any first-order definable set of nested words over two stacks is recognized by some unambiguous 2VPA, i.e., by a 2VPA in which an accepting run is unique. To achieve such an automaton, the coloring of spheres as performed by $\mathcal{B}_r$ by simply *guessing* and subsequently verifying it has to be done unambiguously.

We do not know if EMSO logic over nested words becomes more expressive if we allow atomic formulas $x < y$ with the obvious meaning. For this logic, it is no longer possible to apply Hanf's theorem as the degree of the corresponding class of structures is not bounded anymore.

Finally, it might be worthwhile to study if our technique leads to a logical characterization of 2VPA for more general 2-stack call-return alphabets as introduced in [6].

References

[1] R. Alur and P. Madhusudan. Visibly pushdown languages. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC 2004)*, pages 202–211. ACM Press, 2004.
[2] R. Alur and P. Madhusudan. Adding nesting structure to words. In *Proceedings of the 10th International Conference on Developments in Language Theory (DLT 2006)*, volume 4036 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2006.
[3] B. Bollig and D. Kuske. Distributed Muller automata and logics. Research Report LSV-06-11, Laboratoire Spécification et Vérification, ENS Cachan, France, May 2006.
[4] B. Bollig and M. Leucker. Message-passing automata are expressively equivalent to EMSO logic. *Theoretical Computer Science*, 358(2-3):150–172, 2006.
[5] J. Büchi. Weak second order logic and finite automata. *Z. Math. Logik, Grundlag. Math.*, 5:66–62, 1960.
[6] D. Carotenuto, A. Murano, and A. Peron. 2-visibly pushdown automata. In *Proceedings of the 11th International Conference on Developments in Language Theory (DLT 2007)*, volume 4588 of *Lecture Notes in Computer Science*, pages 132–144. Springer, 2007.
[7] M. Droste, P. Gastin, and D. Kuske. Asynchronous cellular automata for pomsets. *Theoretical Computer Science*, 247(1-2):1–38, 2000.
[8] C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.*, 98:21–52, 1961.
[9] W. Hanf. Model-theoretic methods in the study of elementary logic. In J. W. Addison, L. Henkin, and A. Tarski, editors, *The Theory of Models*. North-Holland, Amsterdam, 1965.
[10] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages and Computability*. Addison-Wesley, 2000.
[11] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.
[12] O. Matz, N. Schweikardt, and W. Thomas. The monadic quantifier alternation hierarchy over grids and graphs. *Information and Computation*, 179(2):356–383, 2002.
[13] W. Thomas. Elements of an automata theory over partial orders. In *Proceedings of Workshop on Partial Order Methods in Verification (POMIV 1996)*, volume 29 of *DIMACS*. AMS, 1996.
[14] W. Thomas. Automata theory on trees and partial orders. In *Proceedings of Theory and Practice of Software Development (TAPSOFT 1997), 7th International Joint Conference CAAP/FASE*, volume 1214 of *Lecture Notes in Computer Science*, pages 20–38. Springer, 1997.

[15] W. Thomas. Languages, automata and logic. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume 3, Beyond Words, pages 389–455. Springer, 1997.

[16] S. La Torre, P. Madhusudan, and G. Parlato. A robust class of context-sensitive languages. In *Proceedings of the 22nd IEEE Symposium on Logic in Computer Science (LICS 2007)*, pages 161–170. IEEE Computer Society Press, 2007.

Appendix A. The Naïve Approach Fails

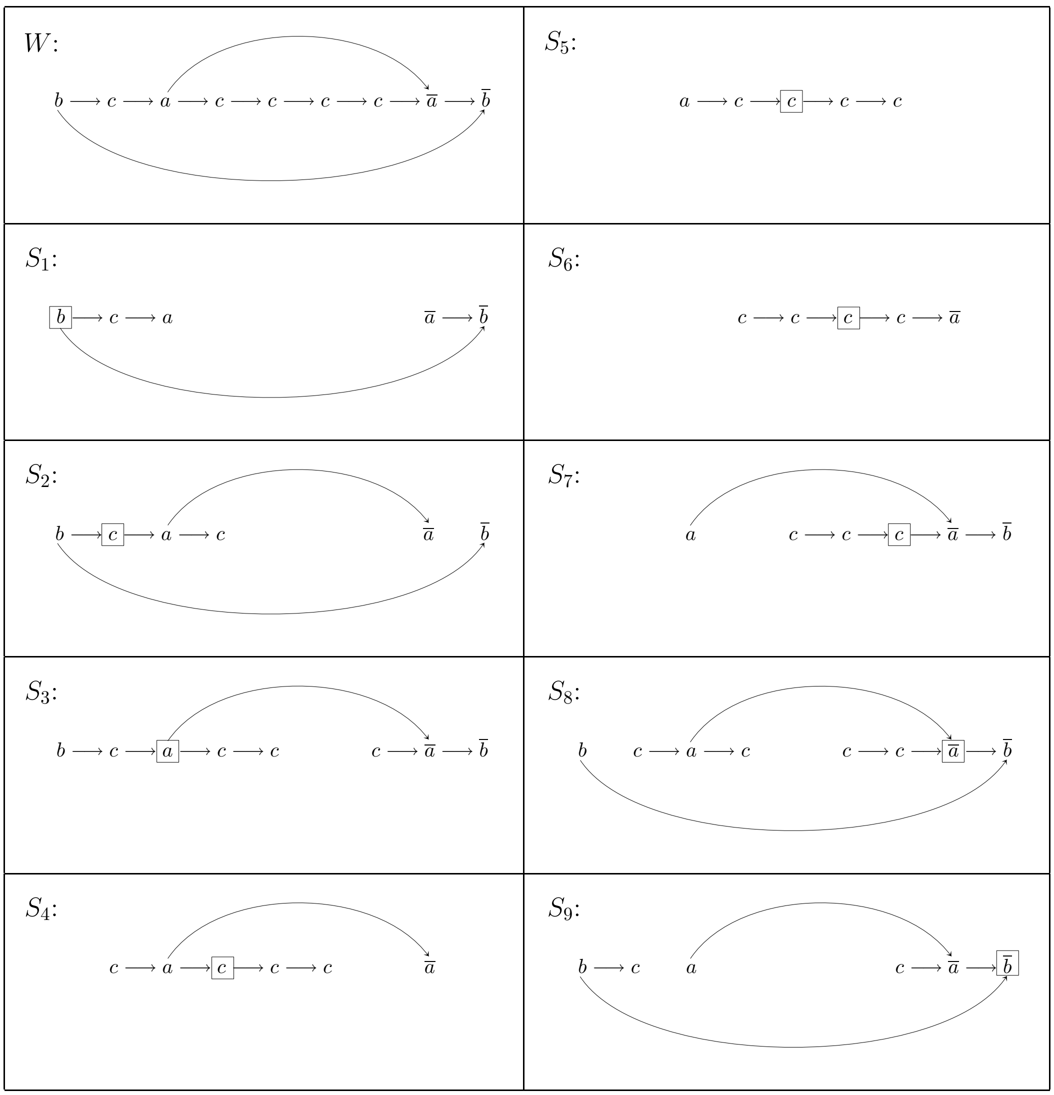

Figure 11: An accepting run of the naïve automaton on W