FIRST-ORDER LOGIC ON HIGHER-ORDER NESTED PUSHDOWN TREES

ALEXANDER KARTZOW
UNIVERSITÄT LEIPZIG

Abstract. We introduce a new hierarchy of *higher-order nested pushdown trees* generalising Alur et al.'s concept of nested pushdown trees. Nested pushdown trees are useful representations of control flows in the verification of programmes with recursive calls of first-order functions. Higher-order nested pushdown trees are expansions of unfoldings of graphs generated by higher-order pushdown systems. Moreover, the class of nested pushdown trees of level n is uniformly first-order interpretable in the class of collapsible pushdown graphs of level $n+1$. The relationship between the class of higher-order pushdown graphs and the class of collapsible higher-order pushdown graphs is not very well understood. We hope that the further study of the nested pushdown tree hierarchy leads to a better understanding of these two hierarchies. In this paper, we are concerned with the first-order model checking problem on higher-order nested pushdown trees. We show that the first-order model checking on the first two levels of this hierarchy is decidable. Moreover, we obtain an 2-EXPSPACE algorithm for the class of nested pushdown trees of level 1. The proof technique involves a pseudo-local analysis of strategies in the Ehrenfeucht-Fraïssé games on two identical copies of a nested pushdown tree. Ordinary locality arguments in the spirit of Gaifman's lemma do not apply here because nested pushdown trees tend to have small diameters. We introduce the notion of relevant ancestors which provide a sufficient description of the FO_k-type of each element in a higher-order nested pushdown tree. The local analysis of these ancestors allows us to prove the existence of restricted winning strategies in the Ehrenfeucht-Fraïssé game. These strategies are then used to create a first-order model checking algorithm.

1. Introduction

During the last decade, different generalisations of pushdown systems have gained attention in the field of software verification and model checking. Knapik et al. [15] showed that Higher-order pushdown systems, first defined by Maslov [16, 17], generate the same class of trees as *safe* higher-order recursion schemes. Safety is a syntactic condition concerning the order of the output compared to the order of the inputs of higher-order recursion schemes. A higher-order pushdown system is a pushdown system that uses a nested stack structure instead of an ordinary stack. This means that a level 2 pushdown system (2-PS) uses a stack of stacks, a level 3 pushdown system (3-PS) uses a stack of stacks of stacks, etc. Hague et al. [10] defined the class of collapsible pushdown systems by adding a new

Key words and phrases: higher-order pushdown graph, higher-order pushdown system, first-order logic, FO, decidability , pumping lemma, nested pushdown tree, Ehrenfeucht-Fraïssé game, first-oder model checking.

stack operation called collapse. They proved that the trees generated by level l collapsible pushdown systems coincide with the trees generated by level l recursion schemes. The exact relationship between the higher-order pushdown hierarchy and the collapsible pushdown hierarchy remains an open problem. It is not known whether the trees generated by safe recursion schemes are a proper subclass of the trees generated by all recursion schemes.[1] Due to the correspondence of recursion schemes and pushdown trees, this question can be equivalently formulated as follows: is there some collapsible pushdown system that generates a tree which is not generated by any higher-order pushdown system?

Also from a model theoretic perspective these hierarchies are interesting classes. The graphs generated by higher-order pushdown systems are exactly the graphs in the Caucal-hierarchy [6]. Thus, they are one of the largest known classes with decidable monadic second-order theories. In contrast, Broadbent [4] recently showed that the first-order theories of graphs generated by level 3 collapsible pushdown systems are in general undecidable (level 2 collapsible pushdown graphs have undecidable monadic second-order theories but decidable first-order theories [10, 13]). Thus, the collapse operation induces a drastic change with respect to classical decidability issues.

Furthermore, collapsible pushdown graphs have decidable modal μ-calculus theories [10]. In fact, the class of collapsible pushdown graphs and the class of nested pushdown trees are the only known natural classes with decidable modal μ-calculus theories but undecidable monadic second-order theories.

Further study of the higher-order pushdown hierarchy and the collapsible pushdown hierarchy is necessary for a better understanding of these results. It may also reveal an answer to the question whether safety implies a semantical restriction for recursion schemes.

In this paper we introduce the hierarchy of higher-order nested pushdown trees. This is a new hierarchy between the hierarchy of higher-order pushdown trees and that of collapsible pushdown graphs. We hope that its study reveals more insights into the structure of these hierarchies.

Nested pushdown trees were first introduced by Alur et al. [1]. These are trees generated by pushdown systems (of level 1) enriched by a new *jump relation* that connects each push operation with the corresponding pop operations. They introduced these trees in order to verify specifications concerning pre/postconditions on function calls/returns in recursive first-order programmes. "Ordinary" pushdown trees offer suitable representations of control flows of recursive first-order functions. Since these trees have decidable monadic second-order theories [18], one can use these representations fruitfully for verification purposes. But monadic second-order logic does not provide the expressive power necessary for defining the position before and after the call of a certain function in such a pushdown tree. Alur et al.'s new jump relation makes these pairs of positions definable by a quantifier-free formula. Unfortunately, this new relation turns the monadic second-order theories undecidable. But they showed that modal μ-calculus model checking is still decidable on the class of nested pushdown trees. Thus, nested pushdown trees form a suitable representation for control flows of first-order recursive programmes for the verification of modal μ-calculus definable properties of the control flows including pre/postconditions on function calls/returns.

Of course, the idea of making corresponding push and pop operations visible is not restricted to pushdown systems of level 1. We define a level n nested pushdown tree (n-NPT) to be a tree generated by a level n pushdown system (without collapse!) expanded by

[1] Recently, P. Parys [20] proved the uniform safety conjecture for level 2: there is a level 2 recursion scheme that is not generated by any level 2 safe scheme.

a jump relation that connects every push of level n with the corresponding pop operations (of level n).

This new hierarchy contains by definition expansions of higher-order pushdown trees. Moreover, we show that the class of n-NPT is uniformly first-order interpretable in the class of level $n+1$ collapsible pushdown graphs.

We then study first-order model checking on the first two levels of this new hierarchy. Especially, we provide a 2-EXPTIME alternating Turing machine deciding the model checking problem for the class of 1-NPT. We already proved the same complexity bound in [12]. Here, we reprove the statement with a different technical approach that generalises to the higher levels of the nested pushdown tree hierarchy.

Outline. Section 2 contains some basic definitions concerning first-order logic and higher-order pushdown systems. Moreover, we recall the basics of Ehrenfeucht-Fraïssé games and explain how the analysis of strategies in these games can be used to derive first-order model checking algorithms on certain classes of structures. In Section 3, we then introduce the hierarchy of nested pushdown trees. We relate this hierarchy to the hierarchies of pushdown trees and of collapsible pushdown graphs. From that point on, we only focus on the first-order model checking problem for the first two levels of the nested pushdown tree hierarchy. In Section 3.1, we explain the rough picture how the ideas of Section 2.1 lead to a model checking algorithm for this class. We then give an outline of the Sections 4–7 which provide the details of the correctness proof for the model checking algorithm presented in Section 7. Finally, we give some concluding remarks and point to open problems in Section 8.

2. Preliminaries and Basic Definitions

We denote first-order logic by FO. The quantifier rank of some formula $\varphi \in \mathrm{FO}$ is the maximal number of nestings of existential and universal quantifiers in φ. We denote by FO_ρ the set of first-order formulas of quantifier rank up to ρ and by $\equiv_\rho$ equivalence of structures with respect to all FO_ρ formulas. This means that $\mathfrak{A} \equiv_\rho \mathfrak{B}$ if and only if for all $\varphi \in \mathrm{FO}_\rho, \quad \mathfrak{A} \models \varphi \Leftrightarrow \mathfrak{B} \models \varphi$ holds.

The FO model checking problem on a class $\mathcal{C}$ asks for an algorithm that determines whether $\mathfrak{A} \models \varphi$ on input $(\mathfrak{A}, \varphi)$ where $\mathfrak{A} \in \mathcal{C}$ and $\varphi \in \mathrm{FO}$. In Section 2.1, we develop a translation from *dynamic-small-witness strategies* in Ehrenfeucht-Fraïssé games to FO model checking algorithms on nice classes of structures. A dynamic-small-witness strategy in the Ehrenfeucht-Fraïssé game allows Duplicator to answer any challenge of Spoiler by choosing some element with a short representation. In Section 2.2 we introduce higher-order pushdown systems.

2.1. Ehrenfeucht-Fraïssé Games and First-Order Model Checking.

The equivalence $\equiv_\rho$ has a nice characterisation via *Ehrenfeucht-Fraïssé games.* Based on the work of Fraïssé [9], Ehrenfeucht [7] introduced these games which have become one of the most important tools for proving inexpressibility of properties in first-order logic. In this paper, we use a nonstandard application of Ehrenfeucht-Fraïssé game analysis to the FO model checking problem: strategies of Duplicator that only choose elements with small representations can be turned into a model checking algorithm. After briefly recalling the basic definitions, we explain this approach to model checking in detail. In the main part of this paper, we will see that this approach yields an FO model checking algorithm on the class of nested pushdown trees of level 2.

Definition 2.1. Let $\mathfrak{A}_1$ and $\mathfrak{A}_2$ be σ-structures. For tuples
$$\bar{a}^1 = a^1_1, a^1_2, \dots, a^1_m \in A_1^m \text{ and } \bar{a}^2 = a^2_1, a^2_2, \dots, a^2_m \in A_2^m$$
we write $\bar{a}^1 \mapsto \bar{a}^2$ for the map that maps a^1_i to a^2_i for all $1 \leq i \leq m$. In the n-round Ehrenfeucht-Fraïssé game on $\mathfrak{A}_1, a^1_1, a^1_2, \dots, a^1_m$ and $\mathfrak{A}_2, a^2_1, a^2_2, \dots, a^2_m$ for $a^j_i \in A_j$ there are two players, Spoiler and Duplicator, which play according to the following rules. The game is played for n rounds. The i-th round consists of the following steps.

(1) Spoiler chooses one of the structures, i.e., he chooses $j \in \{1, 2\}$.
(2) Then he chooses one of the elements of his structure, i.e., he chooses some $a^j_{m+i} \in A_j$.
(3) Now, Duplicator chooses some $a^{3-j}_{m+i} \in A_{3-j}$.

Having executed n rounds, Spoiler and Duplicator have chosen tuples
$$\bar{a}^1 := a^1_1, a^1_2, \dots, a^1_{m+n} \in A_1^{m+n} \text{ and } \bar{a}^2 := a^2_1, a^2_2, \dots, a^2_{m+n} \in A_2^{m+n}.$$
Duplicator wins the play if $f : \bar{a}^1 \mapsto \bar{a}^2$ is a partial isomorphism, i.e., if f satisfies

(1) $a^1_i = a^1_j$ if and only if $a^2_i = a^2_j$ for all $1 \leq i \leq j \leq m + n$, and
(2) for each $R_i \in \sigma$ of arity r the following holds: for $i_1, i_2, \dots, i_r$ numbers between 1 and $m + n$, $\mathfrak{A}_1, \bar{a}^1 \models R_i x_{i_1} x_{i_2} \dots x_{i_r}$ if and only if $\mathfrak{A}_2, \bar{a}^2 \models R_i x_{i_1} x_{i_2} \dots x_{i_r}$.

Lemma 2.2 ([9, 7])**.** *Let $\mathfrak{A}_1$, $\mathfrak{A}_2$ be structures and let $\bar{a}^1 \in \mathfrak{A}^n$, $\bar{a}^2 \in \mathfrak{A}_2$ be n-tuples. Duplicator has a winning strategy in the ρ-round Ehrenfeucht-Fraïssé game on $\mathfrak{A}_1, \bar{a}^1$ and $\mathfrak{A}_2, \bar{a}^2$ if and only if $\mathfrak{A}_1, \bar{a}^1 \equiv_\rho \mathfrak{A}_2, \bar{a}^2$.*

In this paper apply the analysis of Ehrenfeucht-Fraïssé games to the FO model checking problem. We use a variant of the notion of H-boundedness of Ferrante and Rackoff [8])The existence of certain restricted strategies in the game played on two identical copies of a structure yields an FO model checking algorithm.

We consider the game played on two copies of the same structure, i.e., the game on $\mathfrak{A}, \bar{a}^1$ and $\mathfrak{A}, \bar{a}^2$ with identical choice of the initial parameter $\bar{a}^1 = \bar{a}^2 \in \mathfrak{A}$. Of course, Duplicator has a winning strategy in this setting: she can copy each move of Spoiler. But we look for winning strategies with certain constraints. In our application the constraint is that Duplicator is only allowed to choose elements that are represented by short runs of higher-order pushdown systems, but the idea can be formulated more generally.

Definition 2.3. Let $\mathcal{C}$ be a class of structures. Assume that $S^{\mathfrak{A}}(m) \subseteq \mathfrak{A}^m$ is a subset of the m-tuples of the structure $\mathfrak{A}$ for each $\mathfrak{A} \in \mathcal{C}$ and each $m \in \mathbb{N}$. Set $S := (S^{\mathfrak{A}}(m))_{m \in \mathbb{N}, \mathfrak{A} \in \mathcal{C}}$. We call S a *constraint for Duplicator's strategy* and we say Duplicator has an *S-preserving* winning strategy if she has a strategy for each game played on two copies of $\mathfrak{A}$ for some $\mathfrak{A} \in \mathcal{C}$ with the following property. Let $\bar{a}^1 \mapsto \bar{a}^2$ be a position reached after m rounds where Duplicator used her strategy. If $\bar{a}^2 \in S^{\mathfrak{A}}(m)$ and Spoiler chooses some element in the first copy of $\mathfrak{A}$, then her strategy chooses an element a^2_{m+1} such that $\bar{a}^2, a^2_{m+1} \in S^{\mathfrak{A}}(m+1)$.

Remark 2.4. We write $S(m)$ for $S^{\mathfrak{A}}(m)$ if $\mathfrak{A}$ is clear from the context.

We now want to turn an S-preserving strategy of Duplicator into a model checking algorithm. The idea is to restrict the search for witnesses of existential quantifications to the sets defined by S. In order to obtain a terminating algorithm, S must be finitary in the sense of the following definition.

Definition 2.5. Given a class $\mathcal{C}$ of finitely represented structures, we call a constraint S for Duplicator's strategy *finitary* on $\mathcal{C}$, if for each $\mathfrak{A} \in \mathcal{C}$ we can compute a function $f_{\mathfrak{A}}$ such that for all $n \in \mathbb{N}$

- $S^{\mathfrak{A}}(n)$ is finite,
- there is a representation for each $\bar{a} \in S^{\mathfrak{A}}(n)$ in space $f_{\mathfrak{A}}(n)$, and
- $\bar{a} \in S^{\mathfrak{A}}(n)$ is effectively decidable.

Recall the following fact: if Duplicator uses a winning strategy in the n round game, her choice in the $(m+1)$-st round is an element a^2_{m+1} such that $\mathfrak{A}, \bar{a}^1, a^1_{m+1} \equiv_{n-m-1} \mathfrak{A}, \bar{a}^2, a^2_{m+1}$. Hence, if Duplicator has an S-preserving winning strategy, then for every formula $\varphi(x_1, x_2, \dots, x_{m+1}) \in \mathrm{FO}_{n-m-1}$ and for all $\bar{a} \in A^m$ with $\bar{a} \in S(m)$ the following holds:

$$\begin{aligned}
&\text{there is an element } a \in A \text{ such that } \bar{a}, a \in S(m+1) \text{ and } \mathfrak{A}, \bar{a}, a \models \varphi\\
\text{iff } &\text{there is an element } a \in A \text{ such that } \mathfrak{A}, \bar{a}, a \models \varphi\\
\text{iff } &\mathfrak{A}, \bar{a} \models \exists x_{m+1}\varphi.
\end{aligned}$$

Let us fix some class $\mathcal{C}$ of finitely represented structures and let S be some finitary constraint on $\mathcal{C}$ such that Duplicator has an S-preserving winning strategy. In this case the alternating Turing machine described in Algorithm 1 solves the FO model checking problem on $\mathcal{C}$. [2] Its running time on input $(\mathfrak{A}, \varphi)$ is $f_{\mathfrak{A}}(|\varphi|) \cdot |\varphi|$ and it uses at most $|\varphi|$ many alternations.

2.2. **Higher-Order Pushdown Systems.** In order to define what a level n nested pushdown tree is, we first have to introduce pushdown systems of level n (n-PS).

An n-PS can be seen has a finite automaton with access to an n-fold nested stack structure. This generalises the notion of a pushdown system by replacing a single stack with a structure that is a stack of stacks of stacks ... of stacks. A nested stack of level n can be manipulated by level l push and pop operations for each level $l \leq n$. For $2 \leq l \leq n$, the level l push operation clone_l duplicates the topmost entry of the topmost level l stack. The level 1 push operation push_σ writes the symbol σ on top of the topmost level 1 stack. For $1 \leq l \leq n$, the level l pop operation pop_l removes the topmost entry of the topmost level l stack.

For some alphabet Σ, we inductively define the *set of level n stacks over* Σ (n-stacks), denoted by Σ^{+n} as follows. Let $\Sigma^{+1} := \Sigma^{+}$ denote the set of all nonempty finite words over alphabet Σ. We then define $\Sigma^{+(n+1)} := (\Sigma^{+n})^{+}$.

Let us fix an $(n+1)$-stack $s \in \Sigma^{+(n+1)}$. This stack s consists of an ordered list $s_1, s_2, \dots, s_m \in \Sigma^{+n}$. If we want to state this list explicitly, we separate them by colons writing $s = s_1 : s_2 : \dots : s_m$. By $|s|$ we denote the number of n-stacks s consists of, i.e., $|s| = m$. We call $|s|$ the *width* of s. We also use the notion of the *height* of s. This is $\mathrm{hgt}(s) := \max\{|s_i| : 1 \leq i \leq m\}$, i.e., the width of the widest n-stack occurring in s.

Let s' be an $(n+1)$-stack such that $s' = s'_1 : s'_2 : \dots : s'_l \in \Sigma^{+(n+1)}$. We write $s : s'$ for the concatenation $s_1 : s_2 : \dots : s_m : s'_1 : s'_2 : \dots : s'_l$.

If $s \in \Sigma^{+(n-1)}$, we denote by $[s]$ the n-stack that only consists of a list of length 1 that contains s. We regularly omit the brackets if no confusion arises.

Let Σ be some finite alphabet with a distinguished bottom-of-stack symbol $\bot \in \Sigma$. The *initial stack* $\bot_l$ of level l over Σ is inductively defined by $\bot_1 := \bot$ and $\bot_n := [\bot_{n-1}]$.

[2]Without loss of generality we assume first-order formulas to be generated from atomic and negated atomic formulas only by means of disjunction $\vee$, conjunction $\wedge$, existential quantification $\exists$ and universal quantification $\forall$.

```
SModelCheck
Input: a structure 𝔄 , a formula φ ∈ FO_ρ, an assignment x̄ ↦ ā for tuples x̄, ā of
       arity m such that ā ∈ S(m)
if φ is an atom then
 | if 𝔄, ā ⊨ φ(x̄) then accept else reject;
end
if φ = φ1 ∨ φ2 then
 | if SModelCheck(𝔄, ā, φ1) = accept then accept else
 |  | if SModelCheck(𝔄, ā, φ2)= accept then accept else reject;
 | end
end
if φ = φ1 ∧ φ2 then
 | if SModelCheck(𝔄, ā, φ1) = accept then
 |  | if SModelCheck(𝔄, ā, φ2)= accept then accept else reject;
 | end
 | else reject;
end
if φ = ¬φ1 then
 | if SModelCheck(𝔄, ā, φ1)= accept then reject else accept;
end
if φ = ∃xφ1(x̄, x) then
 | guess an a ∈ 𝔄 with ā, a ∈ S(m + 1) and SModelCheck(𝔄, āa, φ1);
end
if φ = ∀x_iφ1 then
 | universally choose an a ∈ 𝔄 with ā, a ∈ S(m + 1) and SModelCheck(𝔄, āa, φ1);
end
```

Algorithm 1: FO-model checking on S-preserving structures

Before we formally define the stack operations, we introduce an auxiliary function top_k that returns the topmost entry of the topmost k-stack. Let $s = s_1 : s_2 : \cdots : s_n \in \Sigma^{+l}$ be some stack and let $1 \leq k \leq l$. We define the *topmost level* $k-1$ *stack of* s to be

$$\mathrm{top}_k(s) := \begin{cases} s_n & \text{if } k = l, \\ \mathrm{top}_k(s_n) & \text{otherwise.} \end{cases}$$

Definition 2.6. For $s = s_1 : s_2 : \cdots : s_n \in \Sigma^{+l}$, for $\sigma \in \Sigma \setminus \{\bot\}$, for $1 \leq k \leq l$ and for $2 \leq j \leq l$, we define the stack operations

$$\mathrm{clone}_j(s) := \begin{cases} s_1 : s_2 : \cdots : s_{n-1} : s_n : s_n & \text{if } j = l \geq 2, \\ s_1 : s_2 : \cdots : s_{n-1} : \mathrm{clone}_j(s_n) & \text{otherwise.} \end{cases}$$

$$\mathrm{push}_\sigma(s) := \begin{cases} s\sigma & \text{if } l = 1, \\ s_1 : s_2 : \cdots : s_{n-1} : \mathrm{push}_\sigma(s_n) & \text{otherwise.} \end{cases}$$

$$\mathrm{pop}_k(s) := \begin{cases} s_1 : s_2 : \cdots : s_{n-1} : \mathrm{pop}_k(s_n) & \text{if } k < l, \\ s_1 : s_2 : \cdots : s_{n-1} & \text{if } k = l, n > 1, \\ \text{undefined} & \text{otherwise, i.e.,} k = l, n = 1, \end{cases}$$

and $\mathrm{id}(s) := s$.

The *set of level l operations* is denoted by OP_l.

For $2 \leq i \leq n$ and $\sigma \in \Sigma$, we call push_σ a push of level 1 and clone_i a push of level i.

For stacks s, s' we write $s \leq s'$ and say s is a substack of s' if s is generated from s' by application of a sequence of pop operations (of possibly different levels). Note that on 1-stacks, i.e., on words, $\leq$ coincides with the usual prefix relation.

Having defined l-stacks, we present pushdown systems of level l.

Definition 2.7. A *pushdown system* of level l (l-PS) is a tuple $\mathcal{S} = (Q, \Sigma, \Delta, q_0)$ where Q is a finite set of states, Σ a finite stack alphabet with a distinguished bottom-of-stack symbol $\bot \in \Sigma$, $q_0 \in Q$ the initial state, and $\Delta \subseteq Q \times \Sigma \times Q \times \mathrm{OP}_l$ the transition relation.

An *l-configuration* is a pair (q, s) where $q \in Q$ and $s \in \mathrm{Stacks}_l(\Sigma)$. For $q_1, q_2 \in Q$, $s, t \in \mathrm{Stacks}_l(\Sigma)$ and for $\delta = (q_1, \sigma, q_2, op) \in \Delta$, we define the δ-relation $\vdash^\delta$ as follows. Set $(q_1, s) \vdash^\delta (q_2, t)$ if $\mathrm{op}(s) = t$ and $\mathrm{top}_1(s) = \sigma$. We call $\vdash := \bigcup_{\delta \in \Delta} \vdash^\delta$ the transition relation of $\mathcal{S}$.

Definition 2.8. Let $\mathcal{S}$ be a l-PS. A *run* ρ of $\mathcal{S}$ is a sequence of configurations that are connected by transitions, i.e., a sequence

$$c_0 \vdash^{\delta_1} c_1 \vdash^{\delta_2} c_2 \vdash^{\delta_3} \cdots \vdash^{\delta_n} c_n.$$

We also write $\rho(i) := c_i$ for the i-th configuration occurring within ρ. We call n the *length* of ρ and set $\mathrm{len}(\rho) := n$. If some run π is an initial segment of the run ρ, we write $\pi \leq \rho$. We write $\pi \prec \rho$ if π is a proper initial segment of ρ.

If π is a run from c_0 to c_1 and ρ is a run from c_1 to c_2, then we denote by $\pi \circ \rho$ the composition of π and ρ which is the run from c_0 to c_2 defined by $c_0 \overset{\pi}{\cdots} c_1 \overset{\rho}{\cdots} c_2$.

3. The Nested Pushdown Tree Hierarchy

Generalising the definition of nested pushdown trees (cf. [1]), we define a hierarchy of higher-order nested pushdown trees. A nested pushdown tree is the unfolding of the configuration graph of a pushdown system expanded by a new relation (called *jump relation*) which connects each push operation with the corresponding pop operations. Since higher-order pushdown systems have push and pop operations for each stack level, there is no unique generalisation of this concept to trees generated by higher-order pushdown systems. We choose the following version: we connect corresponding push and pop operations of the highest stack level. This choice ensures that the jump edges form a well-nested relation. The exact definition of a higher-order nested pushdown tree is as follows.

Definition 3.1. Let $\mathcal{N} = (Q, \Sigma, q_0, \Delta)$ be an n-PS. The *level n nested pushdown tree* (n-NPT) $\mathfrak{N} := \mathrm{NPT}(\mathcal{N})$ is the unfolding of the pushdown graph of $\mathcal{N}$ from its initial configuration expanded by the *jump relation* $\curvearrowright$ which connects each level n push operation with all corresponding pop_n operations, i.e., for runs ρ_1, ρ_2 of $\mathcal{N}$ we have $\rho_1 \curvearrowright \rho_2$ if ρ_2 decomposes as $\rho_2 = \rho_1 \circ \rho$ for some run ρ from (q, s) to (q', s) of length m such that

$\rho(0)$ and $\rho(1)$ are connected by a level n push operation,

$\rho(m-1)$ and $\rho(m)$ are connected by a level n pop operation, and

$\rho(i) \neq (\hat{q}, s)$ for all $1 \leq i < m$ and all $\hat{q} \in Q$.

Remark 3.2. Note that for all $\rho_2 \in \mathrm{NPT}(\mathcal{N})$ there is at most one $\rho_1 \in \mathrm{NPT}(\mathcal{N})$ with $\rho_1 \curvearrowright \rho_2$, but for each $\rho_1 \in \mathrm{NPT}(\mathcal{N})$ there may be infinitely many $\rho_2 \in \mathrm{NPT}(\mathcal{N})$ with $\rho_1 \curvearrowright \rho_2$.

For each $l < l'$ the class of nested pushdown trees of level l are uniformly first-order interpretable in the class of nested pushdown trees of level l'. If $l = 1$, one just replaces any push_σ transition by a $\mathrm{clone}_{l'}$ transition followed by a push_σ transition. Furthermore, one replaces each pop_1 transition by a $\mathrm{pop}_{l'}$ transition. In all other cases, we just replace clone_l and pop_l by $\mathrm{clone}_{l'}$ and $\mathrm{pop}_{l'}$.

The hierarchy of higher-order nested pushdown trees is a hierarchy strictly extending the hierarchy of trees generated by higher-order pushdown systems. Furthermore, it is first-order interpretable in the collapsible pushdown graph hierarchy.[3] We now sketch the proof of this claim. Fix an l-PS $\mathcal{N}$. $\mathcal{N}$ generates an l-NPT $\mathfrak{N}$. Each node of $\mathfrak{N}$ represents a run of $\mathcal{N}$ starting in the initial configuration. A run can be seen as a list of configurations. This is a list of pairs of states and stacks. Pushing the state on top of the stack, a run can be represented as a list of l-stacks. Let $s_1, s_2, s_3, \dots, s_n$ be the stacks representing some run. Then $s_1 : s_2 : \dots : s_n$ is an $(l+1)$-stack representing the run. In this representation, an edge in the l-NPT corresponds to the extension of $s_1 : s_2 : \dots : s_n$ to a list $s_1 : s_2 : \dots : s_n : s_{n+1}$ where s_{n+1} is generated from s_n by removing the state written on top of s_n, applying a stack operation and writing the new final state on top of the stack. Hence, we can use this representation and define a level $(l+1)$-PS $\mathcal{S}$ such that the tree generated by $\mathcal{N}$ is first-order interpretable in the configuration graph of $\mathcal{S}$. We interpret each edge of $\mathrm{NPT}(\mathcal{N})$ as a path of length 4 in the configuration graph of $\mathcal{S}$. Such a path performs the operations clone_n - pop_1 - op - push_q for some level l operation op. Replacing $\mathcal{S}$ by a certain collapsible pushdown system $\hat{\mathcal{S}}$, we can generate the same graph but with additional collapse-transitions that form exactly the reversals of the jump edges of $\mathfrak{N}$. This means that if $a, b \in \mathfrak{N}$ such that $a \curvearrowright b$, and a', b' are the representatives of a and b in the configuration graph of $\hat{\mathcal{S}}$, then there is a collapse edge from b to a. A detailed proof of this claim can be found in [14].

[3]The hierarchy of collapsible pushdown graphs is the class of configuration graphs of *collapsible pushdown systems*. Collapsible pushdown systems of level l are defined analogously to l-PS but with transitions that may also use a stack-operation called collapse. This new operation summarises several pop operations. For a detailed definition, see [10] or [14].

3.1. **Towards Model Checking on Level 2 NPT.** In the following, we develop an FO model checking algorithm on 2-NPT. In fact, we prove that the general approach via the dynamic-small-witness property developed in Section 2.1 is applicable in this case. In other words, we prove that we can compute a finitary constraint for Duplicator's strategy on an arbitrary 2-NPT $\mathfrak{N}$.

Fix some 2-PS $\mathcal{N}$ of level 2 and set $\mathfrak{N} := \mathrm{NPT}(\mathcal{N})$. We show the following. If $\mathfrak{N}, \bar{\rho} \models \exists x \varphi$ for some formula $\varphi \in \mathrm{FO}$, then there is a short witness $\rho \in \mathfrak{N}$ for this existential quantification. Here, the length of an element is given by the length of the run representing this element. We consider a run to be short if its size is bounded in terms of the length of the runs representing the parameters $\bar{\rho}$.

We stress that locality arguments in the spirit of Gaifman's theorem do not apply in this setting: the jump edges tend to make the diameter of 2-NPT small.

The rough picture of our proof is as follows. We analyse the α-round Ehrenfeucht-Fraïssé game on two copies of $\mathfrak{N}$ and show that Duplicator has a restricted winning-strategy. Our main technical tool is the concept of *relevant ancestors*. For each element of $\mathfrak{N}$, the relevant l-ancestors are a finite set of initial subruns of this element. Intuitively, some run ρ' is a relevant l-ancestors of a run ρ if it is an ancestor of ρ which is connected to ρ via a path of length up to l that witnesses the fact that ρ' is an ancestor of ρ. It turns out that there are at most 4^l such ancestors. Surprisingly, the set of 2^l-ancestors characterises the FO_l-type of ρ. Thus, Duplicator has a winning strategy choosing small runs if for every element of $\mathfrak{N}$ there is a small one that has an isomorphic set of relevant ancestors.

The analysis of relevant ancestors reveals that a relevant ancestor ρ_1 is connected to the next one, say ρ_2, by either a single transition or by a run π of a certain kind. This run π satisfies the following conditions: ρ_2 decomposes as $\rho_2 = \rho_1 \circ \pi$, the initial stack of π is $s : w$ where s is some stack and w is some word. The final stack of π is $s : w : v$ for some word v and π does never pass a proper substack of $s : w$.

Due to this result, a typical set of relevant ancestors is of the form

$$\rho_1 \prec \rho_2 \prec \rho_3 \prec \cdots \prec \rho_m = \rho,$$

where ρ_{n+1} extends ρ_n by either one transition or by a run that extends the last stack of ρ_n by a new word v. If we want to construct a short run ρ' with isomorphic relevant ancestor set, we have to provide short runs

$$\rho'_1 \prec \rho'_2 \prec \rho'_3 \prec \cdots \prec \rho'_m = \rho'$$

where ρ'_{n+1} extends ρ'_n in exactly the same manner as ρ_{n+1} extends ρ_n.

We first concentrate on one step of this construction. Assume that ρ_1 ends in some configuration $(q, s : w)$ and ρ_2 extends ρ_1 by a run creating the stack $s : w : v$. How can we find another stack s' and words w', v' such that there is a run ρ'_1 to $(q, s' : w')$ and a run ρ'_2 that extends ρ'_1 by a run from $(q, s' : w')$ to the stack $s' : w' : v'$?

We introduce a family of equivalence relations on words that preserves the existence of such runs. If we find some w' that is equivalent to w with respect to the i-th equivalence relation, then for each run from $s : w$ to $s : w : v$ there is a run from $s' : w'$ to $s' : w' : v'$ for v and v' equivalent with respect to the $(i-1)$-st equivalence relation.

Let us explain these equivalence relations. Let ρ_1 be a run to some stack $s : w$ and let ρ_2 be a run that extends ρ_1 and ends in a stack $s : w : v$. We can prove that this extension is of the form $\mathrm{op}_n \circ \lambda_n \circ \mathrm{op}_{n-1} \circ \lambda_{n-1} \circ \cdots \circ \mathrm{op}_1 \circ \lambda_1$ where the λ_i are loops, i.e., runs that start and end with the same stack and $\mathrm{op}_n, \mathrm{op}_{n-1}, \dots, \mathrm{op}_1$ is the minimal sequence generating $s : w : v$

from $s:w$. Thus, we are especially interested in the loops of each prefix $\mathrm{pop}_1^k(w)$ of w and each prefix $\mathrm{pop}_1^k(w')$ of w'. For this purpose we consider the word models of w and w' enriched by information on runs between certain prefixes of w or w'. Especially, each prefix is annotated with the number of possible loops of each prefix. w and w' are equivalent with respect to the first equivalence relation if the FO_k-types of their enriched word structures coincide. The higher-order equivalence relations are then defined as follows. We colour every element of the word model of some word w by the equivalence class of the corresponding prefix with respect to the $(i-1)$-st equivalence relation. For the i-th equivalence relation we compare the FO_k-types of these coloured word models. This means that two words w and w' are equivalent with respect to the i-th equivalence relation if the FO_k-types of their word models expanded by predicates encoding the $(i-1)$-st equivalence class of each prefix coincide.

This iteration of equivalence of prefixes leads to the following result. Let w and w' be equivalent with respect to the i-th relation. Then we can transfer runs creating i words in the following sense: if ρ is a run creating $w:v_1:v_2:\dots:v_i$ from w, then there is a run ρ' creating $w':v_1':v_2':\dots:v_i'$ from w' such that v_k and v_k' are equivalent with respect to the $(i-k)$-th relation. This property then allows us to construct isomorphic relevant ancestors for a given set of relevant ancestors of some run ρ. We only have to start with a stack $s':w'$ such that w' is i-equivalent to the topmost word of the minimal element of the relevant ancestors of ρ for some large $i\in\mathbb{N}$.

This observation reduces the problem of constructing runs with isomorphic relevant ancestors to the problem of finding runs whose last configurations have equivalent topmost words (with respect to the i-th equivalence relation for some sufficiently large i) such that one of these runs is always short.

We solve this problem by developing shrinking constructions that allow the preservation of the equivalence class of the topmost word of the final configuration of a run while shrinking the length of the run.

Putting all these results together, for every nested pushdown tree of level 2, we can compute a finitary constraint S such that Duplicator has an S-preserving strategy. This shows that the general model checking algorithm from Section 2.1 solves the the FO model checking problem on 2-NPT.

3.2. **Outline of the Proof Details.** In the next section, we discuss the theory of *loops* first developed in [13]. We also develop shrinking lemmas for long runs. Then we introduce the central notion of *relevant ancestors* and develop some basic theory concerning these sets in Section 5. In Section 5.2 we define equivalence relations on words and trees which can be used to construct runs with isomorphic relevant ancestors. In Section 6, we first lift these equivalences on stacks to equivalences on tuples of elements of 2-NPT by pairwise comparison of the equivalence type of the topmost stacks of each relevant ancestor of each element in the tuples. We then prove that the preservation of equivalence classes of relevant ancestors is a winning strategy in the Ehrenfeucht-Fraïssé game. Furthermore, Duplicator can always choose small representatives. Finally, in Section 7 we derive an FO model checking algorithm on 2-NPT. In that section we also show that this algorithm restricted to the class of all 1-NPT is in 2-EXPSPACE. Unfortunately, we do not obtain any complexity bounds for the algorithm on the class of 2-NPT.

4. Shrinking Lemmas for Runs of 2-PS

In this section, we analyse runs of 2-PS between certain configurations. Especially, we look at runs starting at the initial configuration and at runs extending its starting stack s by some word $w \in \Sigma^+$, i.e., runs starting in s and ending in $s : w$ that do not visit substacks of s. If there is a run ρ of one of these types, we will see that there is also a short run ρ' with the same initial and final configuration as ρ. "short" means that $\text{len}(\rho')$ is bounded by some function depending on the pushdown system, on the width and height of s and on the length of w. We first developed this theory in [13]. We briefly recall this theory and sketch the main proofs. For a detailed presentation, see [14].

We first introduce the concept of milestones and generalised milestones of a 2-stack s. A stack t is a generalised milestone of s if every run from the initial configuration to s has to pass t. Milestones are those generalised milestones that are substacks of s. Generalised milestones of s induce a natural decomposition of any run from the initial configuration to some configuration with stack s. Due to a result of Carayol (that we state soon), generalised milestones can be characterised as follows.

Definition 4.1. Let $s = w_1 : w_2 : \cdots : w_k$ be a 2-stack. We define the set of *generalised milestones of* s, denoted by $\text{GMS}(s)$, as follows. A stack m is in $\text{GMS}(s)$ if

$$m = w_1 : w_2 : \cdots : w_i : v_{i+1} \text{ where } 0 \leq i < k,$$
$$w_i \sqcap w_{i+1} \leq v_{i+1} \text{ and}$$
$$v_{i+1} \leq w_i \text{ or } v_{i+1} \leq w_{i+1}.$$

where $w \sqcap v$ denotes the maximal common prefix of w and v. If $v_{i+1} \leq w_{i+1}$, we call m a milestone of s. The set of milestones of s is denoted by $\text{MS}(s)$.

Remark 4.2. Note that $|\text{GMS}(s)| \leq 2 \cdot \text{hgt}(s) \cdot |s|$.

In order to show that this definition fits our informal description, we define the notions of loops and returns. Loops appear as a central notion in our formulation of Carayol's result. Moreover, the theory of loops and returns is also the key ingredient to our shrinking constructions. We will generate short runs from longer ones by replacing long loops by shorter ones.

Definition 4.3. Let $s = t : w$ be some stack with topmost word w and $q, q' \in Q$. A run λ from (q, s) to (q', s) is called *loop* if it does not pass t. It is called a *high loop* if it additionally does not pass $\text{pop}_1(s)$. A run ρ from $(q, t : w)$ to (q', t) is called *return* if it visits t only in its final configuration.

The following lemma of Carayol shows that generalised milestones and loops play a crucial role in understanding the existence of runs.

Lemma 4.4 ([5]). *For each stack s there is a minimal sequence of operations $\text{op}_1, \text{op}_2, \ldots, \text{op}_n \in \{\text{push}_\sigma, \text{pop}_1, \text{clone}_2\}$ such that $s = \text{op}_n(\text{op}_{n_1}(\ldots(\text{op}_1(\bot_2))\ldots))$.*

For each $0 \leq j \leq n$, the stack $\text{op}_j(\text{op}_{j-1}(\ldots \text{op}_0(\bot_2)))$ is a generalised milestone of s and every generalised milestone is generated by such a sequence.

Furthermore, every run ρ from $(q_0, \bot_2)$ to s passes all generalised milestones of s. More precisely, ρ decomposes as $\rho = \lambda_n \circ \rho_n \circ \lambda_{n-1} \circ \rho_{n-1} \circ \ldots \circ \lambda_1 \circ \rho_1 \circ \lambda_0$, where ρ_i is a run of length 1 that performs the operation op_i and λ_i is a loop of the i-th generalised milestone of s.

Thus, loops play a crucial role in understanding the existence of runs from one configuration to another. Moreover, returns and loops of smaller stacks provide a natural decomposition of loops. Hence, counting the number of loops and returns of certain stacks can be used in order to count the number of runs between certain configurations. We define the following notation concerning counting loops and returns up to some threshold $k \in \mathbb{N}$.

Definition 4.5. Let $\mathcal{S} = (Q, \Sigma, q_I, \Delta)$ be a 2-PS. Let $\#\mathrm{Ret}^k_{\mathcal{S}}(s) : Q \times Q \to \{0, 1, \dots, k\}$ be the function that assigns (q, q') to the number of returns from (q, s) to $(q', \mathrm{pop}_2(s))$ up to threshold k. Analogously, let $\#\mathrm{Loop}^k_{\mathcal{S}}(s)$ and $\#\mathrm{HLoop}^k_{\mathcal{S}}(s)$ count the loops and high loops, respectively, from (q, s) to (q', s) up to threshold k.

If $\mathcal{S}$ is clear from the context, we omit it and write $\#\mathrm{Loop}^k$ for $\#\mathrm{Loop}^k_{\mathcal{S}}$, etc.

Remark 4.6. Note that loops and returns of some stack $s : w$ never look into s. Hence, $\#\mathrm{Loop}^k(s : w) = \#\mathrm{Loop}^k(t : w)$ for every word w and arbitrary stacks s and t. If s and t are both nonempty stacks, the analogous statement holds for $\#\mathrm{Ret}^k$. Thus, we will write $\#\mathrm{Loop}^k(w)$ for $\#\mathrm{Loop}^k(s : w)$ where s is an arbitrary stack. Analogously $\#\mathrm{Ret}^k(w) := \#\mathrm{Ret}^k(s : w)$ for any nonempty stack s.

The number of loops and returns of some stack s inductively depend on the number of loops and returns of its substacks. In preparation of the proof of this observation, we recall the notion of *stack replacement* introduced in [3].

Definition 4.7. For some level 2 stack t and some substack $s \leq t$ we say that s is a *prefix* of t and write $s \trianglelefteq t$, if there are $n \leq m \in \mathbb{N}$ such that $s = w_1 : w_2 : \dots : w_{n-1} : w_n$ and $t = w_1 : w_2 \cdots : w_{n-1} : v_n : v_{n+1} : \dots : v_m$ such that $w_n \leq v_j$ for all $n \leq j \leq m$. This means that s and t agree on the first $|s| - 1$ words and the last word of s is a prefix of all other words of t. We extend this definition to runs by writing $s \trianglelefteq \rho$ if ρ is some run and $s \trianglelefteq \rho(i)$ for all $i \in \mathrm{dom}(\rho)$.

Let s, t, u be 2-stacks such that $s \trianglelefteq t$. Assume that

$$\begin{aligned} s &= w_1 : w_2 : \dots : w_{n-1} : w_n, \\ t &= w_1 : w_2 \cdots : w_{n-1} : v_n : v_{n+1} : \dots : v_m, \text{ and} \\ u &= x_1 : x_2 : \dots : x_p \end{aligned}$$

for numbers $n, m, p \in \mathbb{N}$ with $n \leq m$. For $n \leq i \leq m$, let $\hat{v}_i$ be the unique word such that $v_i = w_n \circ \hat{v}_i$. Set

$$t[s/u] := x_1 : x_2 : \dots : x_{p-1} : (x_p \circ \hat{v}_n) : (x_p \circ \hat{v}_{n+1}) : \dots : (x_p \circ \hat{v}_m)$$

and call $t[s/u]$ *the stack obtained from t by replacing the prefix s by u.* For $c = (q, t)$ a configuration, set $c[s/u] := (q, t[s/u])$.

Lemma 4.8 ([3])**.** *Let ρ be a run of some 2-PS $\mathcal{S}$. Let $s, u \in \Sigma^{+2}$ be stacks such that $s \trianglelefteq \rho$ and* $\mathrm{top}_1(u) = \mathrm{top}_1(s)$*. Then the function $\rho[s/u]$ defined by $\rho[s/u](i) := \rho(i)[s/u]$ is a run of $\mathcal{S}$.*

We extend the notion of prefix replacement to runs that are s-prefixed at the beginning and at the end and that never visit the substack $\mathrm{pop}_2(s)$. Such a run may contain "holes", i.e., parts that are not prefixed by s. We show that these holes are always loops or returns. Thus, we can replace a prefix by another one if these two share the same types of loops and returns. The following lemma prepares this new kind of prefix replacement.

Lemma 4.9. *Let $\mathcal{N}$ be some 2-PS and let ρ be a run of $\mathcal{N}$ of length n. Let s be a stack with topmost word $w := \mathrm{top}_2(s)$ such that*

$$s \trianglelefteq \rho(0), s \trianglelefteq \rho(n), \text{ and } |s| \leq |\rho(i)| \text{ for all } 0 \leq i \leq n.$$

There is a unique sequence $0 = i_0 \leq j_0 < i_1 \leq j_1 < \cdots < i_{m-1} \leq j_{m-1} < i_m \leq j_m = n$ such that

1. *$s \trianglelefteq \rho{\upharpoonright}_{[i_k, j_k]}$ for all $0 \leq k \leq m$ and*
2. *$\mathrm{top}_2(\rho(j_k+1)) = \mathrm{pop}_1(w)$, $\rho{\upharpoonright}_{[j_k,i_{k+1}]}$ is either a loop or a return, and $\rho{\upharpoonright}_{[j_k,i_{k+1}]}$ does not visit the stack of $\rho(j_k)$ between its initial configuration and its final configuration for all $0 \leq k < m$.*

Proof. If $s \trianglelefteq \rho$, then we set $m := 0$ and we are done.

Otherwise, we proceed by induction on the length of ρ. There is a minimal position j_0+1 such that $s \ntrianglelefteq \rho(j_0+1)$. By assumption on s, $\rho(j_0+1) \neq \mathrm{pop}_2(s)$. Thus, $\mathrm{top}_2(\rho(j_0)) = w$ and $\mathrm{top}_2(\rho(j_0+1)) = \mathrm{pop}_1(w)$. Now, let $i_1 > j_0$ be minimal such that $s \trianglelefteq \rho(i_1)$. Concerning the stack at i_1 there are the following possibilities.

1. If $\rho(i_1) = \mathrm{pop}_2(\rho(j_0))$ then $\rho{\upharpoonright}_{[j_0,i_1]}$ is a return.
2. Otherwise, the stacks of $\rho(j_0)$ and $\rho(i_1)$ coincide whence $\rho{\upharpoonright}_{[j_0,i_1]}$ is a loop (note that between j_0 and i_1 the stack $\mathrm{pop}_2(\rho(j_0))$ is never visited due to the minimality of i_1 and due to assumption 3).

$\rho{\upharpoonright}_{[i_1,n]}$ is shorter than ρ. Thus, it decomposes by induction hypothesis and the lemma follows immediately. ◻

This lemma gives rise to the following extension of the prefix replacement for prefixes s, u where s and u share similar loops and returns.

Definition 4.10. Let s be some stack and ρ be a run of some 2-PS $\mathcal{N}$ such that $s \trianglelefteq \rho(0)$, $s \trianglelefteq \rho(\mathrm{len}(\rho))$ and $|s| \leq |\rho(i)|$ for all $i \in \mathrm{dom}(\rho)$. Let u be some stack such that $\mathrm{top}_1(u) = \mathrm{top}_1(s)$, $\#\mathrm{Loop}^1(u) = \#\mathrm{Loop}^1(s)$ and $\#\mathrm{Ret}^1(u) = \#\mathrm{Ret}^1(s)$.

Let $0 = i_0 \leq j_0 < i_1 \leq j_1 < \cdots < i_{m-1} \leq j_{m-1} < i_m \leq j_m = \mathrm{len}(\rho)$ be the sequence corresponding to ρ in the sense of the previous lemma. We set $(q_k, s_k) := \rho(j_k)$ and $(q'_k, s'_k) := \rho(i_{k+1})$. By definition, $\rho{\upharpoonright}_{[j_k,i_{k+1}]}$ is a loop or a return from (q_k, s_k) to (q'_k, s'_k) and $\mathrm{top}_2(s_k) = \mathrm{top}_2(s)$ and $s \trianglelefteq s_k$. Thus, $\mathrm{top}_2(s_k[s/u]) = \mathrm{top}_2(u)$. Since $\#\mathrm{Ret}^1(u) = \#\mathrm{Ret}^1(s)$ and $\#\mathrm{Loop}^1(u) = \#\mathrm{Loop}^1(s)$, there is a run from $(q_k, s_k[s/u])$ to $(q'_{k+1}, s'_{k+1}[s/u])$. We set ρ_k to be the length-lexicographically shortest run from $(q_k, s_k[s/u])$ to $(q'_{k+1}, s'_{k+1}[s/u])$.[4]

Then we define the run $\rho[s/u]$ by

$$\rho[s/u] := \rho{\upharpoonright}_{[i_0,j_0]}[s/u] \circ \rho_0 \circ \rho{\upharpoonright}_{[i_1,j_1]}[s/u] \circ \rho_1 \circ \cdots \circ \rho_{m-1} \circ \rho{\upharpoonright}_{[i_m,j_m]}[s/u].$$

Remark 4.11. Note that $\rho[s/u]$ is a well-defined run from $\rho(0)[s/u]$ to $\rho(\mathrm{len}(\rho))[s/u]$.

Next, we turn to the analysis of loops and returns. In this part, we show that for each 2-PS there is a function relating the height of the topmost word of a stack with a bound on the length of the shortest loops and returns of this stack.

First, we characterise loops and returns in terms of loops and returns of smaller stacks. The proof is completely analogous to the proof of Lemma 4.9. Every loop or return of some stack s decomposes into parts that are prefixed by s and parts that are returns or loops of stacks with topmost word $\mathrm{pop}_1(\mathrm{top}_2(s))$. This observation gives rise to the observation that the number of loops and returns of a stack s only depend on its topmost symbol and the number of loops and returns of $\mathrm{pop}_1(s)$.

[4] We assume Δ to be equipped with some fixed but arbitrary linear order.

Lemma 4.12. *Let ρ be some return from some stack $s:w$ to s of length n. Then there is a unique sequence $0 = i_0 \leq j_0 < i_1 \leq j_1 < \cdots < i_{m-1} \leq j_{m-1} < i_m \leq j_m = n$ such that the following holds.*

(1) $s:w \trianglelefteq \rho\restriction_{[i_k, j_k - 1]}$ *for all* $0 \leq k \leq m$ *and*
(2) $\rho\restriction_{[j_k, i_{k+1}]}$ *is a return from some stack* $s' : \mathrm{pop}_1(w)$ *to* s'.

Let λ be some loop of length n starting and ending in some stack $s:w$. Then there is a unique sequence $0 = i_0 \leq j_0 < i_1 \leq j_1 < \cdots < i_{m-1} \leq j_{m-1} < i_m \leq j_m = n$ such that the following holds.

(1) $s:w \trianglelefteq \lambda\restriction_{[i_k, j_k]}$ *for all* $0 \leq k \leq m$ *and*
(2) $\lambda\restriction_{[j_k+1, i_{k+1}]}$ *is either a return starting in some stack* $s' : \mathrm{pop}_1(w)$ *or it is a loop of* $\mathrm{pop}_1(s:w)$ *followed by a* $\mathrm{push}_{\mathrm{top}_1(w)}$ *transition. In this case* $\lambda(j_k + 1)$ *is the first and* $\lambda(i_{k+1} - 1)$ *is the last occurrence of* $\mathrm{pop}_1(s:w)$ *in* λ.

This lemma gives rise to the inductive computation of the number of loops and returns of a stack.

Proposition 4.13. *Given a 2-PS $\mathcal{S}$, $\#\mathrm{Loop}^k(w)$, $\#\mathrm{HLoop}^k(w)$ and $\#\mathrm{Ret}^k(w)$ only depend on $\mathrm{top}_1(w)$, $\mathrm{pop}_1(\mathrm{top}_1(w))$, $\#\mathrm{Loop}^k(\mathrm{pop}_1(w))$, and $\#\mathrm{Ret}^k(\mathrm{pop}_1(w))$.*

sketch. We only consider the return case and prove the following claim. Let $s:w$ and $s':w'$ be stacks such that s and s' are nonempty, $\mathrm{top}_1(w) = \mathrm{top}_1(w')$, and $\#\mathrm{Ret}^k(\mathrm{pop}_1(w)) = \#\mathrm{Ret}^k(\mathrm{pop}_1(w'))$. The number of returns of $s:w$ and of $s':w'$ coincide, i.e., $\#\mathrm{Ret}^k(w) = \#\mathrm{Ret}^k(w')$.

For reasons of simplicity, we assume that $R(q,q') \coloneqq \#\mathrm{Ret}^k(\mathrm{pop}_1(w))(q,q') < k$ for each $q, q' \in Q$ and we fix an enumeration $\rho_1^{q,q'}, \rho_2^{q,q'}, \dots, \rho_{R(q,q')}^{q,q'}$ of the returns from $(q, s:\mathrm{pop}_1(w))$ to (q', s). Similarly, let $\hat\rho_1^{q,q'}, \hat\rho_2^{q,q'}, \dots, \hat\rho_{R(q,q')}^{q,q'}$ be an enumeration of the returns from $(q, s':\mathrm{pop}_1(w'))$ to (q', s').

For each return from $(q_1, s:w)$ to (q_2, s) we construct a return from $(q_1, s':w')$ to (q_2, s') as follows. Let ρ be such a return and let $0 = i_0 \leq j_0 < i_1 \leq j_1 < \cdots < i_{m-1} \leq j_{m-1} < i_m \leq j_m$ be the unique sequence according to Lemma 4.12. Now, $\rho_k \coloneqq \rho\restriction_{[i_k, j_k - 1]}$ is prefixed by $s:w$ and ends in topmost word w. Hence, there is a run $\hat\rho_k \coloneqq \rho_k[s:w/s':w']$ ending in topmost word w'. Furthermore, $\pi_k \coloneqq \rho\restriction_{[j_k - 1, i_{k+1}]}$ is a pop_1 transition followed by a return σ_k. Since σ_k starts with topmost word $\mathrm{pop}_1(w)$, it is equivalent to some $\rho_l^{q,q'}$ in the sense that it performs the same transitions as $\rho_l^{q,q'}$. Now let $\hat\sigma_k$ be the return that copies the transitions of $\hat\rho_l^{q,q'}$ and starts in $(q, \mathrm{pop}_1(\hat\rho_k))$. Since $\mathrm{top}_1(w) = \mathrm{top}_1(w')$ we can define a run $\hat\pi_k$ that starts in the final configuration of $\hat\rho_k$ performs the same first transition as π_k and then agrees with $\hat\sigma_k$.

It is straightforward to prove that $\hat\rho \coloneqq \hat\rho_0 \circ \hat\pi_0 \circ \hat\rho_1 \circ \cdots \circ \hat\rho_{m-1} \circ \hat\pi_{m-1} \circ \hat\rho_m$ is a return from $(q_1, s':w')$ to (q_2, s'). Furthermore, this construction transforms distinct returns from $(q_1, s:w)$ to (q_2, s) into distinct returns from $(q_1, s':w')$ to (q_2, s'). Hence, there are at least as many returns from $(q_1, s':w')$ to (q_2, s') as from $(q_1, s:w)$ to (q_2, s).

Reversing the roles of $s:w$ and $s':w'$ we obtain the reverse result and we conclude that $\#\mathrm{Ret}^k(w)(q_1, q_2) = \#\mathrm{Ret}^k(w')(q_1, q_2)$. □

Proposition 4.14. *There is an algorithm that, on input some 2-PS $\mathcal{S}$ and a natural number k, computes a function $\mathrm{LL}_k^{\mathcal{S}} : \mathbb{N} \to \mathbb{N}$ with the following properties.*

(1) *For all stacks s, all $q_1, q_2 \in Q$ and for $i := \#\mathrm{Loop}^k(s)(q_1, q_2)$, the length-lexicographically shortest loops $\lambda_1, \dots, \lambda_i$ from (q_1, s) to (q_2, s) satisfy* $\mathrm{len}(\lambda_j) \leq \mathrm{LL}_k^{\mathcal{S}}(|\mathrm{top}_2(s)|)$ *for all* $1 \leq j \leq i$.
(2) *If there is a loop λ from (q_1, s) to (q_2, s) with* $\mathrm{len}(\lambda) > \mathrm{LL}_k^{\mathcal{S}}(|\mathrm{top}_2(s)|)$*, then there are k loops from (q_1, s) to $(q_2, \mathrm{pop}_2(s))$ of length at most* $\mathrm{LL}_k^{\mathcal{S}}(|\mathrm{top}_2(s)|)$.

Analogously, there are functions $\mathrm{RL}_k^{\mathcal{S}} : \mathbb{N} \to \mathbb{N}$ *and* $\mathrm{HLL}_k^{\mathcal{S}} : \mathbb{N} \to \mathbb{N}$ *(computable from $\mathcal{N}$) that satisfy the same assertions for the set of returns and high loops, respectively.*

sketch. Again, we only sketch the proof for the case of returns.

The previous proof showed that for stacks $s : w$ and $s' : w'$ with $\mathrm{top}_1(w) = \mathrm{top}_1(w')$ and $\#\mathrm{Ret}^k(w) = \#\mathrm{Ret}^k(w')$ the returns of $s : w$ and of $s' : w'$ are closely connected via the replacement $[s : w / s' : w']$. A return from $s : w$ to s decomposes into parts prefixed by $s : w$ and parts that are returns of stacks with topmost word $\mathrm{pop}_1(w)$. It is a straightforward observation that the k shortest returns from $s : w$ to s only contain returns from stacks with topmost word $\mathrm{pop}_1(w)$ among the k shortest of such returns. Furthermore, the decomposition of returns of $s : w$ and $s' : w'$ agree on their $s : w$-prefixed and $s' : w'$-prefixed parts in the sense that they perform the same transitions. Now, let $R(q_1, q_2) := \#\mathrm{Ret}^k(w)(q_1, q_2)$ and let $\rho_1, \rho_2, \dots, \rho_{R(q_1,q_2)}$ be the $R(q_1, q_2)$ shortest returns from $(q_1, s : w)$ to (q_2, s). Let $m(\rho_i)$ denote the number of positions in ρ_i that are $s : w$ prefixed and let $n(\rho_i)$ denote the number of returns from stacks with topmost word $\mathrm{pop}_1(w)$ occurring in ρ_i. Let m be the maximum over all $m(\rho_i)$ and n the maximum over all $n(\rho_i)$. Then there are $\#\mathrm{Ret}^k(w')(q_1, q_2)$ many returns from $(q_1, s' : w')$ to (q_2, s') that have at most m positions that are $s' : w'$ prefixed and that contain at most n many returns of stacks with topmost word $\mathrm{pop}_1(w')$. Thus, if we have already defined $\mathrm{RL}_k^{\mathcal{S}}(|\mathrm{pop}_1(w')|)$, then the shortest $\#\mathrm{Ret}^k(w')(q_1, q_2)$ many returns from $(q_1, s' : w')$ to (q_2, s') have length at most $m + n \cdot \mathrm{RL}_k^{\mathcal{S}}(|\mathrm{pop}_1(w')|)$.

Due to the previous lemma, we can compute a maximal finite sequence of words $w_1, w_2, \dots, w_k$ such that for each pair w_i, w_j, $\mathrm{top}_1(w_i) \neq \mathrm{top}_1(w_j)$ or $\mathrm{RL}_k^{\mathcal{S}}(w_i) \neq \mathrm{RL}_k^{\mathcal{S}}(w_j)$.

Repeating this construction for each of these words, we obtain numbers $m_{\max}$ and $n_{\max}$ as the maximum over the m and n obtained in each step such that the shortest $\#\mathrm{Ret}^k(w')(q_1, q_2)$ many returns from $(q_1, s' : w')$ to (q_2, s') have length at most $m_{\max} + n_{\max} \cdot \mathrm{RL}_k^{\mathcal{S}}(|\mathrm{pop}_1(w')|)$ for all stacks $s' : w'$. Thus, setting $\mathrm{RL}_k^{\mathcal{S}}(0) := 0$ and $\mathrm{RL}_k^{\mathcal{S}}(n+1) := m_{\max} + n_{\max} \cdot \mathrm{RL}_k^{\mathcal{S}}(n)$ defines $\mathrm{RL}_k^{\mathcal{S}}$ appropriately. □

Remark 4.15. Note that we do not know any bound on $m_{\max}$ and $n_{\max}$ in terms of $|\mathcal{S}|$.

We conclude this section with two corollaries that allow us to shrink runs between certain configurations. These corollaries are important in the proof that 2-NPT have the dynamic-small-witness property.

Corollary 4.16. *Let $\mathcal{S}$ be some level 2 collapsible pushdown system. Furthermore, let (q, s) be some configuration and $\rho_1, \dots, \rho_n$ be pairwise distinct runs from the initial configuration to (q, s). There is a run $\hat{\rho}_1$ from the initial configuration to (q, s) such that the following holds.*

(1) $\hat{\rho}_1 \neq \rho_i$ *for* $2 \leq i \leq n$ *and*
(2) $\mathrm{len}(\hat{\rho}_1) \leq 2 \cdot |s| \cdot \mathrm{hgt}(s)(1 + \mathrm{LL}_n^{\mathcal{S}}(\mathrm{hgt}(s)))$.

Proof. If $\mathrm{len}(\rho_1) \leq 2 \cdot |s| \cdot \mathrm{hgt}(s)(1 + \mathrm{LL}_n^{\mathcal{S}}(\mathrm{hgt}(s)))$, set $\hat{\rho}_1 := \rho_1$ and we are done. Otherwise, Remark 4.2 and Lemma 4.4 implies that ρ_1 decomposes as

$$\rho_1 = \lambda_0 \circ \mathrm{op}_1 \circ \lambda_1 \circ \dots \circ \lambda_{m-1} \circ \mathrm{op}_m \circ \lambda_m$$

where every λ_i is a loop, every op_i is a run of length 1, and $m \le 2 \cdot |s| \cdot \mathrm{hgt}(s)$. Proposition 4.14 implies the following: If $\mathrm{len}(\lambda_i) > \mathrm{LL}_n^{\mathcal{S}}(\mathrm{hgt}(s)))$, then there are n loops from $\lambda_i(0)$ to $\lambda_i(\mathrm{len}(\lambda_i))$ of length at most $\mathrm{LL}_n^{\mathcal{S}}(\mathrm{hgt}(s)))$. At least one of these can be plugged into the position of λ_i such that the resulting run does not coincide with any of the $\rho_2, \rho_3, \dots, \rho_n$. In other words, there is some loop λ_i' of length at most $\mathrm{LL}_n^{\mathcal{S}}(\mathrm{hgt}(s)))$ such that

$$\hat{\rho}_1 := \lambda_0 \circ \mathrm{op}_1 \circ \lambda_1 \circ \dots \circ \mathrm{op}_i \circ \lambda_i' \circ \mathrm{op}_{i+1} \circ \lambda_{i+1} \circ \dots \circ \lambda_{m-1} \circ \mathrm{op}_m \circ \lambda_m$$

is a run to (q, s) distinct from $\rho_2, \rho_3 \dots, \rho_n$ and shorter than ρ_1. Iterated replacement of large loops results in a run ρ_1' with the desired properties. □

We state a second corollary that is quite similar to the previous one but deals with runs of a different form. These runs become important in Section 6.

Corollary 4.17. *Let $\hat{\rho}_1, \hat{\rho}_2, \dots \hat{\rho}_n$ be runs from the initial configuration to some configuration (q, s). Furthermore, let w be some word and $\rho_1, \rho_2, \dots \rho_n$ be runs from (q, s) to $(q', s : w)$ that do not visit proper substacks of s. If $\hat{\rho}_1 \circ \rho_1, \hat{\rho}_2 \circ \rho_2, \dots, \hat{\rho}_n \circ \rho_n$ are pairwise distinct, then there is a run ρ_1' from (q, s) to $(q', s : w)$ that satisfies the following.*

1. *ρ_1' does not visit a proper substack of s,*
2. *$\mathrm{len}(\rho_1') \le 2 \cdot \mathrm{hgt}(s : w) \cdot (1 + \mathrm{LL}_n^{\mathcal{S}}(\mathrm{hgt}(s : w)))$, and*
3. *$\hat{\rho}_1 \circ \rho_1'$ is distinct from each $\hat{\rho}_i \circ \rho_i$ for $2 \le i \le n$.*

Proof. It is straightforward to see that ρ_1 decomposes as

$$\rho_1 = \lambda_0 \circ \mathrm{op}_1 \circ \lambda_1 \circ \dots \circ \lambda_{m-1} \circ \mathrm{op}_m \circ \lambda_m$$

where every λ_i is a loop and every op_i is a run of length 1 such that $m \le 2 \cdot \mathrm{hgt}(s : w)$. Now apply the construction from the previous proof again. □

5. Relevant Ancestors

This section aims at identifying those ancestors of a run ρ in a 2-NPT $\mathfrak{N}$ that characterise its FO_k-type. We show that only finitely many ancestors of a certain kind already fix the FO_k-type of ρ. We call these finitely many ancestors the *relevant 2^k-ancestors* of ρ.

5.1. Definition and Basic Observations.

Before we formally define relevant ancestors, we introduce some sloppy notation concerning runs. We apply functions defined on stacks to configurations. For example if $c = (q, s)$ we write $|c|$ for $|s|$ and $\mathrm{pop}_2(c)$ for $\mathrm{pop}_2(s)$. We further abuse this notation by application of functions defined on stacks to runs, meaning that we apply the function to the last stack occurring in a run. For instance, we write $\mathrm{top}_2(\rho)$ for $\mathrm{top}_2(s)$ and $|\rho|$ for $|s|$ if $\rho(\mathrm{len}(\rho)) = (q, s)$. In the same sense one has to understand equations like $\rho(i) = \mathrm{pop}_1(s)$. This equation says that $\rho(i) = (q, \mathrm{pop}_1(s))$ for some $q \in Q$. Keep in mind that $|\rho|$ denotes the width of the last stack of ρ and not its length $\mathrm{len}(\rho)$. Also recall that we write $\rho \le \rho'$ if the run ρ is an initial segment of the run ρ'.

Definition 5.1. Let $\mathfrak{N}$ be some n-NPT. Define the relation $\curvearrowright \subseteq \mathfrak{N} \times \mathfrak{N}$ by

$$\rho \curvearrowright \rho' \text{ if } \rho < \rho', |\rho| = |\rho'| - 1, \text{ and } |\pi| > |\rho| \text{ for all } \rho < \pi < \rho'.$$

We define the set of *relevant l-ancestors of ρ* by induction on l. The set of relevant 0-ancestors of ρ is $\mathrm{RA}_0(\rho) := \{\rho\}$. Set

$$\mathrm{RA}_{l+1}(\rho) := \mathrm{RA}_l(\rho) \cup \left\{\pi \in \mathfrak{N} : \exists \pi' \in \mathrm{RA}_l(\rho)\ \pi \vdash \pi' \text{ or } \pi \rightsquigarrow \pi' \text{ or } \pi \curvearrowright \pi'\right\}.$$

For $\bar{\rho} = (\rho_1, \rho_2, \dots, \rho_n)$, we write $\mathrm{RA}_l(\bar{\rho}) := \bigcup_{i=1}^{n} \mathrm{RA}_l(\rho_i)$.

Remark 5.2. For 1-NPT, the relation $\curvearrowright$ can be characterised as follows: for runs ρ, ρ', we have $\rho \curvearrowright \rho'$ if and only if $\rho' = \rho \circ \pi$ for a run π starting at some w_ρ and ending in $w_\rho a$, the first operation of π is a push_a and π visits w_ρ only in its initial configuration.

For 2-NPT, there is a similar characterisation: we have $\rho \curvearrowright \rho'$ if and only if $\rho' = \rho \circ \pi$ for some run π starting at some stack s_ρ and ending in some stack $s_\rho : w$, the first operation of π is a clone and π visits s_ρ only in its initial configuration.

The motivation for the definition is the following. If there are elements $\rho, \rho' \in \mathfrak{N}$ such that $\rho' \leq \rho$ and there is a path in $\mathfrak{N}$ of length at most l that witnesses that ρ' is an ancestor of ρ, then we want that $\rho' \in \mathrm{RA}_l(\rho)$. The relation $\curvearrowright$ is tailored towards this idea. Assume that there are runs $\rho_1 < \rho_2 \vdash^\delta \rho_3$ with δ some pop_n operation such that $\rho_2 \vdash^\delta \rho_3 \backsim \rho_1$. This path of length 2 witnesses that ρ_1 is a predecessor of ρ_2. By definition, one sees immediately that $\rho_1 \curvearrowright \rho_2$ whence $\rho_1 \in \mathrm{RA}_1(\rho_2)$. In this sense, $\curvearrowright$ relates the ancestor ρ_1 with ρ_2 if ρ_1 may be reachable from ρ_2 via a short path passing a descendant of ρ_2.

In the following, it may be helpful to think of a relevant l-ancestor ρ' of a run ρ as an ancestor of ρ that may have a path of length up to l witnessing that ρ' is an ancestor of ρ. We do not state this idea more precisely, but it may be helpful to keep this picture in mind.

From the definitions, we obtain immediately the following lemmas.

Lemma 5.3. *For each run ρ' there is at most one run ρ such that $\rho \curvearrowright \rho'$. ρ is the maximal ancestor of ρ' satisfying $\rho = \mathrm{pop}_n(\rho)$.*

Lemma 5.4. *Let ρ and ρ' be runs such that $\rho \rightsquigarrow \rho'$. Let $\hat{\rho}$ be the predecessor of ρ', i.e., $\hat{\rho}$ is the unique element such that $\hat{\rho} \vdash \rho'$. Then $\rho \curvearrowright \hat{\rho}$.*

Lemma 5.5. *If $\rho, \rho' \in \mathfrak{N}$ such that $\rho \vdash \rho'$ or $\rho \rightsquigarrow \rho'$, then $\rho \in \mathrm{RA}_1(\rho')$.*

Lemma 5.6. *For all $l \in \mathbb{N}$ and $\rho \in \mathfrak{N}$, $|\mathrm{RA}_l(\rho)| \leq 4^l$ and $\mathrm{RA}_l(\rho)$ is linearly ordered by $\leq$.*

We now investigate the structure of relevant ancestors and the possible intersections of relevant ancestors of different runs. First, we characterise the minimal element of $\mathrm{RA}_l(\rho)$.

Lemma 5.7. *Let $\mathfrak{N}$ be a n-NPT. Let $\rho_l \in \mathrm{RA}_l(\rho)$ be minimal with respect to $\leq$. Then*

Either $|\rho_l| = 1$ and $|\rho| \leq l$

or $\rho_l = \mathrm{pop}_n^l(\rho)$ and $|\rho_l| < |\rho'|$ for all $\rho' \in \mathrm{RA}_l(\rho) \setminus \{\rho_l\}$.

Remark 5.8. Recall that $|\rho| \leq l$ implies that $\mathrm{pop}_n^l(\rho)$ is undefined.

Proof. The proof is by induction on l. For $l = 0$, there is nothing to show because $\rho_0 = \rho = \mathrm{pop}_n^0(\rho)$. Now assume that the statement is true for some l.

(1) Consider the case $|\rho| \leq l + 1$. Then ρ_l satisfies $|\rho_l| = 1$. If ρ_l has no predecessor it is also the minimal element of $\mathrm{RA}_{l+1}(\rho)$ and we are done. Otherwise, there is a maximal ancestor $\hat{\rho} < \rho_l$ such that $|\hat{\rho}| = 1$. Either $\hat{\rho} \vdash \rho_l$ or $\hat{\rho} \rightsquigarrow \rho_l$ whence $\hat{\rho} \in \mathrm{RA}_{l+1}(\rho) \setminus \mathrm{RA}_l(\rho)$. Furthermore, there is no ancestor of $\hat{\rho}$ can be contained in $\mathrm{RA}_{l+1}(\rho)$. Heading for a contradiction, assume that there is some element $\tilde{\rho} < \hat{\rho}$

such that $\tilde{\rho} \in \mathrm{RA}_{l+1}(\rho)$. Then there is some $\tilde{\rho}' \in \mathrm{RA}_l(\rho)$ with $\tilde{\rho} \prec \hat{\rho} \prec \tilde{\rho}'$ such that $\tilde{\rho} \curvearrowright \tilde{\rho}'$ or $\tilde{\rho} \curvearrowright \tilde{\rho}'$. But this leads to the contradiction $1 \leq |\tilde{\rho}'| < |\hat{\rho}| = 1$. Thus, the minimal element of $\mathrm{RA}_{l+1}(\rho)$ is $\rho_{l+1} = \hat{\rho}$.

(2) Now assume that $|\rho| > l+1$. Let $\hat{\rho}$ be the maximal ancestor of ρ_l such that $|\hat{\rho}|+1 = |\rho_l|$. Then $\hat{\rho} \curvearrowright \rho_l$ or $\hat{\rho} \vdash \rho_1$, whence $\hat{\rho} \in \mathrm{RA}_{l+1}(\rho)$. We have to show that $\hat{\rho}$ is the minimal element of $\mathrm{RA}_{l+1}(\rho)$ and that there is no other element of width $|\hat{\rho}|$ in $\mathrm{RA}_{l+1}(\rho)$. For the second part, assume that there is some $\rho' \in \mathrm{RA}_{l+1}(\rho)$ with $|\rho'| = |\hat{\rho}|$. Then ρ' has to be connected via $\vdash$, $\curvearrowright$, or $\curvearrowright$ to some element $\rho'' \in \mathrm{RA}_l(\rho)$. By definition of these relations $|\rho''| \leq |\rho'| + 1$. By induction hypothesis, this implies $\rho'' = \rho_l$. But then it is immediately clear that $\rho' = \hat{\rho}$ by definition.

Similar to the previous case, the minimality of $\hat{\rho}$ in $\mathrm{RA}_{l+1}(\rho)$ is proved by contradiction. Assume that there is some $\rho' \prec \hat{\rho}$ such that $\rho' \in \mathrm{RA}_{l+1}(\rho)$. Then there is some $\hat{\rho} \prec \rho_l \leq \rho'' \in \mathrm{RA}_l(\rho)$ such that $\rho' \curvearrowright \rho''$ or $\rho' \curvearrowright \rho''$. By the definition of $\curvearrowright$ and $\curvearrowright$, we obtain $|\rho''| \leq |\hat{\rho}|$. But this contradicts $|\rho''| \geq |\rho_l| > |\hat{\rho}|$. Thus, we conclude that $\hat{\rho}$ is the minimal element of $\mathrm{RA}_{l+1}(\rho)$, i.e., $\hat{\rho} = \rho_{l+1}$.

□

The previous lemma shows that the width of stacks among the relevant ancestors cannot decrease too much. Furthermore, the width cannot grow too much.

Corollary 5.9. *Let $\pi, \rho \in \mathfrak{N}$ such that $\pi \in \mathrm{RA}_l(\rho)$. Then $\big||\rho| - |\pi|\big| \leq l$.*

Proof. From the previous lemma, we know that the minimal width of the last stack of an element in $\mathrm{RA}_l(\rho)$ is $|\rho| - l$. We prove by induction that the maximal width is $|\rho| + l$. The case $l = 0$ is trivially true. Assume that $|\pi| \leq |\rho| + l - 1$ for all $\pi \in \mathrm{RA}_{l-1}(\rho)$. Let $\hat{\pi} \in \mathrm{RA}_l(\rho) \setminus \mathrm{RA}_{l-1}(\rho)$. Then there is an $\pi \in \mathrm{RA}_{l-1}(\rho)$ such that $\hat{\pi} \vdash \pi$, $\hat{\pi} \curvearrowright \pi$, $\hat{\pi} \curvearrowright \pi$. For the last two cases the width of $\hat{\pi}$ is smaller than the width of π whence $|\hat{\pi}| \leq |\rho| + l - 1$. For the first case, recall that all stack operations of an n-PS alter the width of the stack by at most 1. Thus, $|\hat{\pi}| \leq |\pi| + 1 \leq |\rho| + l$. □

Next we show a kind of triangle inequality for the relevant ancestor relation. If ρ_2 is a relevant ancestor of ρ_1 then all relevant ancestors of ρ_1 that are ancestors of ρ_2 are relevant ancestors of ρ_2.

Lemma 5.10. *Let $\rho_1, \rho_2 \in \mathfrak{N}$ and let $l_1, l_2 \in \mathbb{N}$. If $\rho_1 \in \mathrm{RA}_{l_1}(\rho_2)$, then*

$$\mathrm{RA}_{l_2}(\rho_1) \subseteq \mathrm{RA}_{l_1+l_2}(\rho_2) \text{ and}$$
$$\mathrm{RA}_{l_2}(\rho_2) \cap \{\pi : \pi \leq \rho_1\} \subseteq \mathrm{RA}_{l_1+l_2}(\rho_1).$$

Proof. The first inclusion holds by induction on the definition of relevant ancestors.

For the second claim, we proceed by induction on l_2. For $l_2 = 0$ the claim holds because $\mathrm{RA}_0(\rho_2) = \{\rho_2\}$ and $\rho_1 \leq \rho_2$ imply that $\mathrm{RA}_0(\rho_2) \cap \{\pi : \pi \leq \rho_1\} \neq \emptyset$ if and only if $\rho_1 = \rho_2$ whence $\{\rho_2\} \in \mathrm{RA}_0(\rho_1)$. For the induction step assume that

$$\mathrm{RA}_{l_2-1}(\rho_2) \cap \{\pi : \pi \leq \rho_1\} \subseteq \mathrm{RA}_{l_1+l_2-1}(\rho_1).$$

Furthermore, assume that $\pi \in \mathrm{RA}_{l_2}(\rho_2) \cap \{\pi : \pi \leq \rho_1\}$. We show that $\pi \in \mathrm{RA}_{l_1+l_2}(\rho_1)$. By definition there is some $\pi \prec \hat{\pi}$ such that $\hat{\pi} \in \mathrm{RA}_{l_2-1}(\rho_2)$ and $\pi \in \mathrm{RA}_1(\hat{\pi})$. We distinguish the following cases.

- Assume that $\hat{\pi} \leq \rho_1$. By hypothesis, $\hat{\pi} \in \mathrm{RA}_{l_1+l_2-1}(\rho_1)$ whence $\pi \in \mathrm{RA}_{l_1+l_2}(\rho_1)$.

- Assume that $\pi < \rho_1 < \hat{\pi} < \rho_2$. This implies that $\pi \curvearrowright \hat{\pi}$ or $\pi \curvearrowright \hat{\pi}$ whence $|\pi| = |\hat{\pi}| - j < |\rho_1|$ for some $j \in \{0,1\}$. From Corollary 5.9, we know that
$$\big||\hat{\pi}| - |\rho_2|\big| \leq l_2 - 1 \text{ and} \big||\rho_1| - |\rho_2|\big| \leq l_1.$$
This implies that $|\rho_1| - |\pi| \leq l_1 + l_2$. By definition of $\curvearrowright$ and $\curvearrowright$, there cannot be any element π' with $\pi < \pi' < \hat{\pi}$ and $|\pi'| = |\pi|$. Thus, π is the maximal predecessor of ρ_1 with $\pi = \mathrm{pop}_2^{|\rho_1|-|\pi|}(\rho_1)$. Application of Lemma 5.7 shows that π is the minimal element of $\mathrm{RA}_{|\rho_1|-|\pi|}(\rho_1)$. Hence, $\pi \in \mathrm{RA}_{|\rho_1|-|\pi|}(\rho_1) \subseteq \mathrm{RA}_{l_1+l_2}(\rho_1)$.

□

Corollary 5.11. *If* $\rho \in \mathrm{RA}_l(\rho_1) \cap \mathrm{RA}_l(\rho_2)$ *then* $\mathrm{RA}_l(\rho_1) \cap \{\pi : \pi \leq \rho\} \subseteq \mathrm{RA}_{3l}(\rho_2)$.

Proof. By the previous lemma, $\rho \in \mathrm{RA}_l(\rho_1)$ implies $\mathrm{RA}_l(\rho_1) \cap \{\pi : \pi \leq \rho\} \subseteq \mathrm{RA}_{2l}(\rho)$. Using the lemma again, $\rho \in \mathrm{RA}_l(\rho_2)$ implies $\mathrm{RA}_{2l}(\rho) \subseteq \mathrm{RA}_{3l}(\rho_2)$. □

The previous corollary shows that if the relevant l-ancestors of two elements ρ_1 and ρ_2 intersect at some point ρ, then all relevant l-ancestors of ρ_1 that are ancestors of ρ are contained in the relevant $3l$-ancestors of ρ_2. Later, we use the contraposition of this result in order to prove that relevant ancestors of certain runs are disjoint sets.

The following proposition describes how $\mathrm{RA}_l(\rho)$ embeds into the full 2-NPT $\mathfrak{N}$. Successive relevant ancestors of some run ρ are either connected by a single edge or by a $\curvearrowright$-edge. We will see that this proposition allows us, given an arbitrary run ρ, to explicitly construct an relevant ancestor set isomorphic to $\mathrm{RA}_l(\rho)$ that consists of small runs.

Proposition 5.12. *Let* $\rho_1 < \rho_2 < \rho$ *such that* $\rho_1, \rho_2 \in \mathrm{RA}_l(\rho)$. *If* $\pi \notin \mathrm{RA}_l(\rho)$ *for all* $\rho_1 < \pi < \rho_2$, *then either* $\rho_1 \vdash \rho_2$ *or* $\rho_1 \curvearrowright \rho_2$.

Proof. Assume that $\rho_1 \not\vdash \rho_2$. Consider the set $M := \{\pi \in \mathrm{RA}_l(\rho) : \rho_1 \curvearrowright \pi\}$. M is nonempty because there is some $\pi \in \mathrm{RA}_{l-1}(\rho)$ such that either $\rho_1 \curvearrowright \pi$ (whence $\pi \in M$) or $\rho_1 \curvearrowright \pi$ (whence the predecessor $\hat{\pi}$ of π satisfies $\hat{\pi} \in M$). Let $\hat{\rho} \in M$ be minimal. It suffices to show that $\hat{\rho} = \rho_2$. For this purpose, we show that $\pi \notin \mathrm{RA}_l(\rho)$ for all $\rho_1 < \pi < \hat{\rho}$. Since $\hat{\rho} \in \mathrm{RA}_l(\rho)$, this implies that $\hat{\rho} = \rho_2$. We start with two general observations.

(1) For all $\rho_1 < \pi < \hat{\rho}$, $|\pi| \geq |\hat{\rho}|$ due to the definition of $\rho_1 \curvearrowright \hat{\rho}$. Furthermore, due to the minimality of $\hat{\rho}$ in M, for all $\rho_1 < \pi < \hat{\rho}$ with $\pi \in \mathrm{RA}_l(\rho)$, $|\pi| > |\hat{\rho}|$ (otherwise $\pi \in M$ which contradicts the minimality of $\hat{\rho}$).
(2) Note that there cannot exist $\rho_1 < \pi < \hat{\rho} < \hat{\pi}$ with $\pi \curvearrowright \hat{\pi}$ or $\pi \curvearrowright \hat{\pi}$ because $|\pi| \geq |\hat{\rho}|$.

Heading for a contradiction, assume that there is some $\rho_1 < \pi < \hat{\rho}$ such that $\pi \in \mathrm{RA}_l(\rho)$.

Due to observation 2, there is a chain $\pi_0 := \pi, \pi_1, \dots, \pi_{n-1}, \pi_n := \hat{\rho}$ such that for each $0 \leq i < n$ there is $* \in \{\vdash, \curvearrowright, \curvearrowright\}$ such that $\pi_i * \pi_{i+1}$ and $\pi_i \in \mathrm{RA}_{l-i}(\rho)$. By assumption, $n \neq 0$, whence $\hat{\rho} \in \mathrm{RA}_{l-1}(\rho)$. Due to observation 1, we have $|\rho_1| < |\hat{\rho}| < |\pi|$. Since each stack operation alters the width of the stack by at most 1, we conclude that the set
$$M' := \left\{\pi' : \rho_1 < \pi' < \hat{\rho}, |\hat{\rho}| = |\pi'|\right\}$$
is nonempty because on the path from ρ_1 to π there occurs at least one run with final stack of width $|\hat{\rho}|$. But the maximal element $\pi' \in M'$ satisfies $\rho_1 \curvearrowright \pi' \vdash \hat{\rho}$ or $\rho_1 \curvearrowright \pi' \curvearrowright \hat{\rho}$. Since $\hat{\rho} \in \mathrm{RA}_{l-1}(\rho)$, this would imply $\pi' \in M$ which contradicts the minimality of $\hat{\rho}$ in M. Thus, no $\rho_1 < \pi < \hat{\rho}$ with $\pi \in \mathrm{RA}_l(\rho)$ can exist.

We conclude that $\pi \notin \mathrm{RA}_l(\rho)$ for all $\rho_1 < \pi < \hat{\rho}$ and $\rho_1 \curvearrowright \hat{\rho} = \rho_2$. □

In the final part of this section, we consider relevant ancestors of two different runs ρ and ρ'. Since we aim at a construction of small runs $\hat{\rho}$ and $\hat{\rho}'$ such that the relevant ancestors of ρ and ρ' are isomorphic to the relevant ancestors of $\hat{\rho}$ and $\hat{\rho}'$, we need to know how sets of relevant ancestors touch each other. Every isomorphism from the relevant ancestors of ρ and ρ' to those of $\hat{\rho}$ and $\hat{\rho}'$ has to preserve edges between a relevant ancestor of ρ and another one of ρ'.

The positions where the relevant l-ancestors of ρ and $\hat{\rho}$ touch can be identified by looking at the intersection of their relevant $(l+1)$-ancestors. This is shown in the next Lemma. For A and B subsets of some n-NPT $\mathfrak{N}$ and ρ some run of $\mathfrak{N}$, we say A and B *touch after* ρ if there are runs $\rho \prec \rho_A, \rho \prec \rho_B$ such that $\rho_A \in A$, $\rho_B \in B$ and either $\rho_A = \rho_B$ or $\rho_A * \rho_B$ for some $* \in \{\vdash, \dashv, \rightharpoondown, \leftharpoondown, \curvearrowright, \curvearrowleft\}$. In this case we say A and B touch at (ρ_A, ρ_B). In the following, we reduce the question whether l-ancestors of two elements touch after some ρ to the question whether the $(l+1)$-ancestors of these elements intersect after ρ.

Lemma 5.13. *If ρ_1, ρ_2 are runs such that $\mathrm{RA}_{l_1}(\rho_1)$ and $\mathrm{RA}_{l_2}(\rho_2)$ touch after some ρ_0, then $\mathrm{RA}_{l_1+1}(\rho_1) \cap \mathrm{RA}_{l_2+1}(\rho_2) \cap \{\pi : \rho_0 \leq \pi\} \neq \emptyset$.*

Proof. Let ρ_0 be some run, $\rho_0 \prec \hat{\rho}_1 \in \mathrm{RA}_{l_1}(\rho_1)$, and $\rho_0 \prec \hat{\rho}_2 \in \mathrm{RA}_{l_2}(\rho_2)$ such that the pair $(\hat{\rho}_1, \hat{\rho}_2)$ is minimal and $\mathrm{RA}_{l_1}(\rho_1)$ and $\mathrm{RA}_{l_2}(\rho_2)$ touch at $(\hat{\rho}_1, \hat{\rho}_2)$. Then one of the following holds.

(1) $\hat{\rho}_1 = \hat{\rho}_2$: there is nothing to prove because $\hat{\rho}_1 \in \mathrm{RA}_{l_1}(\rho_1) \cap \mathrm{RA}_{l_2}(\rho_2) \cap \{\pi : \rho_0 \leq \pi\}$.
(2) $\hat{\rho}_1 \vdash \hat{\rho}_2$ or $\hat{\rho}_1 \rightharpoondown \hat{\rho}_2$ or $\hat{\rho}_1 \curvearrowright \hat{\rho}_2$: this implies that $\hat{\rho}_1 \in \mathrm{RA}_{l_2+1}(\rho_2) \cap \mathrm{RA}_{l_1}(\rho_1)$.
(3) $\hat{\rho}_1 \dashv \hat{\rho}_2$ or $\hat{\rho}_1 \leftharpoondown \hat{\rho}_2$ or $\hat{\rho}_1 \curvearrowleft \hat{\rho}_2$: this implies that that $\hat{\rho}_2 \in \mathrm{RA}_{l_1+1}(\rho_1) \cap \mathrm{RA}_{l_2}(\rho_2)$.

□

Corollary 5.14. *If ρ and ρ' are runs such that $\mathrm{RA}_{l_1}(\rho)$ and $\mathrm{RA}_{l_2}(\rho')$ touch after some run ρ_0 then there exists some $\rho_0 \prec \rho_1 \in \mathrm{RA}_{l_1+1}(\rho) \cap \mathrm{RA}_{l_2+1}(\rho')$ such that*

$$\mathrm{RA}_{l_1+1}(\rho) \cap \{x : x \leq \rho_1\} \subseteq \mathrm{RA}_{l_2+2l_1+3}(\rho').$$

Proof. Use the previous lemma and Lemma 5.10. □

5.2. **A Family of Equivalence Relations on Words and Stacks.** Next, we define a family of equivalences on words that is useful for constructing runs with similar relevant ancestors. The basic idea is to classify words according to the FO_k-type of the word model associated to the word w expanded by information about certain runs between prefixes of w. This additional information describes

(1) the number of possible loops and returns with certain initial and final state of each prefix $v \leq w$, and
(2) the number of runs from (q, w) to (q', v) for each prefix $v \leq w$ and all pairs q, q' of states.

It turns out that this equivalence has the following property: if w and w' are equivalent and ρ is a run starting in (q, w) and ending in $(q', w : v)$, then there is a run from (q, w') to $(q, w' : v')$ such that the loops and returns of v and v' agree. This is important because runs of this kind connect consecutive elements of relevant ancestor sets (cf. Proposition 5.12).

In order to copy relevant ancestors, we want to apply this kind of transfer property iteratively. For instance, we want to take a run from (q_1, w_1) via $(q_2, w_1 : w_2)$ to $(q_3, w_1 : w_2 : w_3)$ and translate it into some run from (q_1, w_1') via $(q_2, w_1' : w_2')$ to $(q_3, w_1' : w_2' : w_3')$

such that the loops and returns of w_3 and w_3' agree. Analogously, we want to take a run creating n new words and transfer it to a new run starting in another word and creating n words such that the last words agree on their loops and returns. If we can do this, then we can transfer the whole set of relevant ancestors from some run to another one. This allows us to construct isomorphic relevant ancestors that consist only of short runs.

The family of equivalence relations that we define have the following transfer property. Words that are equivalent with respect to the n-th relation allow a transfer of runs creating n new words. The idea of the definition is as follows. Assume that we have already defined the $(i-1)$-st equivalence relation. We take the word model of some word w and annotate each prefix of the word by its equivalence class with respect to the $(i-1)$-st relation. Then we define two words to be equivalent with respect to the i-th relation if the FO_k-types of their enriched word models agree.

These equivalence relations and the transfer properties that they induce are an important tool in the next section. There we apply them to an arbitrary set of relevant ancestors in order to obtain isomorphic copies of the substructure induced by these ancestors.

For the rest of this section, we fix some 2-PS $\mathcal{N}$. For w some word, we use w_{-n} as an abbreviation for $\mathrm{pop}_1^n(w)$.

Definition 5.15. For each word $w \in \Sigma^*$, we define a family of expanded word models $\mathfrak{Lin}_n^{k;z}(w)$ by induction on n. Note that for $n = 0$ the structure will be independent of the parameter k but for greater n this parameter influences the expansion of the structure. Let $\mathfrak{Lin}_0^{k;z}(w)$ be the expanded word model

$$\mathfrak{Lin}_0^{k;z}(w) := (\{0,1,\dots,|w|-1\}, \mathrm{succ}, (P_\sigma)_{\sigma\in\Sigma}, (S^j_{q,q'})_{(q,q')\in Q^2, j\le z}, (R_j)_{j\in J}, (L_j)_{j\in J}, (H_j)_{j\in J})$$

such that for $0 \le i < |w|$ the following holds.

- succ and P_σ form the standard word model of w in reversed order, i.e., succ is the usual successor relation on the domain and $i \in P_\sigma$ if and only if $\mathrm{top}_1(w_{-i}) = \sigma$,
- $i \in S^j_{q,q'}$, if there are j pairwise distinct runs $\rho_1,\dots,\rho_j$ such that each run starts in (q,w) and ends in (q', w_{-i}).
- The predicates R_j encode the function $i \mapsto \#\mathrm{Ret}^z(w_{-i})$ (cf. Definition 4.5).
- The predicates L_j encode the function $i \mapsto \#\mathrm{Loop}^z(w_{-i})$.
- The predicates H_j encode the function $i \mapsto \#\mathrm{HLoop}^z(w_{-i})$.

Now, set $\mathrm{Type}_0^{k;z}(w) := \mathrm{FO}_k[\mathfrak{Lin}_0^{k;z}(w)]$, the quantifier rank k theory of $\mathfrak{Lin}_0^{k;z}(w)$.

Inductively, we define $\mathfrak{Lin}_{n+1}^{k;z}(w)$ to be the expansion of $\mathfrak{Lin}_n^{k;z}(w)$ by predicates describing $\mathrm{Type}_n^{k;z}(v)$ for each prefix $v \le w$. More formally, fix a maximal list $\theta_1,\theta_2,\dots,\theta_m$ of pairwise distinct FO_k-types that are realised by some $\mathfrak{Lin}_n^{k;z}(w)$. We define predicates $T_1,T_2,\dots,T_m$ such that $i \in T_j$ if $\mathrm{Type}_n^{k;z}(w_{-i}) = \theta_j$ for all $0 \le i \le n$. Now, let $\mathfrak{Lin}_{n+1}^{k;z}(w)$ be the expansion of $\mathfrak{Lin}_n^{k;z}(w)$ by the predicates $T_1,T_2,\dots,T_m$. We conclude the inductive definition by setting $\mathrm{Type}_{n+1}^{k;z}(w) := \mathrm{FO}_k[\mathfrak{Lin}_{n+1}^{k;z}(w)]$.

Remark 5.16. Each element of $\mathfrak{Lin}_n^{k;z}(w)$ corresponds to a prefix of w. In this sense, we write $v \in S^j_{q,q'}$ for some prefix $v \le w$ if $v = w_{-i}$ and $\mathfrak{Lin}_n^{k;z}(w) \models i \in S^j_{q,q'}$.

It is an important observation that $\mathfrak{Lin}_n^{k;z}(w)$ is a finite successor structure with finitely many colours. Thus, for all $n,k,z \in \mathbb{N}$, $\mathrm{Type}_n^{k;z}$ has finite image.

For our application, k and z can be chosen to be some fixed large numbers, depending on the quantifier rank of the formula we are interested in. Furthermore, it will turn out that the conditions on k and z coincide whence we will assume that $k = z$. This is due to the fact that both parameters are counting thresholds in some sense: z is the threshold for counting the existence of loops and returns, while k can be seen as the threshold for distinguishing different prefixes of w which have the same atomic type. Thus, we identify k and z in the following definition of the equivalence relation induced by $\mathrm{Type}_n^{k;z}$.

Definition 5.17. For words $w, w' \in \Sigma^*$, we write $w \equiv_n^z w'$ if $\mathrm{Type}_n^{z;z}(w) = \mathrm{Type}_n^{z;z}(w')$.

As a first step, we want to show that $\equiv_n^z$ is a right congruence. We prepare the proof of this fact in the following lemma.

Lemma 5.18. *Let $n \in \mathbb{N}$, $z \geq 2$ and $\mathcal{N}$ be some 2-PS. Let w be some word and $\sigma \in \Sigma$ some letter. For each $0 \leq i < |w|$, the atomic types of i and of 0 in $\mathfrak{Lin}_n^{z;z}(w)$ determine the atomic type of $i+1$ in $\mathfrak{Lin}_n^{z;z}(w\sigma)$.*

Proof. Recall that $i \in \mathfrak{Lin}_n^{z;z}(w)$ represents w_{-i} and $i+1 \in \mathfrak{Lin}_n^{z;z}(w\sigma)$ represents $w\sigma_{-(i+1)}$. Since $w_{-i} = w\sigma_{-(i+1)}$, it follows directly that the two elements agree on $(P_\sigma)_{\sigma\in\Sigma}$, $(R_j)_{j\in J}$, $(L_j)_{j\in J}$, and $(H_j)_{j\in J}$ and that $w_{-i} \equiv_{n-1}^z w\sigma_{-(i+1)}$ (recall that the elements of $\mathfrak{Lin}_n^{z;z}(w)$ are coloured by $\equiv_{n-1}^z$-types).

We claim that the function $\#\mathrm{Ret}^z(w)$ and the set

$$\{(j, q, q') \in \mathbb{N} \times Q \times Q : j \leq z, \mathfrak{Lin}_n^{z;z}(w) \models i \in S_{q,q'}^j\}$$

determine whether $\mathfrak{Lin}_n^{z;z}(w\sigma) \models (i+1) \in S_{q,q'}^j$. Recall that the predicates $S_{q,q'}^j$ in $\mathfrak{Lin}_n^{z;z}(w)$ encode at each position j the number of runs from (q, w) to (q', w_{-j}). We now want to determine the number of runs from $(q, w\sigma)$ to $(q', w\sigma_{-(i+1)}) = (q', w_{-i})$.

It is clear that such a run starts with a high loop from $(q, w\sigma)$ to some $(\hat{q}, w\sigma)$. Then it performs some transition of the form $(\hat{q}, \sigma, \hat{q}', \mathrm{pop}_1)$ and then it continues with a run from $(\hat{q}', w)$ to (q', w_{-i}).

In order to determine whether $\mathfrak{Lin}_n^{z;z}(w\sigma) \models (i+1) \in S_{q,q'}^j$, we have to count whether j runs of this form exist. To this end, we define the numbers

$$\begin{aligned} k_{(\hat{q},\hat{q}')} &:= \#\mathrm{HLoop}^z(w\sigma)(q, \hat{q}), \\ j_{(\hat{q},\hat{q}')} &:= |\{(\hat{q}, \sigma, \hat{q}', \mathrm{pop}_1) \in \Delta\}|, \text{ and} \\ i_{(\hat{q},\hat{q}')} &:= \max\{k : w_{-i} \in S_{(\hat{q}',q')}^k\} \end{aligned}$$

for each pair $\bar{q} = (\hat{q}, \hat{q}') \in Q^2$. It follows directly that there are $\sum_{\bar{q}\in Q^2} i_{\bar{q}} j_{\bar{q}} k_{\bar{q}}$ many such runs up to threshold z. Note that $j_{\bar{q}}$ only depends on the pushdown system. Due to Corollary 4.13, $\#\mathrm{HLoop}^z(w\sigma)$ is determined by σ and $\#\mathrm{Ret}^z(w)$. Thus, $k_{\bar{q}}$ is determined by the atomic type of 0 in $\mathfrak{Lin}_n^{z;z}(w)$. $i_{\bar{q}}$ only depends on the atomic type of i in $\mathfrak{Lin}_n^{z;z}(w)$. These observations complete the proof. □

Corollary 5.19. *Let $n, z \in \mathbb{N}$ such that $z \geq 2$. Let w_1 and w_2 be words such that $w_1 \equiv_n^z w_2$. Any strategy of Duplicator in the z round Ehrenfeucht-Fraïssé game on $\mathfrak{Lin}_n^{z;z}(w_1)$ and $\mathfrak{Lin}_n^{z;z}(w_2)$ translates directly into a strategy of Duplicator in the z round Ehrenfeucht-Fraïssé game on $\mathfrak{Lin}_n^{z;z}(w_1\sigma){\restriction}_{[1,|w_1|]}$ and $\mathfrak{Lin}_n^{z;z}(w_2\sigma){\restriction}_{[1,|w_2|]}$.*

Proof. It suffices to note that the existence of Duplicator's strategy implies that the atomic types of 0 in $\mathfrak{Lin}_n^{z;z}(w_1)$ and $\mathfrak{Lin}_n^{z;z}(w_2)$ agree. Hence, the previous lemma applies. Thus, if the atomic type of $i \in \mathfrak{Lin}_n^{z;z}(w_1)$ and $j \in \mathfrak{Lin}_n^{z;z}(w_2)$ agree, then the atomic types of

$i+1 \in \mathfrak{Lin}_n^{z;z}(w_1\sigma)$ and $j+1 \in \mathfrak{Lin}_n^{z;z}(w_2\sigma)$ agree. Hence, we can obviously translate Duplicator's strategy on $\mathfrak{Lin}_n^{z;z}(w_1)$ and $\mathfrak{Lin}_n^{z;z}(w_2)$ into a strategy on $\mathfrak{Lin}_n^{z;z}(w_1\sigma){\restriction}_{[1,|w_1|]}$ and $\mathfrak{Lin}_n^{z;z}(w_2\sigma){\restriction}_{[1,|w_2|]}$. □

The previous corollary is the main ingredient for the following lemma. It states that $\equiv_n^z$ is a right congruence.

Lemma 5.20. *For $z \geq 2$, $\equiv_n^z$ is a right congruence, i.e., if* $\mathrm{Type}_n^{z;z}(w_1) = \mathrm{Type}_n^{z;z}(w_2)$ *for some $z \geq 2$, then* $\mathrm{Type}_n^{z;z}(w_1 w) = \mathrm{Type}_n^{z;z}(w_2 w)$ *for all $w \in \Sigma^*$.*

Proof. It is sufficient to prove the claim for $w = \sigma \in \Sigma$. The lemma then follows by induction on $|w|$. First observe that

$$\begin{aligned}&\#\mathrm{Loop}^z(w_1\sigma) = \#\mathrm{Loop}^z(w_2\sigma),\\ &\#\mathrm{HLoop}^z(w_1\sigma) = \#\mathrm{HLoop}^z(w_2\sigma), \text{ and}\\ &\#\mathrm{Ret}^z(w_1\sigma) = \#\mathrm{Ret}^z(w_2\sigma),\end{aligned}$$

because these values are determined by the values of the corresponding functions at w_1 and w_2 (cf. Proposition 4.13). These functions agree on w_1 and w_2 because the first elements of $\mathfrak{Lin}_n^{z;z}(w_1)$ and $\mathfrak{Lin}_n^{z;z}(w_2)$ are $\mathrm{FO}_2 \subseteq \mathrm{FO}_z$ definable.

For $i \in \{1,2\}$, $\mathfrak{Lin}_n^{z;z}(w_i\sigma) \models 0 \in S^j_{(q,q')}$ if and only if there are j loops from $(q, w_i\sigma)$ to $(q', w_i\sigma)$ (at position 0 the runs counted by the $S^j_{(q,q')}$ coincide with loops). Since $\#\mathrm{Loop}^z(w_1\sigma) = \#\mathrm{Loop}^z(w_2\sigma)$, we conclude that the atomic types of the first elements of $\mathfrak{Lin}_0^{z;z}(w_1\sigma)$ and of $\mathfrak{Lin}_0^{z;z}(w_2\sigma)$ coincide.

Due to the previous corollary, we know that Duplicator has a strategy in the z round Ehrenfeucht-Fraïssé game on $\mathfrak{Lin}_n^{z;z}(w_1\sigma){\restriction}_{[1,|w_1|]}$ and $\mathfrak{Lin}_n^{z;z}(w_2\sigma){\restriction}_{[1,|w_2|]}$.

Standard composition arguments for Ehrenfeucht-Fraïssé games on word structures directly imply that $\mathfrak{Lin}_0^{z;z}(w_1\sigma) \equiv_z \mathfrak{Lin}_0^{z;z}(w_2\sigma)$. But this directly implies that the atomic types of $w_1\sigma$ in $\mathfrak{Lin}_1^{z;z}(w_1\sigma)$ and of $w_2\sigma$ in $\mathfrak{Lin}_1^{z;z}(w_2\sigma)$ coincide. If $n \geq 1$, we can apply the same standard argument and obtain that $\mathfrak{Lin}_1^{z;z}(w_1\sigma) \equiv_z \mathfrak{Lin}_1^{z;z}(w_2\sigma)$. By induction one concludes that $\mathfrak{Lin}_n^{z;z}(w_1\sigma) \equiv_z \mathfrak{Lin}_n^{z;z}(w_2\sigma)$. But this is the definition of $w_1\sigma \equiv_n^z w_2\sigma$. □

In terms of stack operations, the previous lemma can be seen as a compatibility result of $\equiv_n^z$ with the push_σ operation. Next, we lift the equivalences from words to 2-stacks in such a way that the new equivalence relations are compatible with all stack operations. We compare the stacks word-wise beginning with the topmost word, then the word below the topmost one, etc. up to some threshold m.

Definition 5.21. Let s, s' be stacks. We write $s \;{}_m{\equiv}_n^z\; s'$ if either

$$\begin{aligned}&|s| > m, |s'| > m, \text{ and } \mathrm{top}_2\left(\mathrm{pop}_2^i(s)\right) \equiv_n^z \mathrm{top}_2\left(\mathrm{pop}_2^i(s')\right) \text{ for all } 0 \leq i \leq m, \text{ or}\\ &l := |s| = |s'| \leq m, \text{ and } \mathrm{top}_2\left(\mathrm{pop}_2^i(s)\right) \equiv_n^z \mathrm{top}_2\left(\mathrm{pop}_2^i(s')\right) \text{ for all } 0 \leq i < l.\end{aligned}$$

Proposition 5.22. *Let $z \geq 2$ and let s_1, s_2 be stacks such that $s_1 \;{}_m{\equiv}_n^z\; s_2$. Then* $\mathrm{push}_\sigma(s_1) \;{}_m{\equiv}_n^z\; \mathrm{push}_\sigma(s_2)$, $\mathrm{pop}_1(s_1) \;{}_m{\equiv}_{n-1}^z\; \mathrm{pop}_1(s_2)$, $\mathrm{clone}_2(s_1) \;{}_{m+1}{\equiv}_n^z\; \mathrm{clone}_2(s_2)$, *and* $\mathrm{pop}_2(s_1) \;{}_{m-1}{\equiv}_n^z\; \mathrm{pop}_2(s_2)$.

Proof. Assume that $\mathrm{op} = \mathrm{pop}_1$. Quantifier rank z suffices to define the second element of a word structure. Hence, $w \equiv_n^z w'$ implies that $\mathrm{Type}_{n-1}^{z;z}(w_{-1}) = \mathrm{Type}_{n-1}^{z;z}(w'_{-1})$. But this implies $w_{-1} \equiv_{n-1}^z w'_{-1}$.

For $\mathrm{op} = \mathrm{push}_\sigma$ we use Lemma 5.20. For clone_2 and pop_2, the claim is trivial. □

The previous proposition shows that the equivalence relations on stacks are compatible with the stack operations. Recall that successive relevant ancestors of a given run ρ are runs $\rho_1 \prec \rho_2 \preceq \rho$ such that either $\rho_1 \vdash \rho_2$ or $\rho_1 \curvearrowright \rho_2$ (cf. Proposition 5.12). In the next section, we are concerned with the construction of a short run $\hat{\rho}$ such that its relevant ancestors are isomorphic to those of ρ. A necessary condition for a run $\hat{\rho}$ to be short is that it only passes small stacks. We construct $\hat{\rho}$ using the following construction. Let $\rho_0 \prec \rho_1 \prec \rho_2 \dots \prec \rho$ be the set of relevant ancestors of ρ. We then first define a run $\hat{\rho}_0$ that ends in some small stack that is equivalent to the last stack of ρ_0. Then, we iterate the following construction. If ρ_{i+1} extends ρ_i by a single transition, then we define $\hat{\rho}_{i+1}$ to be the extension of $\hat{\rho}_i$ by the same transition. Due to the previous proposition this preserves equivalence of the topmost stacks of ρ_i and $\hat{\rho}_i$. Otherwise, ρ_{i+1} extends ρ_i by some run that creates a new word w_{i+1} on top of the last stack of ρ_i. Then we want to construct a short run that creates a new word w'_{i+1} on top of the last stack of $\hat{\rho}_i$ such that w_{i+1} and w'_{i+1} are equivalent and w'_{i+1} is small. Then we define $\hat{\rho}_{i+1}$ to be $\hat{\rho}_i$ extended by this run.

Finally, this procedure defines a run $\hat{\rho}$ that corresponds to ρ in the sense that the relevant ancestors of the two runs are isomorphic but $\hat{\rho}$ is a short run.

In the following, we prepare this construction. We show that for any run ρ_0 there is a run $\hat{\rho}_0$ that ends in some small stack that is equivalent to the last stack of ρ_0. This is done in Lemma 5.25. Furthermore, we show that for runs ρ_i and $\hat{\rho}_i$ that end in equivalent stacks, any run that extends the last stack of ρ_i by some word w can be transferred into a run that extends $\hat{\rho}_i$ by some small word that is equivalent to w. This is shown in Proposition 5.27.

The proofs of Lemma 5.25 and Proposition 5.27 are based on the property that prefixes of equivalent stacks share the same number of loops and returns for each pair of initial and final states. Recall that our analysis of generalised milestones showed that the existence of loops with certain initial and final states has a crucial influence on the question whether runs between certain stacks exist. We first define functions that are used to define what a small stack is. Afterwards, we show that any run to some stack can be replaced by some run to a short equivalent stack.

Definition 5.23. Let $\mathcal{N} = (Q, \Sigma, \Delta, q_0)$ be a 2-PS. Set $\alpha(n,z) = |Q| \cdot |{}^{\Sigma^*}\!/\!_{\equiv^z_n}| + 1$, where $|{}^{\Sigma^*}\!/\!_{\equiv^z_n}|$ is the number of equivalence classes of $\equiv^z_n$. Furthermore, set $\mathrm{B}_{\mathrm{hgt}} := |{}^{\Sigma^*}\!/\!_{\equiv^2_0}| \cdot |Q|^2$ and $\beta(n) := |Q| \cdot (|\Sigma| + 1)^n$.

Remark 5.24. $\beta(n)$ is an upper bound for the number of pairs of states and words of length up to n. Note that $\alpha, \mathrm{B}_{\mathrm{hgt}}$ and β computably depend on $\mathcal{N}$.

Lemma 5.25. *Let $n \in \mathbb{N}$, $z \geq 2$, and let ρ be a run from the initial configuration to some configuration (q,s). There is a run ρ' starting in the initial configuration such that*

$$|\mathrm{top}_2(\rho)| - \alpha(n,z) \leq |\mathrm{top}_2(\rho')| \leq |\mathrm{top}_2(\rho)|,$$
$$\mathrm{hgt}(\rho') \leq |\mathrm{top}_2(\rho')| + \mathrm{B}_{\mathrm{hgt}},$$
$$|\rho'| \leq \beta(\mathrm{hgt}(\rho')) \text{ and}$$
$$\mathrm{top}_2(\rho) \equiv^z_n \mathrm{top}_2(\rho').$$

Furthermore, if $|\mathrm{top}_2(\rho)| > \alpha(n,z)$, then $|\mathrm{top}_2(\rho')| < |\mathrm{top}_2(\rho)|$.

We prove this lemma in three steps.

(1) For each run ρ with long topmost word, we generate a run ρ' with equivalent but smaller topmost word.

$$s = \begin{array}{cccc} & & f & e \\ b & d & e & e \\ a & a & a & a \\ \bot & \bot & \bot & \bot \end{array} \quad m_1 = \bot \quad m_2 = \begin{array}{ccc} b & d & \\ a & a & a \\ \bot & \bot & \bot \end{array} \quad m_3 = \begin{array}{ccc} b & d & e \\ a & a & a \\ \bot & \bot & \bot \end{array}$$

Figure 1: Illustration for the construction in the first step of the proof of Lemma 5.25.

$$s = m_2' = \begin{array}{ccccc} & d & f & & \\ b & d & d & d & \\ a & a & a & a & a \\ \bot & \bot & \bot & \bot & \bot \end{array} \quad m = m_4 = \begin{array}{ccc} & d & f \\ b & d & d \\ a & a & a \\ \bot & \bot & \bot \end{array}$$

$$m_3 = \begin{array}{ccc} & d & \\ b & d & d \\ a & a & a \\ \bot & \bot & \bot \end{array} \quad m_2 = \begin{array}{cc} b & \\ a & a \\ \bot & \bot \end{array} \quad m_4' = \begin{array}{cccc} & d & f & f \\ b & d & d & d \\ a & a & a & a \\ \bot & \bot & \bot & \bot \end{array} \quad m_3' = \begin{array}{cccc} & d & f & \\ b & d & d & d \\ a & a & a & a \\ \bot & \bot & \bot & \bot \end{array}$$

Figure 2: Illustration for the construction in the second step of the proof of Lemma 5.25.

(2) For each run ρ, we generate a run ρ' such that $\mathrm{top}_2(\rho) = \mathrm{top}_2(\rho')$ and the height of ρ' is bounded by $|\mathrm{top}_2(\rho)| + \mathrm{B}_{\mathrm{hgt}}$.
(3) For each run ρ, we generate a run ρ' such that $\mathrm{top}_2(\rho) = \mathrm{top}_2(\rho')$, the height of ρ' is bounded in terms of $\mathrm{hgt}(\rho)$ and the width of ρ' is bounded in terms of its height.

of step 1. Let ρ be some run with $|\mathrm{top}_2(\rho)| > \alpha(n,z)$. Let $(q,s) := \rho(\mathrm{len}(\rho))$ be the final configuration of ρ. For each $k \leq \alpha(n,z)$, there is a maximal milestone $m_k \in \mathrm{MS}(s)$ with $|\mathrm{top}_2(m_k)| = k$. Figure 1 illustrates this definition. Let $w_k := \mathrm{top}_2(m_k)$ and let $\rho_k \leq \rho$ be the largest initial segment of ρ that ends in m_k. Note that $m_k \trianglelefteq m_{k'} \trianglelefteq s$ for all $k \leq k' \leq \alpha(n,z)$ by the maximality of m_k and $m_{k'}$.

Then there are $i < j \leq \alpha(n,z)$ such that $\mathrm{top}_2(\rho_i) \equiv^z_n \mathrm{top}_2(\rho_j)$ and the final states of ρ_i and ρ_j agree.

Due to the maximality of ρ_j, no substack of $\mathrm{pop}_2(m_j)$ is visited by ρ after $k := \mathrm{len}(\rho_j)$. Thus, the run $\pi := (\rho{\restriction}_{[k,\mathrm{len}(\rho)]})[m_j/m_i]$ is well-defined (cf. Definition 4.10). Note that π starts in (q', m_i) for $q' \in Q$ the final state of ρ_i. Thus, we can set $\hat{\rho} := \rho_i \circ \pi$. Since $w_i \equiv^z_n w_k$ and since $\equiv^z_n$ is a right congruence, it is clear that $\mathrm{top}_2(\hat{\rho}) \equiv^z_n \mathrm{top}_2(\rho)$. Since $0 < |w_j| - |w_i| < \alpha(n,z)$, it also follows directly that

$$|\mathrm{top}_2(\rho)| - \alpha(n,z) \leq |\mathrm{top}_2(\hat{\rho})| < |\mathrm{top}_2(\rho)|.$$

□

of step 2. The proof is by induction on the number of words in the last stack of ρ that have length $h := \mathrm{hgt}(\rho)$. Assume that ρ is some run such that

$$\mathrm{hgt}(\rho) > |\mathrm{top}_2(\rho)| + \mathrm{B}_{\mathrm{hgt}}.$$

In the following, we define several generalised milestones of the final stack s of ρ. An illustration of these definitions can be found in Figure 2.

Let $m \in \mathrm{MS}(s)$ be a milestone of the last stack of ρ such that $|\mathrm{top}_2(m)| = h$. For each $|\mathrm{top}_2(\rho)| \leq i \leq h$ let $m_i \in \mathrm{MS}(m)$ be the maximal milestone of m with $|\mathrm{top}_2(m_i)| = i$. Let n_i

be maximal such that $\rho(n_i) = (q', m_i)$ for some $q' \in Q$. Let $m'_i \in \mathrm{GMS}(s) \setminus \mathrm{GMS}(m)$ be the minimal generalised milestone after m such that $\mathrm{top}_2(m'_i) = \mathrm{top}_2(m_i)$. Let n'_i be maximal with $\rho(n'_i) = (q', m'_i)$ for some $q' \in Q$.

There are $|\mathrm{top}_2(\rho)| \leq k < l \leq \mathrm{hgt}(\rho)$ satisfying the following conditions.

(1) There is a $q \in Q$ such that $\rho(n_k) = (q, m_k)$ and $\rho(n_l) = (q, m_l)$.
(2) There is a $q' \in Q$ such that $\rho(n'_k) = (q', m'_k)$ and $\rho(n'_l) = (q', m'_l)$.
(3) $\mathrm{top}_2(m_k) \equiv^2_0 \mathrm{top}_2(m_l)$ (note that this implies that $\#\mathrm{Loop}^1(m_k) = \#\mathrm{Loop}^1(m_l)$ and $\#\mathrm{Ret}^1(m_k) = \#\mathrm{Ret}^1(m_l)$).

By definition, we have $m_l \trianglelefteq m'_l$. Thus, the run $\pi_1 := (\rho{\restriction}_{[n_l, n'_l]})[m_l/m_k]$ is well defined (cf. Definition 4.10). Note that π_1 starts in (q, m_k) and ends in $(q', \hat{s})$ for $\hat{s} := m'_l[m_l/m_k]$). Moreover, $\mathrm{top}_2(\pi_1) = \mathrm{top}_2(m'_k) = \mathrm{top}_2(\rho(n'_k))$. Furthermore, $\rho{\restriction}_{[n'_k, \mathrm{len}(\rho)]}$ never looks below the topmost word of m'_k because n'_k is the maximal node where the generalised milestone m'_k is visited. Thus, $(\mathrm{pop}_2(m'_k) : \bot) \trianglelefteq \rho{\restriction}_{[n'_k, \mathrm{len}(\rho)]}$ whence

$$\pi_2 := \rho{\restriction}_{[n'_k, \mathrm{len}(\rho)]}[\mathrm{pop}_2(m'_k) : \bot / \mathrm{pop}_2(\hat{s}) : \bot]$$

is well defined. It starts in the last stack of π_1. Now, we define the run $\hat{\rho} := \rho{\restriction}_{[0, n_k]} \circ \pi_1 \circ \pi_2$. Either $\mathrm{hgt}(\hat{\rho}) < \mathrm{hgt}(\rho)$ and we are done or there are less words of height $\mathrm{hgt}(\rho)$ in the last stack of $\hat{\rho}$ than in the last stack of ρ and we conclude by induction. □

of step 3. Assume that ρ is a run with $\mathrm{len}(\rho) > \beta(\mathrm{hgt}(\rho))$. We denote by n_i the maximal position in ρ such that the stack at $\rho(n_i)$ is $\mathrm{pop}_2^i(\rho)$ for each $0 \leq i \leq |\rho|$. There are less than $\frac{\beta(\mathrm{hgt}(\rho))}{|Q|}$ many words of length up to $\mathrm{hgt}(\rho)$. Thus, there are $i < j$ such that

(1) $\mathrm{top}_2\left(\mathrm{pop}_2^i(\rho)\right) = \mathrm{top}_2\left(\mathrm{pop}_2^j(\rho)\right)$, and
(2) $\rho(n_i) = \left(q, \mathrm{pop}_2^i(\rho)\right)$ and $\rho(n_j) = \left(q, \mathrm{pop}_2^j(\rho)\right)$ for some $q \in Q$.

Now, let $s_i := \mathrm{pop}_2^{i+1}(\rho)$ and $s_j := \mathrm{pop}_2^{j+1}(\rho)$. There is a unique stack s such that $\rho(\mathrm{len}(\rho)) = (\hat{q}, s_i : s)$. $\rho{\restriction}_{[n_i, \mathrm{len}(\rho)]}$ is a run from $\mathrm{pop}_2^i(\rho)$ to $s_i : s$ that never visits s_i. Thus,

$$\hat{\rho}_1 := \rho{\restriction}_{[n_i, \mathrm{len}(\rho)]}[s_i : \bot / s_j : \bot]$$

is a well defined run. The composition $\hat{\rho} := \rho{\restriction}_{[0, n_j]} \circ \hat{\rho}_1$ satisfies the claim. □

The previous corollary deals with the reachability of some stack from the initial configuration. The following proposition is concerned with the extension of a given stack by just one word. Recall that such a run corresponds to a $\curvearrowright$ edge. We first define the function that is used to bound the size of the new word. Recall that the equivalence relation $\equiv^z_n$ depends on the choice of the fixed 2-PS $\mathcal{N} = (Q, \Sigma, \Delta, q_0)$. $|{}^{\Sigma^*}/_{\equiv^z_n}|$ denotes the number of equivalence classes of $\equiv^z_n$.

Definition 5.26. Set $\gamma(a, b, c, d) := 1 + b + a(|Q||{}^{\Sigma^*}/_{\equiv^d_c}|)$.

Before we state the proposition concerning the compatibility of $\equiv^z_n$ with $\curvearrowright$ edges, we explain its meaning. The proposition says that given two equivalent words w and $\hat{w}$ and a run ρ from $(q, s : w)$ to $(q', s : w : w')$ that does not pass any substack of $s : w$, then, for each stack $\hat{s} : \hat{w}$, we find a run $\hat{\rho}$ from $(q, \hat{s} : \hat{w})$ to $(q', \hat{s} : \hat{w} : \hat{w}')$ for some short word $\hat{w}'$ that is equivalent to w'. Furthermore, this transfer of runs works simultaneously on a tuple of such runs, i.e., given m runs starting at $s : w$ of the form described above, we find m corresponding runs starting at $\hat{s} : \hat{w}$. This simultaneous transfer becomes important when we search an isomorphic copy of the relevant ancestors of several runs. In this case the simultaneous transfer allows us to copy the relevant ancestors of a certain run while avoiding an intersection with the relevant ancestors of other given runs.

Proposition 5.27. *Let $n, z, m \in \mathbb{N}$ such that $n \geq 1$, $z > m$, and $z \geq 2$. Let $c = (q, s : w)$,$\hat{c} = (q, \hat{s} : \hat{w})$ be configurations such that $w \equiv_n^z \hat{w}$. Let $\rho_1, \dots, \rho_m$ be pairwise distinct runs such that for each i, $|\rho_i(j)| > |s : w|$ for all $j \geq 1$ and such that ρ_i starts at c and ends in $(q_i, s : w : w_i)$. Analogously, let $\hat{\rho}_1, \dots, \hat{\rho}_{m-1}$ be pairwise distinct runs such that each $\hat{\rho}_i$ starts at $\hat{c}$ and ends in $(q_i, \hat{s} : \hat{w} : \hat{w}_i)$ and $|\hat{\rho}_i(j)| > |\hat{s} : \hat{w}|$ for all $j \geq 1$. If*

$$w_i \equiv_{n-1}^z \hat{w}_i \text{ for all } 1 \leq i \leq m-1,$$

then there is some run $\hat{\rho}_m$ from $\hat{c}$ to $(q_0, \hat{s} : \hat{w} : \hat{w}_m)$ such that

$$w_m \equiv_{n-1}^z \hat{w}_m,$$
$$\hat{\rho}_m \text{ is distinct from each } \hat{\rho}_i \text{ for } 1 \leq i < m, \text{ and}$$
$$|\hat{w}_m| \leq \gamma(m, |\hat{w}|, n, z).$$

We prepare the proof of this proposition with the following lemmas.

Lemma 5.28. *Let $z, m, n \in \mathbb{N}$ such that $z \geq 2$ and $z > m$. Let w, w' be words and let $\rho_1, \rho_2, \dots, \rho_m$ be pairwise distinct runs such that ρ_i starts in (q_i, w) and ends in $(\hat{q}_i, v_i)$ for some prefix $v_i \leq w$. If $w \equiv_{n+1}^z w'$, then there are prefixes $v'_1, v'_2, \dots, v'_m$ of w' such that $v_i \equiv_n^z v'_i$ for all $1 \leq i \leq m$ and there are pairwise distinct runs $\rho'_1, \rho'_2, \dots, \rho'_m$ such that ρ'_i starts in (q_i, w'), ends in $(\hat{q}_i, v'_i)$.*

Furthermore, $v_i = w$ if and only if $v'_i = w'$ and $v_i < w$ implies that there is a letter a_i and words u_i, u'_i such that $w = v_i a_i u_i$ and $w' = v'_i a_i u'_i$.

Proof. Without loss of generality, assume that $q_i = q_j = q$ and $\hat{q}_i = \hat{q}_j = \hat{q}$ for all $1 \leq i, j \leq m$. Since $\mathfrak{Lin}_{n+1}^{z;z}(w) \simeq_z \mathfrak{Lin}_{n+1}^{z;z}(w')$, a winning strategy in the Ehrenfeucht-Fraïssé game induces words $v'_1, v'_2, \dots, v'_m$ such that $(v_1, v_2, \dots, v_m) \mapsto (v'_1, v'_2, \dots, v'_m)$ is a partial isomorphism. Thus, $v'_i = v'_j$ iff $v_i = v_j$. Since $z > m$, Duplicator can maintain this partial isomorphism for at least one more round of the game. Therefore, the labels of the direct neighbours of v'_i agree with the labels of the direct neighbours of v_i which especially implies $v'_i = w'$ iff $v_i = w$. Furthermore, if $v_{i_1} = v_{i_2} = \dots = v_{i_k}$, then $\rho_{i_1}, \rho_{i_2}, \dots, \rho_{i_k}$ witness that v_{i_1} is coloured by $S^k_{(q,\hat{q})}$ in $\mathfrak{Lin}_{n+1}^{z;z}(w)$. Hence, v'_{i_1} is coloured by $S^k_{(q,\hat{q})}$ in $\mathfrak{Lin}_{n+1}^{z;z}(w')$. Hence, there are k pairwise distinct runs $\rho'_{i_1}, \rho'_{i_2}, \dots, \rho'_{i_k}$ from (q, w') to $(\hat{q}, v'_{i_k})$ that do not visit v'_{i_k} before their final configurations. Since, v_i and v'_i are labelled by the same $\equiv_n^z$-type, the claim follows immediately. □

This lemma provides the transfer of runs from some stack $s : w$ to stacks $s : v_i$ with $v_i \leq w$ to another starting stack $s' : w'$ if w and w' are equivalent words. We still need to investigate runs in the other direction. We provide a transfer property for runs from some word w to extensions $wv_1, wv_2, \dots, wv_m$.

Lemma 5.29. *Let $z, m, n \in \mathbb{N}$ such that $z \geq 2$ and $z > m$. Let $\rho_1, \rho_2, \dots, \rho_m$ be pairwise distinct runs such that for each $1 \leq i \leq m$ the run ρ_i starts in (q_i, w), ends in (q_i, wv_i) and never visits w after its initial configuration. Furthermore, let w' be some word such that $w \equiv_n^z w'$. There are words $v'_1, v'_2, \dots, v'_m$ such that $|v'_i| \leq 1 + m \cdot |Q| \cdot |\Sigma^*/_{\equiv_n^z}|$, the first letter of v_i and v'_i agree (or $v_i = v'_i = \varepsilon$), $wv_i \equiv_n^z w'v'_i$, and there are pairwise distinct runs $\rho'_1, \rho'_2 \dots, \rho'_m$ such that each run ρ'_i starts in w', ends in $w'v'_i$, and never visits w' after its initial configuration.*

Proof. For each run ρ_i, there is a decomposition $\rho_i = \pi_n \circ \lambda_n \circ \cdots \circ \pi_2 \circ \lambda_2 \circ \pi_1 \circ \lambda_1$ where the λ_i are high loops and each π_i is a run of length 1 that performs a push operation. Since the $\equiv^z_n$ type of a word w determines $\#\mathrm{Ret}^z(w)$ and $\#\mathrm{HLoop}^z(w)$, we conclude with Proposition 4.13 that $\#\mathrm{HLoop}^z(wv) = \#\mathrm{HLoop}^z(w'v)$ for all words $v \in \Sigma^*$. Thus, there is a run $\rho'_i = \pi_n \circ \lambda'_n \circ \cdots \circ \pi_2 \circ \lambda'_2 \circ \pi_1 \circ \lambda'_1$ where the λ'_i are high loops such that the runs $\rho'_1, \rho'_2, \dots, \rho'_m$ are pairwise distinct. Note that ρ'_i ends with stack $w'v_i$ and $w'v_i \equiv^z_n wv_i$ because $\equiv^z_n$ is a right congruence.

If $|v_i| \leq 1 + m \cdot |Q| \cdot |\Sigma^*/_{\equiv^z_n}|$ for all $1 \leq i \leq m$ we are done. Otherwise we continue with the following construction. Without loss of generality assume that $|v_1| > 1 + m \cdot |Q| \cdot |\Sigma^*/_{\equiv^z_n}|$. Then we find nonempty prefixes $u_0 < u_1 < u_2 < \cdots < u_m$ such that for all $0 \leq i < j \leq m$

(1) ρ'_1 passes $w'u_i$ and $w'u_j$ in the same state $\hat{q} \in Q$ for the last time,
(2) $w'u_i \equiv^z_n wu_j$, and
(3) $1 \leq |u_i| < |u_j| \leq 1 + m \cdot |Q| \cdot |\Sigma^*/_{\equiv^z_n}|$.

Let n_i be the maximal position in ρ'_1 such that $\rho'_1(n_i) = (\hat{q}, w'u_i)$. For each $0 \leq i < j \leq m$, we can define the run $\rho_1^{i,j} := \rho'_1{\restriction}_{[0,n_i]} \circ \rho'_1{\restriction}_{[n_j,\mathrm{len}(\hat{\rho}_1)]}[wu_j/wu_i]$ which ends in the stack $wv_1[wu_j/wu_i]$. Since $\equiv^z_n$ is a right congruence, this stack is equivalent to wv_1. Furthermore, it is shorter than wv_1. By pigeonhole principle, there are $0 \leq i < j \leq m$ such that $\rho_1^{i,j}$ is distinct from $\rho'_2, \rho'_3, \dots, \rho'_m$. Now, we replace ρ'_1 by $\rho_1^{i,j}$.

Repetition of this argument yields the claim. □

For the proof of Proposition 5.27, we now compose the previous lemmas. Recall that the proposition says the following: given m runs $\rho_1, \dots, \rho_m$ starting in some stack s that only add one word to s and given a stack $\hat{s}$ whose topmost word is $\equiv^z_n$-equivalent to the word on top of s, we can transfer the runs $\rho_1, \dots, \rho_m$ to runs $\rho'_1, \dots, \rho'_m$ that start at $\hat{s}$ such that ρ'_i extends $\hat{s}$ by one word that is $\equiv^z_{n-1}$-equivalent to the word created by ρ_i.

Proof of Proposition 5.27. Let $\rho_1, \rho_2, \dots, \rho_m$ and $\hat{\rho}_1, \hat{\rho}_2, \dots, \hat{\rho}_{m-1}$ be runs as required in the proposition. Assume that $w_i \equiv^z_{n-1} w_j$ and that all runs ρ_i end in the same state, i.e., $q_i = q_j$, for all $1 \leq i \leq j \leq m$. Later we deal with the other cases.

We decompose each run ρ_i as follows. Let $w'_i := w \sqcap w_i$. Then $\rho_i = \rho_i^0 \circ \rho_i^1 \circ \rho_i^2$ where ρ_i^0 is a run of length 1 that performs exactly one clone_2 operation, and ρ_i^1 is the run from $s : w : w$ to the last occurrence of $s : w : w'_i$.

Due to $\mathrm{top}_1(w) = \mathrm{top}_1(\hat{w})$, there are runs $\hat{\rho}_i^0$ from $\hat{c}$ to $\hat{s} : \hat{w} : \hat{w}$ performing only one clone operation and ending in the same state as ρ_i^0.

By Lemma 5.28, we can transfer the ρ_i^1 to runs $\hat{\rho}_i^1$ starting at $(q, \hat{s} : \hat{w} : \hat{w})$ and ending at $\hat{s} : \hat{w} : \hat{u}_i$ such that $\hat{u}_i \leq \hat{w}$ and $w \sqcap w_i \equiv^z_{n-1} \hat{u}_i$. The lemma allows us to enforce that $\hat{\rho}_i^1 = \hat{\rho}_j^1$ iff $\rho_i^1 = \rho_j^1$.

Let v_i be the word such that $w_i = (w \sqcap w_i) \circ v_i$. We use Lemma 5.29 and find words $\hat{v}_1, \dots, \hat{v}_m$ and runs $\hat{\rho}_1^2, \dots, \hat{\rho}_m^2$ such that $\hat{\rho}_i^2$ is a run from $\hat{s} : \hat{w} : \hat{u}_i$ to $(q_i, \hat{s} : \hat{w} : \hat{u}_i\hat{v}_i)$ which visits $\hat{s} : \hat{w} : \hat{u}_i$ only in its initial configuration such that $\hat{u}_i\hat{v}_i \equiv^z_{n-1} w_i$ and such that $\hat{u}_i\hat{v}_i$ has length bounded by

$$\gamma(m, |\hat{w}|, n, z) = |\hat{w}| + 1 + m \cdot |Q| \cdot |\Sigma^*/_{\equiv^z_n}| \,.$$

Furthermore, $\hat{\rho}_i^2$ and $\hat{\rho}_j^2$ coincide if and only if ρ_i^2 and ρ_j^2 coincide. we claim that the runs

$$\hat{\rho}_1^0 \circ \hat{\rho}_1^1 \circ \hat{\rho}_1^2, \hat{\rho}_2^0 \circ \hat{\rho}_2^1 \circ \hat{\rho}_2^2, \dots, \hat{\rho}_m^0 \circ \hat{\rho}_m^1 \circ \hat{\rho}_m^2$$

are pairwise distinct. First of all we show that $\hat{u}_i = \hat{w} \sqcap \hat{u}_i\hat{v}_i$: Due to the last part of Lemma 5.28, there is a letter a_i such that $w = u_i a_i x_i$ for some word x_i and $\hat{w} = \hat{u}_i a_i \hat{x}_i$ for some word $\hat{x}_i$. Furthermore, v_i and $\hat{v}_i$ start with the same letter. Due to $u_i = w \sqcap u_i v_i$, this letter cannot be a_i whence $\hat{u}_i = \hat{w} \sqcap \hat{u}_i\hat{v}_i$.

Heading for a contradiction, assume that $\hat{\rho}_i^0 \circ \hat{\rho}_i^1 \circ \hat{\rho}_i^2 = \hat{\rho}_j^0 \circ \hat{\rho}_j^1 \circ \hat{\rho}_j^2$. Since $\hat{\rho}_i^0$ and $\hat{\rho}_j^0$ have both length 1, this implies that $\hat{\rho}_i^0 = \hat{\rho}_j^0$. Furthermore, we have seen that $\hat{\rho}_i^1$ ends in the last occurrence of the greatest common prefix $\hat{w} \sqcap \hat{u}_i\hat{v}_i = \hat{u}_i$. Hence, the two runs can only coincide if $\hat{u}_i = \hat{u}_j$. But then $\hat{\rho}_i^1 = \hat{\rho}_j^1$ because both parts end in the last occurrence of a stack with topmost word $\hat{u}_i$. But this would also imply that $\hat{\rho}_i^2 = \hat{\rho}_j^2$. By construction of the three parts, this would imply that $\rho_i^0 = \rho_j^0$, $\rho_i^1 = \rho_j^1$, and $\rho_i^2 = \rho_j^2$. But this contradicts the assumption that $\rho_i^0 \circ \rho_i^1 \circ \rho_i^2 = \rho_i \neq \rho_j = \rho_j^0 \circ \rho_j^1 \circ \rho_j^2$.

Since the runs are all distinct, there is some j such that $\hat{\rho}_j^0 \circ \hat{\rho}_j^1 \circ \hat{\rho}_j^2$ does not coincide with any of the $\hat{\rho}_i$ for $1 \leq i \leq m-1$. Note that $\hat{\rho}_m := \hat{\rho}_j^0 \circ \hat{\rho}_j^1 \circ \hat{\rho}_j^2$ satisfies the claim of the proposition.

Now, we come to the case that the runs end in configurations with different states or different $\equiv_{n-1}^z$-types of their topmost words. In this case, we just concentrate on those ρ_i which end in the same state as ρ_m and with a topmost word of the same type as w_m. This is sufficient because some run ρ can only coincide with $\hat{\rho}_i$ if both runs end up in the same state and in stacks whose topmost words have the same type. □

6. Dynamic Small-Witness Property

In this section, we define a family of equivalence relations on tuples in 2-NPT. The equivalence class of a tuple $\rho_1, \dots, \rho_m$ with respect to one of these relations is the isomorphism type of the substructure induced by the relevant l-ancestors of $\rho_1, \dots, \rho_m$ extended by some information for preserving this isomorphism during an Ehrenfeucht-Fraïssé game. Recall that such a game ends in a winning position for Duplicator if the relevant 1-ancestors of the elements that were chosen in the two structures are isomorphic (cf. Lemma 5.5).

We then show how to construct small representatives for each equivalence class. As explained in Section 2.1, this result can be turned into an FO model checking algorithm on the class of 2-NPT.

Definition 6.1. Let $\bar{\rho} = (\rho_1, \rho_2, \dots, \rho_m)$ be runs of a 2-PS $\mathcal{N}$ and let $\mathfrak{N} := \mathrm{NPT}(\mathcal{N})$. Let $l, n_1, n_2, z \in \mathbb{N}$. We define the following relations on $\mathrm{RA}_l(\bar{\rho})$.

(1) For $k \leq l$ and $\rho \in \bar{\rho}$, let $P_\rho^k := \{\pi \in \mathrm{RA}_l(\bar{\rho}) : \pi \in \mathrm{RA}_k(\rho)\}$.
(2) Let ${}_{n_1}{\equiv}_{n_2}^z$-Type be the function that maps a run π to the ${}_{n_1}{\equiv}_{n_2}^z$-equivalence class of the last stack of π.

We write $\mathfrak{R}_{l,n_1,n_2,z}(\bar{\rho})$ for the following expansion of the relevant ancestors of $\bar{\rho}$:

$$\mathfrak{R}_{l,n_1,n_2,z}(\bar{\rho}) := (\mathfrak{N}{\restriction}_{\mathrm{RA}_l(\bar{\rho})}, (\vdash^\delta)_{\delta \in \Delta}, \curvearrowright, \curvearrowright, {}_{n_1}{\equiv}_{n_2}^z\text{-Type}, (P_{\rho_j}^k)_{k \leq l, 1 \leq j \leq m}).$$

For tuples of runs $\bar{\rho} = (\rho_1, \dots, \rho_m)$ and $\bar{\rho}' = (\rho_1', \dots, \rho_m')$ we set $\bar{\rho}\ {}^{l}_{n_1}{\equiv}_{n_2}^z\ \bar{\rho}'$ if

$$\mathfrak{R}_{l,n_1,n_2,z}(\bar{\rho}) \simeq \mathfrak{R}_{l,n_1,n_2,z}(\bar{\rho}').$$

Remark 6.2. • If $\bar{\rho}\ {}^{l}_{n_1}{\equiv}_{n_2}^z\ \bar{\rho}'$ then there is a unique isomorphism $\varphi : \mathfrak{R}_{l,n_1,n_2,z}(\bar{\rho}) \simeq \mathfrak{R}_{l,n_1,n_2,z}(\bar{\rho}')$ witnessing this equivalence: due to the predicate P_j^0, ρ_j is mapped to

ρ'_j for all $1 \leq j \leq m$. Due to the predicate P^l_j, the relevant ancestors of ρ_j are mapped to the relevant ancestors of ρ'_j. Finally, φ must preserve the order of the relevant ancestors of ρ_j because they form a chain with respect to $\vdash \cup \curvearrowright$ (cf. Proposition 5.12).
- Due to Lemma 5.5, it is clear that $\bar\rho \; {}^{l}_{n_1}{\equiv}^{z}_{n_2} \; \bar\rho'$ implies that there is a partial isomorphism mapping $\rho_i \mapsto \rho'_i$ for all $1 \leq i \leq m$.

Since equivalent relevant ancestors induce partial isomorphisms, a strategy that preserves the equivalence between relevant ancestors is winning for Duplicator in the Ehrenfeucht-Fraïssé-game.

Given a 2-PS $\mathcal{N}$, set $\mathfrak{N} := NPT(\mathcal{N})$. We show that there is a strategy in the Ehrenfeucht-Fraïssé game on $\mathfrak{N}, \bar\rho$ and $\mathfrak{N}, \bar\rho'$ in which Duplicator can always choose small elements compared to the size of the elements chosen so far in the structure where he has to choose. Furthermore, this strategy will preserve equivalence of the relevant ancestors in the following sense. Let $\bar\rho, \bar\rho' \subseteq \mathfrak{N}$ be the n-tuples chosen in the previous rounds of the game. Assume that Duplicator managed to maintain the relevant ancestors of these tuples equivalent, i.e., it holds that $\bar\rho \; {}^{l}_{k}{\equiv}^{z}_{n} \; \bar\rho'$. Now, Duplicator's strategy enforces that these tuples are extended by runs π and π' satisfying the following. There are numbers k_i, l_i, n_i such that $\bar\rho, \pi \; {}^{l_i}_{k_i}{\equiv}^{z}_{n_i} \; \bar\rho', \pi'$ and furthermore, the size of the run chosen by Duplicator is small compared to the elements chosen so far. Before we state the exact claim, we define some functions that provide bounds for Duplicator's choices.

Definition 6.3. Let $\mathcal{N}$ be a 2-PS and let $a, b \in \mathbb{N}$. We define the functions

$$\zeta : \mathbb{N}^5 \to \mathbb{N}, \qquad \eta : \mathbb{N}^5 \to \mathbb{N}, \text{ and} \qquad \theta : \mathbb{N}^5 \to \mathbb{N}$$

by induction on the first parameter. We set

$$\zeta(0, x_2, x_3, x_4, x_5) = \eta(0, x_2, x_3, x_4, x_5) = \theta(0, x_2, x_3, x_4, x_5) = 0 \text{ for all } x_2, x_3, x_4, x_5 \in \mathbb{N}.$$

For the inductive step, let $\bar x_{n+1} := (n+1, z, l', n'_1, n'_2) \in \mathbb{N}^5$ be arbitrary. We set $l := 4l' + 5$, $n_1 := n'_1 + 2(l'+1) + 1$, $n_2 := n'_2 + 4^{l'+1} + 1$, and $\bar x_n := (n, z, l, n_1, n_2)$. We define auxiliary values H^{loc}_i for $1 \leq i \leq 4^{l'+1}$ and H^{glob}_i for $1 \leq i \leq n'_1 + 4^{l'}$. Recall that we introduced α, β and $\mathrm{B_{hgt}}$ in Definition 5.23 and γ in Definition 5.26. Set

$$\begin{aligned}
H^{\mathrm{loc}}_1 &:= \gamma(n \cdot 4^{4l'+3}, \zeta(\bar x_n), n_2 - 1, z),\\
H^{\mathrm{loc}}_{i+1} &:= \gamma(1, H^{\mathrm{loc}}_i, n_1 - i + 1, z),\\
H^{\mathrm{glob}}_1 &:= \zeta(\bar x_n) + \mathrm{B_{hgt}} + \alpha(n'_2 + n'_1 + 4^{l'+1} - 1, z), \text{ and}\\
H^{\mathrm{glob}}_{i+1} &:= \gamma(1, H^{\mathrm{glob}}_i, n'_2 + n'_1 + 4^{l'+1} - i, z).
\end{aligned}$$

Now we set

$$\begin{aligned}
\zeta(\bar x_{n+1}) &:= \max\left\{H^{\mathrm{loc}}_{4^{l'+1}}, H^{\mathrm{glob}}_{n'_1+4^{l'}}\right\},\\
\eta(\bar x_{n+1}) &:= \eta(\bar x_n) + \beta(H^{\mathrm{glob}}_1) + n'_1 + 2(l'+1), \text{ and}\\
\theta(\bar x_{n+1}) &:= \theta(\bar x_n) + (4^{l'+1} + 1)\zeta(\bar x_{n+1}) \cdot \eta(\bar x_{n+1}) \cdot (1 + \mathrm{LL}^{\mathcal{N}}_z(\zeta(\bar x_{n+1}))).
\end{aligned}$$

where $\mathrm{LL}^{\mathcal{N}}_z$ is the function from Proposition 4.14 that bounds the length of short loops.

Remark 6.4. Since γ, α, β, $\mathrm{B_{hgt}}$, and LL depend computably on $\mathcal{N}$, the functions ζ, η and θ also depend computably on $\mathcal{N}$.

Proposition 6.5. *Let $\mathcal{N}$ be a 2-PS. Set $\mathfrak{N} := \mathrm{NPT}(\mathcal{N})$. Let $n, z, n_1', n_2', l' \in \mathbb{N}$, $l := 4l' + 5$, $n_1 := n_1' + 2(l'+1) + 1$, and $n_2 := n_2' + 4^{l'+1} + 1$ such that $z \geq 2$ and $z > n \cdot 4^l$. Furthermore, let $\bar{\rho}$ and $\bar{\rho}'$ be n-tuples of runs of $\mathfrak{N}$ such that*

1. $\bar{\rho} \ {}^{l}_{n_1}{\equiv}^{z}_{n_2} \ \bar{\rho}'$, *and*
2. $\mathrm{len}(\pi) \leq \theta(n, z, l, n_1, n_2)$ *for all* $\pi \in \mathrm{RA}_l(\bar{\rho}')$,
3. $\mathrm{hgt}(\pi) \leq \zeta(n, z, l, n_1, n_2)$ *for all* $\pi \in \mathrm{RA}_l(\bar{\rho}')$, *and*
4. $|\pi| \leq \eta(n, z, l, n_1, n_2)$ *for all* $\pi \in \mathrm{RA}_l(\bar{\rho}')$.

For each $\rho \in \mathfrak{N}$ there is some $\rho' \in \mathfrak{N}$ such that

1. $\bar{\rho}, \rho \ {}^{l'}_{n_1'}{\equiv}^{z}_{n_2'} \ \bar{\rho}', \rho'$,
2. $\mathrm{len}(\pi) \leq \theta(n+1, z, l', n_1', n_2')$ *for all* $\pi \in \mathrm{RA}_{l'}(\bar{\rho}', \rho')$,
3. $\mathrm{hgt}(\pi) \leq \zeta(n+1, z, l', n_1', n_2')$ *for all* $\pi \in \mathrm{RA}_l(\bar{\rho}', \rho')$, *and*
4. $|\pi| \leq \eta(n+1, z, l', n_1', n_2')$ *for all* $\pi \in \mathrm{RA}_l(\bar{\rho}', \rho')$.

This proposition can be reformulated as a finitary constraint for Duplicator's strategy in the Ehrenfeucht-Fraïssé game on every 2-NPT. This yields an FO model checking algorithm on 2-NPT. Before we present this application of the proposition in Section 7, we prove this proposition. For this purpose we split the claim into several pieces. The proposition asserts bounds on the length of the runs and on the sizes of the final stacks of the relevant ancestors. As the first step we prove that Duplicator has a strategy that chooses runs with small final stacks. This result relies mainly on the Propositions 5.22 and 5.27. These results allow us to construct equivalent relevant ancestor sets that only contain runs ending in small stacks. Afterwards, we apply the corollaries 4.16 and 4.17 in order to shrink the length of the runs involved.

6.1. **Construction of Isomorphic Relevant Ancestors.** Before we prove that Duplicator can choose short runs, we state some auxiliary lemmas concerning the construction of isomorphic relevant ancestors. The following lemma gives a sufficient criterion for the equivalence of the relevant ancestors of two runs. Afterwards, we show that for each run ρ we can construct a second run ρ' satisfying this criterion.

Lemma 6.6. *Let $\rho_0 \prec \rho_1 \prec \dots \prec \rho_m = \rho$ be runs such that $\mathrm{RA}_l(\rho) = \{\rho_i : 0 \leq i \leq m\}$. If $\hat{\rho}_0 \prec \hat{\rho}_1 \prec \dots \prec \hat{\rho}_m$ are runs such that*

- *the final states of ρ_i and $\hat{\rho}_i$ coincide,*
- $\rho_0 = \mathrm{pop}_2^l(\rho_m)$ *or* $|\rho_0| = |\hat{\rho}_0| = 1$,
- $\rho_0 \ {}_{n_1}{\equiv}^{z}_{n_2} \ \hat{\rho}_0$, *and*
- $\rho_i * \rho_{i+1}$ *iff* $\hat{\rho}_i * \hat{\rho}_{i+1}$ *for all* $1 \leq i < m$ *and* $* \in \{\curvearrowright\} \cup \{\vdash^{\delta} : \delta \in \Delta\}$,

then

$$\mathrm{RA}_l(\hat{\rho}_m) = \{\hat{\rho}_i : 0 \leq i \leq m\}.$$

If additionally $\mathrm{top}_2(\rho_i) \equiv^{z}_{n_2 - i} \mathrm{top}_2(\hat{\rho}_i)$ *for all $0 < i \leq m$, then*

$$\hat{\rho}_m \ {}^{l}_{n_1}{\equiv}^{z}_{n_2 - 4^l} \ \rho_m.$$

Proof. First, we show that for all $0 \leq i < j \leq m$, the following statements are true:

$$\rho_i \vdash^{\delta} \rho_j \text{ iff } \hat{\rho}_i \vdash^{\delta} \hat{\rho}_j, \tag{6.1}$$

$$\rho_i \rightsquigarrow \rho_j \text{ iff } \hat{\rho}_i \rightsquigarrow \hat{\rho}_j, \text{ and} \tag{6.2}$$

$$\rho_i \curvearrowright \rho_j \text{ iff } \hat{\rho}_i \curvearrowright \hat{\rho}_j. \tag{6.3}$$

Note that $\rho_i \vdash^\delta \rho_j$ implies $j = i + 1$. Analogously, $\hat\rho_i \vdash^\delta \hat\rho_j$ implies $j = i + 1$. Thus, (6.1) is true by definition of the sequences.

For the other parts, it is straightforward to see that $|\rho_k| - |\rho_j| = |\hat\rho_k| - |\hat\rho_j|$ for all $0 \leq j \leq k \leq m$: for $k = j$ the claim holds trivially. For the induction step from j to $j+1$, the claim follows from the assumption that $\rho_j * \rho_{j+1}$ if and only if $\hat\rho_j * \hat\rho_{j+1}$ for all $* \in \{\curvearrowright\} \cup \{\vdash^\delta : \delta \in \Delta\}$.

Furthermore, assume that there is some $\hat\pi$ such that $\hat\rho_k \prec \hat\pi \prec \hat\rho_{k+1}$. Then it cannot be the case that $\hat\rho_k \vdash^\delta \hat\rho_{k+1}$. This implies that $\rho_k \curvearrowright \rho_{k+1}$. By definition, it follows that $\hat\rho_k \curvearrowright \hat\rho_{k+1}$. We conclude directly that $|\hat\pi| \geq |\hat\rho_{k+1}| > |\hat\rho_k|$. Thus,

$$
\begin{aligned}
&\rho_j \rightsquigarrow \rho_k \text{ iff}\\
&|\rho_j| = |\rho_k| \text{ and } |\pi| > |\rho_j| \text{ for all } \rho_j \prec \pi \prec \rho_k \text{ iff}\\
&|\hat\rho_j| = |\hat\rho_k| \text{ and } |\hat\pi| > |\hat\rho_j| \text{ for all } \hat\rho_j \prec \hat\pi \prec \hat\rho_k \text{ iff}\\
&\hat\rho_j \rightsquigarrow \hat\rho_k.
\end{aligned}
$$

Analogously, one obtains (6.3).

We now show that $\mathrm{RA}_l(\hat\rho_m) = \{\hat\rho_i : 0 \leq i \leq m\}$. Note that

$$\mathrm{RA}_l(\hat\rho_m) \cap \{\pi : \hat\rho_m \leq \pi\} = \{\hat\rho_m\}$$

Assume that there is some $0 \leq m_0 \leq m$ such that

$$
\begin{aligned}
&\mathrm{RA}_l(\hat\rho_m) \cap \{\pi : \hat\rho_{m_0} \leq \pi\} = \{\hat\rho_i : m_0 \leq i \leq m\} \text{ and}\\
&\rho_i \in \mathrm{RA}_k(\rho) \text{ iff } \hat\rho_i \in \mathrm{RA}_k(\hat\rho_m) \text{ for all } k \leq l \text{ and } i \geq m_0.
\end{aligned}
$$

We distinguish the following cases.

- If $\rho_{m_0-1} \vdash^\delta \rho_{m_0}$ for some transition δ then $\hat\rho_{m_0-1} \vdash^\delta \hat\rho_{m_0}$ due to (6.1). Thus, there are no runs $\rho_{m_0-1} \prec \pi \prec \rho_{m_0}$. Hence, we only have to show that $\rho_{m_0-1} \in \mathrm{RA}_k(\rho_m)$ if and only if $\hat\rho_{m_0-1} \in \mathrm{RA}_k(\hat\rho_m)$ for all $k \leq l$.

  If $\rho_{m_0-1} \in \mathrm{RA}_k(\rho_m)$, then there is some $j \geq m_0$ such that $\rho_j \in \mathrm{RA}_{k-1}(\rho_m)$ and ρ_{m_0-1} is connected to ρ_j via some edge. But then $\hat\rho_j \in \mathrm{RA}_{k-1}(\hat\rho_m)$ and $\hat\rho_{m_0-1}$ is connected with $\hat\rho_j$ via the same sort of edge. Thus, $\hat\rho_{m_0-1} \in \mathrm{RA}_k(\hat\rho_m)$.

  The other direction is completely analogous.
- Otherwise, assume that there is some $\rho_{m_0-1} \prec \pi \prec \rho_{m_0}$. Since its direct predecessor is not in $\mathrm{RA}_l(\rho_m)$, $\rho_{m_0} \notin \mathrm{RA}_{l-1}(\rho)$. Thus, $\hat\rho_{m_0} \notin \mathrm{RA}_{l-1}(\hat\rho)$. By construction, $\hat\rho_{m_0-1} \curvearrowright \hat\rho_{m_0}$. Thus, $|\hat\pi| \geq |\hat\rho_{m_0}|$ for all $\hat\rho_{m_0-1} \prec \hat\pi \prec \hat\rho_{m_0}$. This implies that $\pi \not\rightsquigarrow \hat\rho_i$ and $\pi \not\curvearrowright \hat\rho_i$ for all $m_0 < i \leq m$. This shows that $\pi \notin \mathrm{RA}_l(\hat\rho_m)$.

  We obtain that $\hat\rho_{m_0-1} \in \mathrm{RA}_k(\hat\rho_m)$ iff $\rho_{m_0-1} \in \mathrm{RA}_k(\rho_m)$ for all $k \leq l$ analogously to the previous case.

Up to now, we have shown that $\mathrm{RA}_l(\hat\rho_m) \cap \{\pi : \hat\rho_0 \leq \pi\} = \{\hat\rho_i : 0 \leq i \leq m\}$. In order to prove $\mathrm{RA}_l(\hat\rho_m) = \{\hat\rho_i : 0 \leq i \leq m\}$, we have to show that $\hat\rho_0$ is the minimal element of $\mathrm{RA}_l(\hat\rho_m)$.

There are the following cases

(1) $\rho_0 = \mathrm{pop}_2^l(\rho_m)$. In this case, we conclude that $\hat\rho_0 = \mathrm{pop}_2^l(\hat\rho_m)$ by construction. But Lemma 5.7 then implies that $\hat\rho_0$ is the minimal element of $\mathrm{RA}_l(\hat\rho_m)$.

(2) $|\rho_0| = |\hat\rho_0| = 1$. Note that $\rho_0 \notin \mathrm{RA}_{l-1}(\rho_m)$ because ρ_0 is minimal in $\mathrm{RA}_l(\rho_m)$. Thus, we know that $\hat\rho_0 \notin \mathrm{RA}_{l-1}(\hat\rho_m)$.

Heading for a contradiction, assume that there is some $\hat\pi \in \mathrm{RA}_l(\hat\rho_m)$ with $\hat\pi \prec \hat\rho_0$. We conclude that $\hat\pi \rightsquigarrow \hat\rho_k$ or $\hat\pi \curvearrowright \hat\rho_k$ for some $\hat\rho_k \in \mathrm{RA}_{l-1}(\hat\rho_m)$. But this implies that $|\hat\pi| < |\hat\rho_0| = 1$. Since there are no stacks of width 0, this is a contradiction.

Thus, there is no $\hat\pi \in \mathrm{RA}_l(\hat\rho_m)$ that is a proper prefix of $\hat\rho_0$.

We conclude that $\mathrm{RA}_l(\hat\rho_m) = \{\hat\rho_i : 0 \le i \le m\}$.

Let us turn to the second part of the lemma. Assume that $\mathrm{top}_2(\rho_i) \equiv^z_{n_2-i} \mathrm{top}_2(\hat\rho_i)$ for all $0 \le i \le m$. Since $\hat\rho_i$ and $\hat\rho_{i+1}$ differ in at most one word, a straightforward induction shows that $\rho_i \;{}_{n_1-|\rho_0|+|\rho_i|}{\equiv}^z_{n_2-i}\; \hat\rho_i$ (cf. Proposition 5.22). But this implies $\hat\rho_m \;{}^{l}_{n_1}{\equiv}^z_{n_2-4^l}\; \rho_m$ because $|\rho_0| \le |\rho_i|$ as we have seen in Lemma 5.7. □

The previous lemma gives us a sufficient condition for the equivalence of relevant ancestors of two elements. Now, we show how to construct such a chain of relevant ancestors.

Lemma 6.7. *Let $l, n_1, n_2, m, z \in \mathbb{N}$ such that $n_2 \ge 4^l$ and $z \ge 2$. Let*

$$\rho_0 \prec \rho_1 \prec \dots \prec \rho_m = \rho \text{ be runs such that}$$
$$\mathrm{RA}_l(\rho) \cap \{\pi : \rho_0 \le \pi \le \rho\} = \{\rho_i : 0 \le i \le m\}.$$

Let $\hat\rho_0$ be a run such that $\rho_0 \;{}_{n_1}{\equiv}^z_{n_2}\; \hat\rho_0$. Then we can effectively construct runs

$$\hat\rho_0 \prec \hat\rho_1 \prec \dots \prec \hat\rho_m =: \hat\rho$$

such that

- *the final states of ρ_i and $\hat\rho_i$ coincide for all $0 \le i \le m$,*
- *$\rho_i \vdash^\delta \rho_{i+1}$ iff $\hat\rho_i \vdash^\delta \hat\rho_{i+1}$ and $\rho_i \curvearrowright \rho_{i+1}$ iff $\hat\rho_i \curvearrowright \hat\rho_{i+1}$ for all $0 \le i < m$, and*
- *$\mathrm{top}_2(\rho_i) \equiv^z_{n_2-i} \mathrm{top}_2(\hat\rho_i)$ for all $0 \le i \le m$.*

Proof. Assume that we have constructed

$$\hat\rho_0 \prec \hat\rho_1 \prec \dots \prec \hat\rho_{m_0},$$

for some $m_0 < m$ such that for all $0 \le i \le m_0$

(1) the final states of ρ_i and $\hat\rho_i$ coincide,
(2) $\rho_i \vdash^\delta \rho_{i+1}$ iff $\hat\rho_i \vdash^\delta \hat\rho_{i+1}$ and $\rho_i \curvearrowright \rho_{i+1}$ iff $\hat\rho_i \curvearrowright \hat\rho_{i+1}$ (note that either $\rho_i \vdash \rho_{i+1}$ or $\rho_i \curvearrowright \rho_{i+1}$ hold due to Proposition 5.12), and
(3) $\mathrm{top}_2(\rho_i) \equiv^z_{n_2-i} \hat\rho_i$.

We extend this chain by a new element ρ'_{m_0+1} such that all these conditions are again satisfied. We distinguish two cases.

First, assume that $\rho_{m_0} \vdash^\delta \rho_{m_0+1}$. Since $\rho_{m_0} \equiv^z_{n_2-m_0} \hat\rho_{m_0}$, $\mathrm{top}_1(\rho_{m_0}) = \mathrm{top}_1(\hat\rho_{m_0})$. Due to Condition 1, their final states also coincide. Hence, there is a $\hat\rho_{m_0+1}$ such that $\hat\rho_{m_0} \vdash^\delta \hat\rho_{m_0+1}$. Due to Proposition 5.22, $\hat\rho_{m_0+1}$ satisfies Condition (3).

Now, consider the case $\rho_{m_0} \curvearrowright \rho_{m_0+1}$. The run from ρ_{m_0} to ρ_{m_0+1} starts from some stack s and ends in some stack $s : w$ for w some word, the first operation is a clone and then s is never reached again. Hence, we can use Proposition 5.27 in order to find some appropriate $\hat\rho_{m_0+1}$ that satisfies Condition (3). □

The previous lemmas give us the possibility to construct an isomorphic copy of the relevant ancestors of a single run ρ. In our proofs, we want to construct such a copy while avoiding relevant ancestors of certain other runs. Using the full power of Proposition 5.27 we obtain the following stronger version of the lemma.

Corollary 6.8. *Let $l, n_1, n_2, m, z \in \mathbb{N}$ be numbers such that $z > m \cdot 4^l$ and $n_2 \ge 4^l$. Let $\bar\rho$ and $\bar\rho'$ be m-tuples such that $\bar\rho \;{}^{l}_{n_1}{\equiv}^z_{n_2}\; \bar\rho'$ and φ_l is an isomorphism witnessing this equivalence. Furthermore, let $\rho_0 \prec \rho_1 \prec \dots \prec \rho_m$ be runs such that for each $i < m$ either $\rho_i \vdash^{\rho}{}_{i+1}$ or $\rho_i \curvearrowright \rho_{i+1}$.*

If $\rho_0 \in \mathrm{RA}_l(\bar\rho)$, *and if* $\rho_1 \notin \mathrm{RA}_l(\bar\rho)$ *then we can construct* $\hat\rho_0 := \varphi_l(\rho_0) \prec \hat\rho_1 \prec \hat\rho_2 \prec \dots \prec \hat\rho_m$ *satisfying the conditions from the previous lemma but additionally with the property that* $\hat\rho_1 \notin \mathrm{RA}_l(\bar\rho')$.

Proof. We distinguish two cases.

(1) Assume that $\rho_0 \vdash \rho_1$. Due to the equivalence of ρ_0 and $\hat\rho_0$, we can apply the transition connecting ρ_0 with ρ_1 to $\hat\rho_0$ and obtain a run $\hat\rho_1$. We have to prove that $\hat\rho_1 \notin \mathrm{RA}_l(\bar\rho')$.

   Heading for a contradiction assume that $\hat\rho_1 \in \mathrm{RA}_l(\bar\rho')$. Then φ_l^{-1} preserves the edge between $\hat\rho_0$ and $\hat\rho_1$, i.e., $\rho_0 = \varphi_l^{-1}(\hat\rho_0) \vdash \varphi_l^{-1}(\hat\rho_1)$. But this implies that $\varphi_l^{-1}(\hat\rho_1) = \rho_1$ which contradicts the assumption that $\rho_1 \notin \mathrm{RA}_l(\bar\rho)$.

(2) Assume that $\rho_0 \curvearrowright \rho_1$. Up to threshold z, for each $\hat\pi$ such that $\hat\rho_0 \curvearrowright \hat\pi$ and $\hat\pi \in \mathrm{RA}_l(\bar\rho')$ there is a run $\rho_0 \curvearrowright \varphi_l^{-1}(\hat\pi)$. Since $\rho_1 \notin \mathrm{RA}_l(\bar\rho)$, we find another run $\hat\rho_1$ that satisfies the conditions of the previous lemma and $\hat\rho_1 \notin \mathrm{RA}_l(\bar\rho')$. This is due to the fact that Proposition 5.27 allows us to transfer up to $z > |\mathrm{RA}_l(\bar\rho')|$ many runs simultaneously.

□

6.2. **Construction of Small Equivalent Stacks.** In this section, we prove that Duplicator has a strategy that preserves the isomorphism type of the relevant ancestors while choosing runs whose relevant ancestors end in small stacks. Later, we show how to bound the length of such runs.

The analysis of this strategy decomposes into the local and the global case. We say Spoiler makes a local move if he chooses a new element such that one of its relevant ancestors is an relevant ancestor of the elements chosen so far. We say Spoiler makes a global move if he chooses an element such that its set of relevant ancestors does not intersect with the set of relevant ancestors of the elements chosen so far.

We first head for the result that Duplicator can manage the local case in such a way that he chooses an element such that all its relevant ancestors end in small stacks. Then we show that Duplicator can manage the global case analogously.

Lemma 6.9. *Let* $n, z, l', n_1', n_2' \in \mathbb{N}$ *be numbers such that* $z \geq 2$, $n_1' > 0$, $n_2' > 0$, $l := 4l' + 5$, $z > n \cdot 4^l$, $n_1 := n_1' + 2(l'+1) + 1$, *and* $n_2 := n_2' + 4^{l'+1} + 1$.

Let $\bar\rho, \bar\rho'$ *be* n*-tuples of runs such that* $\bar\rho \,{}^{l}_{n_1}{\equiv}^{z}_{n_2}\, \bar\rho'$ *and such that* $\mathrm{hgt}(\pi) \leq \zeta(n, z, l, n_1, n_2)$ *and* $|\pi| \leq \eta(n, z, l, n_1, n_2)$ *for all* $\pi \in \mathrm{RA}_l(\bar\rho')$. *Furthermore, let* ρ *be some run such that* $\mathrm{RA}_{l'+1}(\rho) \cap \mathrm{RA}_{l'+1}(\bar\rho) \neq \emptyset$. *Then there is some run* ρ' *such that*

$$\mathrm{hgt}(\rho') \leq \zeta(n+1, z, l', n_1', n_2'), \quad |\rho'| \leq \eta(n+1, z, l', n_1', n_2'), \text{ and } \quad (\bar\rho, \rho) \,{}^{l'}_{n_1'}{\equiv}^{z}_{n_2'}\, (\bar\rho', \rho').$$

Proof. Let $\varphi_l : \mathfrak{R}_{l,n_1,n_2,z}(\bar\rho) \simeq \mathfrak{R}_{l,n_1,n_2,z}(\bar\rho')$ denote the isomorphism that witnesses $\bar\rho \,{}^{l}_{n_1}{\equiv}^{z}_{n_2}\, \bar\rho'$. Let $\rho_0 \in \mathrm{RA}_{l'+1}(\rho)$ be maximal such that

$$\mathrm{RA}_{l'+1}(\rho) \cap \{\pi : \pi \leq \rho_0\} \subseteq \mathrm{RA}_{4l'+3}(\bar\rho) \subseteq \mathrm{RA}_l(\bar\rho).$$

There are numbers $m_0 \leq 0 \leq m_1$ and runs

$$\rho_{m_0} \prec \rho_{m_0+1} \prec \dots \prec \rho_0 \prec \rho_1 \prec \dots \prec \rho_{m_1}$$

such that $\mathrm{RA}_{l'+1}(\rho) = \{\rho_i : m_0 \leq i \leq m_1\}$. We set $\rho_i' := \varphi_l(\rho_i)$ for all $m_0 \leq i \leq 0$. Note that $\mathrm{hgt}(\rho_0') \leq \zeta(n, z, l, n_1, n_2)$ and $|\rho_0'| \leq \eta(n, z, l, n_1, n_2)$.

Next, we construct $\rho'_1, \dots, \rho'_{m_1}$ such that $\rho' := \rho'_{m_1}$ has relevant ancestors isomorphic to those of ρ. We first define ρ'_1 such that

(1) the final state of ρ'_1 and ρ_1 coincide,
(2) $\mathrm{top}_2(\rho'_1) \equiv^z_{n_2-1} \mathrm{top}_2(\hat{\rho}_1)$,
(3) $\rho'_0 \vdash^\delta \rho'_1$ iff $\rho_0 \vdash^\delta \rho_1$,
(4) $\rho'_0 \curvearrowright \rho'_1$ iff $\rho_0 \curvearrowright \rho_1$,
(5) $\rho'_1 \notin \mathrm{RA}_{4l'+3}(\bar{\rho}')$, and
(6) $\mathrm{hgt}(\rho'_1) \leq \gamma(n \cdot 4^{4l'+3}, \mathrm{hgt}(\rho'_0), n_2 - 1, z) \leq H^{\mathrm{loc}}_1$ (cf. Definition 6.3).

If $\rho_0 \vdash^\delta \rho_1$, this construction is trivial. Note that there is an element $\rho'_0 \vdash^\delta \rho'_1$ and $\rho'_1 \notin \mathrm{RA}_{4l'+3}(\bar{\rho}')$ (otherwise, $\rho_1 = \varphi_l^{-1}(\rho_1) \in \mathrm{RA}_{4l'+3}(\bar{\rho}')$ contradicting the maximality of ρ_0). If $\rho_0 \curvearrowright \rho_1$, we just apply Proposition 5.27.

Now, we continue constructing $\rho'_2, \dots, \rho'_{m_1} =: \rho'$ such that

(1) the final states of ρ_i and ρ'_i coincide for all $2 \leq i \leq m_1$,
(2) $\mathrm{top}_2(\rho'_i) \equiv^z_{n_2-i} \mathrm{top}_2(\hat{\rho}_i)$,
(3) for all $2 \leq i < m_1$, $\rho'_i \vdash^\delta \rho'_{i+1}$ if $\rho_i \vdash^\delta \rho_{i+1}$; in this case $\mathrm{hgt}(\rho'_{i+1}) \leq \mathrm{hgt}(\rho'_i) + 1$,
(4) for all $0 \leq i < m_1$, $\rho'_i \curvearrowright \rho'_{i+1}$ if $\rho_i \curvearrowright \rho_{i+1}$; in this case the use of Proposition 5.27 ensures that $\mathrm{hgt}(\rho'_{i+1}) \leq \gamma^{n,z}_{l',n_2}(1, \mathrm{hgt}(\rho'_i), n_2 - i, z)$

By definition, it is clear that conditions 1–3 hold also for all $m_0 \leq i < 0$. Using Lemma 6.6, we obtain that $\rho \ {}^{l'+1}_{n'_1}{\equiv}^z_{n'_2} \ \rho'$. A simple induction shows that $\mathrm{hgt}(\rho'_i) \leq H^{\mathrm{loc}}_i$ whence $\mathrm{hgt}(\rho') \leq \zeta(n+1, z, l', n'_1, n'_2)$. Furthermore, due to Lemma 5.9 $|\rho_i| - |\rho_0| \leq 2(l'+1)$ whence $|\rho'_i| \leq |\rho'_0| + 2(l'+1) \leq \eta(n+1, z, l', n'_1, n'_2)$.

We still have to show that the isomorphism between $\mathrm{RA}_l(\bar{\rho})$ and $\mathrm{RA}_l(\bar{\rho}')$ and the isomorphism between $\mathrm{RA}_{l'}(\rho)$ and $\mathrm{RA}_{l'}(\rho')$ are compatible in the sense that they induce an isomorphism between $\mathrm{RA}_{l'}(\bar{\rho}, \rho)$ and $\mathrm{RA}_{l'}(\bar{\rho}', \rho')$. The only possible candidate is

$$\varphi_{l'} : \mathrm{RA}_{l'}(\bar{\rho}, \rho) \to \mathrm{RA}_{l'}(\bar{\rho}', \rho')$$
$$\pi \mapsto \begin{cases} \rho'_i & \text{for } \pi = \rho_i, m_0 \leq i \leq m_1 \\ \varphi_l(\pi) & \text{for } \pi \in \mathrm{RA}_{l'+1}(\bar{\rho}). \end{cases}$$

In order to see that this is a well-defined function, we have to show that if $\rho_i \in \mathrm{RA}_{l'+1}(\bar{\rho})$ then $\rho'_i = \varphi_l(\rho_i)$ for each $m_0 \leq i \leq m_1$. Note that $\rho_i \in \mathrm{RA}_{l'+1}(\bar{\rho}) \cap \mathrm{RA}_{l'+1}(\rho)$ implies that $\pi \in \mathrm{RA}_{3l'+3}(\bar{\rho})$ for all $\pi \in \mathrm{RA}_{l'+1}(\rho)$ with $\pi \leq \rho_i$ (cf. Corollary 5.11). But then by definition $i \leq 0$ and $\rho'_i = \varphi_l(\rho_i)$.

We claim that $\varphi_{l'}$ is an isomorphism. Since we composed $\varphi_{l'}$ of existing isomorphisms $\mathrm{RA}_{l'}(\bar{\rho}) \simeq \mathrm{RA}_{l'}(\bar{\rho}')$ and $\mathrm{RA}_{l'}(\rho) \simeq \mathrm{RA}_{l'}(\rho')$, respectively, we only have to consider the following question: let $\pi \in \mathrm{RA}_{l'}(\bar{\rho})$ and $\hat{\pi} \in \mathrm{RA}_{l'}(\rho)$; does $\varphi_{l'}$ preserve edges between π and $\hat{\pi}$ and does and $\varphi_{l'}^{-1}$ preserve edges between the images of π and $\hat{\pi}$? In other words, we have to show that for each

$$* \in \{\rightarrow, \leftarrow, \curvearrowright, \curvearrowleft\} \cup \{\vdash^\delta : \delta \in \Delta\} \cup \{\dashv^\delta : \delta \in \Delta\}, \text{ we have } \pi * \hat{\pi} \text{ iff } \varphi_{l'}(\pi) * \varphi_{l'}(\hat{\pi}).$$

The following case distinction treats all these cases.

- Assume that there is some $* \in \{\rightarrow, \curvearrowright\} \cup \{\vdash^\delta : \delta \in \Delta\}$ such that $\pi * \hat{\pi}$. Then $\pi \in \mathrm{RA}_{l'+1}(\rho)$. Thus, there are $m_0 \leq i < j \leq m_1$ such that $\pi = \rho_i$ and $\hat{\pi} = \rho_j$. We have already seen that then $\varphi_{l'}(\pi) = \rho'_i$ and $\varphi_{l'}(\hat{\pi}) = \rho'_j$ and these elements are connected by an edge of the same type due to the construction of ρ'_i and ρ'_j.

- Assume that there is some $* \in \{\hookleftarrow, \Lsh\} \cup \{\dashv^\delta : \delta \in \Delta\}$ such that $\pi * \hat{\pi}$. Then $\hat{\pi} \in \mathrm{RA}_{l'+1}(\bar{\rho})$ whence $\varphi_{l'}$ coincides with the isomorphism φ_l on π and $\hat{\pi}$. But φ_l preserves edges whence $\pi * \hat{\pi}$ implies $\varphi_{l'}(\pi) * \varphi_{l'}(\hat{\pi})$.
- Assume that there is some $* \in \{\curvearrowright, \Rsh\} \cup \{\vdash^\delta : \delta \in \Delta\}$ such that $\varphi_{l'}(\pi) * \varphi_{l'}(\hat{\pi})$. By definition, $\varphi_{l'}(\hat{\pi}) \in \mathrm{RA}_{l'}(\rho')$ whence $\varphi_{l'}(\hat{\pi}) = \rho'_j$ for some $m_0 \leq j \leq m_1$. Thus, $\varphi_{l'}(\pi) \in \mathrm{RA}_{l'+1}(\rho')$ whence $\varphi_{l'}(\pi) = \rho'_i$ for some $m_0 \leq i < j$. We claim that $\pi = \rho_i$. Note that due to Corollary 5.11 for all $m_0 \leq k \leq i$ we have $\rho'_k \in \mathrm{RA}_{3l'+3}(\varphi_{l'}(\pi))$. Since $\varphi_{l'}(\pi) = \varphi_l(\pi) \in \mathrm{RA}_{l'}(\bar{\rho}')$, we conclude that $\rho'_k \in \mathrm{RA}_{4l'+3}(\bar{\rho}')$ for all $m_0 \leq k \leq i$. By construction, this implies $i \leq 0$ and $\varphi_{l'}(\pi) = \rho'_i = \varphi_l(\rho_i)$. Furthermore, since $\pi \in \mathrm{RA}_{l'}(\bar{\rho})$, $\varphi_{l'}(\pi) = \varphi_l(\pi)$. Since φ_l is an isomorphism, it follows that $\pi = \rho_i$. But this implies that there is an edge from $\pi = \rho_i$ to $\hat{\pi} = \rho_j$.
- Assume that there is some $* \in \{\hookleftarrow, \Lsh\} \cup \{\dashv^\delta : \delta \in \Delta\}$ such that $\varphi_{l'}(\pi) * \varphi_{l'}(\hat{\pi})$. This implies
$$\varphi_{l'}(\hat{\pi}) \in \mathrm{RA}_{l'+1}(\bar{\rho}') \cap \mathrm{RA}_{l'+1}(\rho). \tag{6.4}$$
By definition, $\hat{\pi} = \rho_j$ and $\varphi_{l'}(\hat{\pi}) = \rho'_j$ for some $m_0 \leq j \leq m_1$. Due to (6.4), $\rho'_i \in \mathrm{RA}_{4l'+3}(\bar{\rho}')$ for all $m_0 \leq i \leq j$. Since $\rho'_1 \notin \mathrm{RA}_{4l'+3}(\bar{\rho}')$, $j \leq 0$. Thus, $\rho_j \in \mathrm{RA}_{4l'+3}(\bar{\rho})$ and $\varphi_{l'}(\hat{\pi}) = \varphi_l(\hat{\pi})$. Since φ_l preserves the relevant ancestors of $\bar{\rho}$ level by level, we obtain that $\hat{\pi} \in \mathrm{RA}_{l'+1}(\bar{\rho})$. Since $\pi \in \mathrm{RA}_{l'+1}(\bar{\rho})$, we obtain that $\varphi_{l'}(\pi) = \varphi_l(\pi)$ and $\varphi_{l'}(\hat{\pi}) = \varphi_l(\hat{\pi})$. Since φ_l is an isomorphism, we conclude that $\pi * \hat{\pi}$

Thus, we have shown that $\varphi_{l'}$ is an isomorphism witnessing $\bar{\rho}, \rho \ {}^{l'}_{n'_1}{\equiv}^{z}_{n'_2} \ \bar{\rho}', \rho'$. □

The previous lemma shows that Duplicator can respond to local moves in such a way that she preserves isomorphisms of relevant ancestors while choosing small stacks. In the following we deal with global moves of Spoiler. We present a strategy for Duplicator that answers a global move by choosing a run with the following property. Duplicator chooses a run such that the isomorphism of relevant ancestors is preserved and such that all relevant ancestors of Duplicator's choice end in small stacks. We split this proof into two lemmas. First, we address the problem that Spoiler may choose an element far away from $\bar{\rho}$ but close to $\bar{\rho}'$. In this situation, Duplicator has to find a run that has isomorphic relevant ancestors and which is far away from $\bar{\rho}'$. Afterwards, we show that Duplicator can even choose such an element whose relevant ancestors all end in small stacks.

Lemma 6.10. *Let $n, l', n'_1, n'_2 \in \mathbb{N}$ be numbers. We set $l := 4l' + 5$, $n_1 := n'_1 + 2(l'+1) + 1$, and $n_2 := n'_2 + 4^{l'+1} + 1$. Let $z \in \mathbb{N}$ satisfy $z > n \cdot 4^l$ and $z \geq 2$.*

Let $\bar{\rho}$ and $\bar{\rho}'$ be n-tuples of runs such that $\bar{\rho} \ {}^{l}_{n_1}{\equiv}^{z}_{n_2} \ \bar{\rho}'$. Furthermore, let ρ be a run such that $\mathrm{RA}_{l'+1}(\bar{\rho}) \cap \mathrm{RA}_{l'+1}(\rho) = \emptyset$. Then there is some run ρ' such that $\bar{\rho}, \rho \ {}^{l'}_{n'_1}{\equiv}^{z}_{n'_2} \ \bar{\rho}', \rho'$.

Proof. Let φ_l be the isomorphism witnessing $\mathrm{RA}_l(\bar{\rho}) \ {}^{l}_{n_1}{\equiv}^{z}_{n_2} \ \mathrm{RA}_l(\bar{\rho}')$. If $\mathrm{RA}_{l'+1}(\bar{\rho}') \cap \mathrm{RA}_{l'+1}(\rho) = \emptyset$, we can set $\rho' := \rho$ and we are done. Otherwise, let $\pi^0_0 < \pi^0_1 < \dots < \pi^0_{n_0}$ be an enumeration of all elements of $\mathrm{RA}_{l'+1}(\bar{\rho}') \cap \mathrm{RA}_{l'+1}(\rho)$. Due to Corollary 5.14, $\mathrm{RA}_{l'+1}(\rho) \cap \{\pi : \pi \leq \pi^0_{n_0}\} \subseteq \mathrm{RA}_{3l'+3}(\bar{\rho}')$. Since $l > 3l' + 3$, we can set $\pi^1_i := \varphi_l^{-1}(\pi^0_i)$ for all $0 \leq i \leq n_0$. Due to Lemma 6.7, there is an extension $\pi^1_{n_0} < \rho^1$ such that $\mathrm{RA}_{l'+1}(\rho) \ {}^{l'+1}_{n'_1}{\equiv}^{z}_{n'_2} \ \mathrm{RA}_{l'+1}(\rho^1)$ and $\pi^1_i \in \mathrm{RA}_{l'+1}(\rho^1)$ for all $0 \leq i \leq n_0$. If $\mathrm{RA}_{l'+1}(\rho^1) \cap \mathrm{RA}_{l'+1}(\bar{\rho}') = \emptyset$ we set $\rho' := \rho^1$ and we are done.

Otherwise we can repeat this process, defining $\pi^2_i := \varphi_l^{-1}(\pi^1_i)$ for the maximal $n_1 \leq n_0$ such that $\pi^1_i \in \mathrm{RA}_{3l'+3}(\bar{\rho}')$ for all $0 \leq i \leq n_0$. Then we extend this run to some run ρ^2. If this process terminates with the construction of some run ρ^i such that $\mathrm{RA}_{l'+1}(\rho^i) \cap \mathrm{RA}_{l'+1}(\bar{\rho}') =$

$\emptyset$, we set $\rho' := \rho^i$ and we are done. If this is not the case, recall that $\mathrm{RA}_{3l'+3}(\bar\rho')$ is finite. Thus, we eventually reach the step were we have defined $\pi_0^0, \pi_0^1, \dots, \pi_0^m$ for some $m \in \mathbb{N}$ such that for the first time $\pi_0^m = \pi_0^i$ for some $i < m$. But if $i > 0$, then

$$\pi_0^{m-1} = \varphi_l(\pi_0^m) = \varphi_l(\pi_0^i) = \pi_0^{i-1}.$$

But this contradicts the minimality of m. We conclude that $\pi_0^m = \pi_0^0$ which implies that $\pi_0^0 \in \mathrm{RA}_{3l'+3}(\bar\rho)$. Furthermore, by definition we have $\pi_0^0 \in \mathrm{RA}_{l'+1}(\rho)$ and there is a maximal i such that $\pi_i^0 \in \mathrm{RA}_{3l'+3}(\bar\rho)$. Since $z > |\mathrm{RA}_l(\bar\rho)|$, we can apply Corollary 6.8 and construct a chain $\varphi_l(\pi_i^0) \prec \rho'_{i+1} \prec \rho'_{i+2} \prec \dots \prec \rho'$ such that $\bar\rho, \rho \;{}^{l'}_{n_1'}{\equiv}^z_{n_2'}\; \bar\rho', \rho'$. □

We have seen that that Duplicator can answer every global challenge of Spoiler. But we still need to prove that she can choose an element whose relevant ancestors all end in small stacks. The use of the pumping construction from Lemma 5.25 allows to prove this fact.

Lemma 6.11. *Let $n, l', n_1', n_2', z \in \mathbb{N}$, let $l := 4l' + 5$, $n_1 := n_1' + 2(l'+1) + 1$, and let $n_2 := n_2' + 4^{l'+1} + 1$. Furthermore, let $\bar\rho$ be an n-tuple of runs such that $\mathrm{hgt}(\pi) \le \zeta(n, z, l, n_1, n_2)$ and $|\pi| \le \eta(n, z, l, n_1, n_2)$ for all $\pi \in \mathrm{RA}_l(\bar\rho)$. If ρ is a run such that $\mathrm{RA}_{l'+1}(\bar\rho) \cap \mathrm{RA}_{l'+1}(\rho) = \emptyset$, then there is some run ρ' such that*

$$\begin{aligned}&\bar\rho, \rho \;{}^{l'}_{n_1}{\equiv}^z_{n_2}\; \bar\rho, \rho',\\ &\mathrm{hgt}(\pi) \le \zeta(n+1, z, l', n_1', n_2'), \text{ and}\\ &|\pi| \le \eta(n+1, z, l', n_1', n_2')\end{aligned}$$

for all $\pi \in \mathrm{RA}_{l'}(\bar\rho, \rho')$.

Proof. Let $\rho_0 \prec \rho_1 \prec \dots \prec \rho_m := \rho$ be runs such that $\mathrm{RA}_{l'+1}(\rho) = \{\rho_i : 0 \le i \le m\}$.

Let $m_0 > -n_1'$ be minimal such that there are runs $\rho_{m_0} \prec \rho_{m_0+1} \prec \dots \prec \rho_0$ such that for each $m_0 \le i < 0$ either $\rho_i \curvearrowright \rho_{i+1}$ or $\rho_i \vdash^\delta \rho_{i+1}$ for some clone_2 transition δ. Due to the construction, either $m_0 = 1 - n_1'$ or $|\rho_0| < n_1'$.

We construct runs $\rho'_{m_0} \prec \rho'_{m_0+1} \prec \dots \prec \rho'_m$ ending in small stacks such that $\rho'_m \;{}^{l'}_{n_1'}{\equiv}^z_{n_2'}\; \rho_m$ as follows. If $\mathrm{hgt}(\rho_{m_0}) \le \zeta(n, z, l, n_1, n_2)$ and $|\rho_{m_0}| \le \eta(n, z, l, n_1, n_2)$, then we set $\rho'_{m_0} := \rho_{m_0}$.

Otherwise, Lemma 5.25 provides a run ρ'_{m_0} such that

$$\begin{aligned}&\mathrm{hgt}(\rho'_{m_0}) \le H_1^{\mathrm{glob}} := \zeta(n, z, l, n_1, n_2) + \mathrm{B}_{\mathrm{hgt}} + \alpha(n_2' + n_1' + 4^{l'+1} - 1)\\ &|\rho'_{m_0}| \le \eta(n, z, l, n_1, n_2) + \beta(H_1^{\mathrm{glob}}),\\ &\text{either } \mathrm{hgt}(\rho'_{m_0}) > \zeta(n, z, l, n_1, n_2) \text{ or } |\rho'_{m_0}| > \eta(n, z, l, n_1, n_2), \text{ and}\\ &\rho_{m_0} \;{}_1{\equiv}^z_{n_2'+n_1'+4^{l'+1}-1}\; \rho'_{m_0}.\end{aligned}$$

The last condition just says that $\mathrm{top}_2(\rho_{m_0}) \equiv^z_{n_2'+n_1'+4^{l'}-1} \mathrm{top}_2(\rho'_{m_0})$.

Having constructed ρ'_i for $i < m$, we construct ρ'_{i+1} as follows.

(1) If $\mathrm{hgt}(\rho_j) \le \zeta(n, z, l, n_1, n_2)$ and $|\rho_j| \le \eta(n, z, l, n_1, n_2)$ for all $m_0 \le j \le i+1$, set $\rho'_{i+1} := \rho_{i+1}$.
(2) Otherwise, if $\rho_i \vdash^\delta \rho_{i+1}$ then define ρ'_{i+1} such that $\rho'_i \vdash^\delta \rho'_{i+1}$.
(3) If none of the previous cases applies, then $\rho_i \curvearrowright \rho_{i+1}$ and using Proposition 5.27 we construct ρ'_{i+1} such that
   (a) $\rho'_i \curvearrowright \rho'_{i+1}$,

(b) $\mathrm{hgt}(\rho'_{i+1}) \le \gamma(1, \max\{\mathrm{hgt}(\rho'_i), \zeta(n,z,l,n_1,n_2)\}, n'_2 + n'_1 - (i+1-m_0))$,
(c) $\mathrm{hgt}(\rho'_{i+1}) > \zeta(n,z,l,n_1,n_2)$ if $\mathrm{hgt}(\rho_j) \le \zeta(n,z,l,n_1,n_2)$ for all $m_0 \le j \le i$ and if $|\rho'_j| \le \eta(n,z,l,n_1,n_2)$ for all $m_0 \le j \le i+1$, and
(d) $\mathrm{top}_2(\rho_{i+1}) \equiv^z_{n'_2+n'_1+4^{l'+1}-(i+1-m_0)-1} \mathrm{top}_2(\rho'_{i+1})$.

First of all, note that $n'_2 + n'_1 + 4^{l'+1} - (0-m_0) - 1 \ge n'_2 + 4^{l'+1}$ whence $\rho_0 \ {}_{n'_1}\!\equiv^z_{n'_2+4^{l'+1}} \rho'_0$. Since $m \le 4^{l'+1}$ and since $|\rho'_j| \ge |\rho'_0|$ for all $0 \le j \le m$, we conclude that $\rho_j \ {}_{n'_1}\!\equiv^z_{n'_2} \rho'_j$ for all $0 \le j \le m$. Using Lemma 6.6, we see that $\rho' \ {}^{l'+1}_{n'_1}\!\equiv^z_{n'_2} \rho$.

Furthermore, $\mathrm{RA}_{l'+1}(\rho') \cap \mathrm{RA}_{l'+1}(\bar\rho) = \emptyset$: heading for a contradiction, assume that

$$\rho'_i \in \mathrm{RA}_{l'+1}(\rho') \cap \mathrm{RA}_{l'+1}(\bar\rho)$$

for some $0 \le i \le m$. Then $\rho'_j \in \mathrm{RA}_{3l'+3}(\bar\rho) \subseteq \mathrm{RA}_l(\bar\rho)$ for all $0 \le j \le i$. Thus, $\mathrm{hgt}(\rho'_j) \le \zeta(n,z,l,n_1,n_2)$ and $|\rho'_j| \le \eta(n,z,l,n_1,n_2)$ for all $0 \le j \le i$. By construction, it follows that $\rho'_j = \rho_j$ for all $m_0 \le i \le j$. But then $\rho_j = \rho'_j \in \mathrm{RA}_{l'+1}(\rho) \cap \mathrm{RA}_{l'+1}(\bar\rho)$ which contradicts $\mathrm{RA}_{l'+1}(\rho) \cap \mathrm{RA}_{l'+1}(\bar\rho) = \emptyset$.

Thus, $\mathrm{RA}_{l'}(\bar\rho)$ and $\mathrm{RA}_{l'}(\rho')$ do not touch whence $\bar\rho, \rho \ {}^{l'}_{n_1}\!\equiv^z_{n_2} \bar\rho, \rho'$. □

6.3. **Construction of Short Equivalent Runs.** Combining the results of the previous section, we obtain that for each n-tuple in $\mathfrak{N}(\mathcal{S})$ there is an FO_k-equivalent one such that the relevant ancestors of the second tuple only contains runs that end in small stacks. In order to prove Proposition 6.5, we still have to bound the length of these runs. For this purpose, we use Corollaries 4.16 and 4.17 in order to replace long runs between relevant ancestors by shorter ones.

of Proposition 6.5. Using the Lemmas 6.9 – 6.11, we find some candidate $\hat\rho'$ such that $\bar\rho, \rho \ {}^{l'}_{n'_1}\!\equiv^z_{n'_2} \bar\rho', \hat\rho'$ and the height and width of the last stacks of all $\pi \in \mathrm{RA}_{l'+1}(\hat\rho')$ are bounded by $\zeta(n+1, z, l', n'_1, n'_2)$ and $\eta(n+1, z, l', n'_1, n'_2)$, respectively.

Recall that there is a chain $\hat\rho'_0 \prec \hat\rho'_1 \prec \dots \prec \hat\rho'_m = \hat\rho'$ such that $\mathrm{RA}_{l'+1}(\hat\rho') = \{\hat\rho'_i : 0 \le i \le m\}$. This chain satisfies $0 \le m \le 4^{(l'+1)}$ and $\hat\rho'_i \vdash^\delta \hat\rho'_{i+1}$ or $\hat\rho'_i \curvearrowright \hat\rho'_{i+1}$ for all $0 \le i < m$.

If $\hat\rho'_0 \notin \mathrm{RA}_{3l'+3}(\bar\rho')$, then we can use Corollary 4.16 and choose some ρ'_0 that ends in the same configuration as $\hat\rho'_0$ such that $\rho'_0 \notin \mathrm{RA}_{3l'+3}(\bar\rho')$ and

$$\mathrm{len}(\rho'_0) \le 1 + 2 \cdot \zeta(n+1, z, l', n'_1, n'_2) \cdot \eta(n+1, z, l', n'_1, n'_2)) \cdot (1 + \mathrm{LL}^{\mathcal{N}}_z(\zeta(n+1, z, l', n'_1, n'_2))).$$

If $\hat\rho'_0 \in \mathrm{RA}_{3l'+3}(\bar\rho')$ let $0 \le i \le m$ be maximal such that $\hat\rho'_i \in \mathrm{RA}_l(\bar\rho')$. In this case let $\rho'_j := \hat\rho'_j$ for all $0 \le j \le i$.

By now, we have obtained a chain $\rho'_0 \prec \rho'_1 \prec \dots \prec \rho'_i$ for some $0 \le i \le m$. Using Corollary 4.17, we can extend this chain to a chain $\{\rho'_i : 0 \le i \le m\}$ such that

(1) $\rho'_i(\mathrm{len}(\rho'_i)) = \hat\rho'_i(\mathrm{len}(\hat\rho'_i))$, i.e. ρ'_i and $\hat\rho'_i$ end in the same configuration,
(2) $\rho'_i \vdash^\delta \rho'_{i+1}$ iff $\hat\rho'_i \vdash^\delta \hat\rho'_{i+1}$ for all $0 \le i < m$,
(3) $\rho'_i \curvearrowright \rho'_{i+1}$ iff $\hat\rho'_i \curvearrowright \hat\rho'_{i+1}$ for all $0 \le i < m$,
(4) $\mathrm{len}(\rho'_{i+1}) \le \mathrm{len}(\rho'_i) + 2 \cdot \zeta(n+1, z, l', n'_1, n'_2) \cdot (1 + \mathrm{LL}^{\mathcal{N}}_z(\zeta(n+1, z, l', n'_1, n'_2)))$, and
(5) $\hat\rho'_j \in \mathrm{RA}_{3l'+3}(\bar\rho')$ for all $0 \le j \le i$ implies $\rho'_i = \hat\rho'_i$ (here we use that $z > n \cdot 4^{3l'+3}$).

Using Lemma 6.6, we conclude that $\rho' \ {}^{l'+1}_{n'_1}\!\equiv^z_{n'_2} \hat\rho'$ for $\rho' := \rho'_m$. Furthermore, we claim that $\mathrm{RA}_{l'+1}(\bar\rho') \cap \mathrm{RA}_{l'+1}(\hat\rho') = \mathrm{RA}_{l'+1}(\bar\rho') \cap \mathrm{RA}_{l'+1}(\rho')$. By definition the inclusion from left to right is clear. For the other direction, assume that there is some element $\rho'_i \in \mathrm{RA}_{l'+1}(\bar{\rho}') \cap \mathrm{RA}_{l'+1}(\rho')$. By Lemma 5.11, this implies that $\rho'_j \in \mathrm{RA}_{3l'+3}(\bar{\rho}')$ for all $0 \leq j \leq i$. Thus, $\rho'_i = \hat{\rho}'_i$, which implies that $\rho'_i \in \mathrm{RA}_{l'+1}(\bar{\rho}') \cap \mathrm{RA}_{l'+1}(\rho')$.

We conclude that $\bar{\rho}, \rho \;{}^{l'}_{n'_1}{\equiv}^{z}_{n'_2}\; \bar{\rho}', \hat{\rho}' \;{}^{l'}_{n'_1}{\equiv}^{z}_{n'_2}\; \bar{\rho}', \rho'$. By definition, the length of ρ' is bounded by $\theta(n+1, z, l', n'_1, n'_2)$ (cf. Definition 6.3). □

7. FO Model Checking Algorithm for Level 2 Nested Pushdown Trees

Fix a 2-PS $\mathcal{N}$ and set $\mathfrak{N} := \mathrm{NPT}(\mathcal{N})$. We have shown that if $\mathfrak{N} \vDash \exists x \varphi(x)$ then there is a small run ρ such that $\mathfrak{N} \vDash \varphi(\rho)$. Even when we add parameters $\rho_1, \dots, \rho_n$ this result still holds, i.e., there is a short witness ρ compared to the length of the parameters. Hence, we can decide FO on 2-NPT with the following algorithm.

(1) Given a 2-PS $\mathcal{N}$ and a first-order sentence φ, the algorithm first computes the quantifier rank q of φ.
(2) Then it computes numbers $z, l^1, l^2, l^3, \dots, l^q, n_1^1, n_1^2, n_1^3 \dots, n_1^q, n_2^1, n_2^2, n_2^3, \dots, n_2^q \in \mathbb{N}$ such that for each $i < q$ the numbers $z, l^i, l^{i+1}, n_1^i, n_1^{i+1}, n_2^i, n_2^{i+1}$ can be used as parameters in Proposition 6.5.
(3) These numbers define a constraint $S = (S^{\mathfrak{N}}(i))_{i \leq q}$ for Duplicator's strategy in the q-round game on $\mathfrak{N}$ and $\mathfrak{N}$ as follows. We set $(\rho_1, \rho_2, \dots, \rho_m) \in S_m^{\mathfrak{N}}$ if for each $i \leq m$ and $\pi \in \mathrm{RA}_{l_i}(\rho_i)$
$$\begin{aligned} &\mathrm{len}(\pi) \leq \theta(i, z, l^i, n_1^i, n_2^i), \\ &\mathrm{hgt}(\pi) \leq \zeta(i, z, l^i, n_1^i, n_2^i), \text{ and} \\ &|\pi| \leq \eta(i, z, l^i, n_1^i, n_2^i). \end{aligned}$$
(4) Due to Proposition 6.5, Duplicator has an S-preserving strategy in the q-round game on $\mathfrak{N}$ and $\mathfrak{N}$. Thus, applying the algorithm SModelCheck (cf. Algorithm 1 in Section 2.1) decides whether $\mathfrak{N} \vDash \varphi$.

7.1. **Complexity of the Algorithm.**

We cannot derive any elementary bound for SModelCheck on 2-NPT. The underlying reason is that the function $\mathrm{LL}_z^{\mathcal{N}}$ bounding the size of the shortest z loops of each stack depends nonuniformly on the 2-PS $\mathcal{N}$. For a fixed 2-PS $\mathcal{N}$ we can compute this dependence, but we have no general bound on the result in terms of $|\mathcal{N}|$. To our knowledge, the only approach for computing such bounds would be Hayashi's pumping lemma for indexed grammars [11]. But his lemma only gives nonelementary bounds on the values of $\mathrm{LL}_z^{\mathcal{N}}$ in terms of $|\mathcal{N}|$ and z.

If we restrict our attention to the case of 1-NPT the picture changes notably. The FO model checking problem on 1-NPT can be solved by an 2-EXPTIME alternating Turing machine. We already proved this bound in [12] using a different approach. In the final part of this section, we sketch how this result follows from the approach in this paper. Recall the characterisation of $\curvearrowright$ in the 1-NPT case from Remark 5.2. We have $\rho \curvearrowright \rho \circ \pi$ if π performs a push transition followed by level 1-loop (i.e., a run that starts and ends in the same stack and never inspects this stack). Furthermore, for ρ_n the minimal element of $\mathrm{RA}_n(\rho)$ we have $\rho_n = \mathrm{pop}_1^n(\rho)$. Furthermore, successive elements of $\mathrm{RA}_n(\rho)$ are connected by a single edge or by $\curvearrowright$. Thus, the final stacks of two successive relevant ancestors differ in at most one letter. The following lemma tells us that the number of $\curvearrowright$ edges starting at some element ρ only depends on the state of ρ and the symbol on top of the stack.

Lemma 7.1. *Let $\mathcal{N}$ be a 1-PS. Let $q, \hat{q} \in Q$, $w, w' \in \Sigma^*$ and $a \in \Sigma$. Then there is a bijection between the runs from (q, wa) to $(\hat{q}, wa)$ that never visit w and the runs from $(q, w'a)$ to $(\hat{q}, w'a)$ that never visit w'.*

Proof. The bijection is given by the stack replacement $[w/w']$ (cf. Lemma 4.8). □

Using the previous observation, it is straightforward to prove the following lemma.

Lemma 7.2. *Let $\mathcal{N}$ be a 1-PS and let $q, \hat{q} \in Q$, $w \in \Sigma^*$ and $a \in \Sigma$. The set*

$$\{\rho : \rho(0) = (q, wa), \rho(\text{len}(\rho))(q', wa), \text{ and } \rho(i) \neq w \text{ for all } 0 \leq i \leq \text{len}(\rho)\}$$

is a context-free language accepted by some 1-PS of size linear in $|\mathcal{N}|$. Furthermore, the set of runs from the initial configuration to (q, w) of $\mathcal{N}$ forms a context-free language that is accepted by some 1-PS of size linear in $|\mathcal{N}|$.

Using the pumping lemma for context free-languages [2], we derive the following bound on short elements of context-free languages.

Lemma 7.3. *There is a fixed polynomial p such that the following holds. Let L be some context-free language that is accepted by a 1-PS $\mathcal{N}$. If L contains k elements, then there are pairwise distinct words $w_1, w_2, \dots, w_k \in L$ such that length $|w_i|$ is bounded by $k \cdot \exp(p(|N|))$ for all $1 \leq i \leq k$.*

These observations imply that it is rather easy to construct runs with similar relevant ancestors in a 1-NPT. In the following, we define a simpler notion of equivalent relevant ancestors in the 1-NPT case that replaces the one for 2-NPT in Definition 6.1.

Definition 7.4. Let $\mathcal{N}$ be some 1-PS generating $\mathfrak{N} := \text{NPT}(\mathcal{N})$. Let $\bar{\rho} := \rho_1, \rho_2, \dots, \rho_n$ and $\bar{\rho}' := \rho'_1, \rho'_2, \dots, \rho'_n$ be elements of $\mathfrak{N}$. We define $\bar{\rho} \approx_l \bar{\rho}'$ if there is a bijection $\varphi : \text{RA}_l(\bar{\rho}) \to \text{RA}_l(\bar{\rho}')$ such that the following holds:

1. For all $l' \leq l$ and $\pi \in \text{RA}_{l'}(\rho_i)$, we have $\varphi(\pi) \in \text{RA}_{l'}(\rho'_i)$.
2. φ preserves $\vdash^\delta$-, $\curvearrowright$- and $\curvearrowright$-edges.
3. φ preserves states and topmost stack elements, i.e., for $\pi \in \text{RA}_l(\bar{\rho})$ such that π ends in configuration (q, wa), then $\varphi(\pi)$ ends in $(q, w'a)$ for some word w'.

Using corresponding constructions as in the proof of proposition 6.5, we can prove that Duplicator has a winning strategy that only uses elements of doubly exponential size.

Proposition 7.5. *There is a polynomial p such that the following holds: Let $\mathcal{N}$ be a 1-PS generating the NPT $\mathfrak{N} := \text{NPT}(\mathcal{N})$. Let $l', C \in \mathbb{N}$, and $l := 4l' + 5$. Furthermore, let $\bar{\rho}$ and $\bar{\rho}'$ be n-tuples of runs of $\mathfrak{N}$ such that $\bar{\rho} \approx_l \bar{\rho}'$, and $\text{len}(\pi) \leq C$ for all $\pi \in \text{RA}_l(\bar{\rho}')$,*

For each $\rho \in \mathfrak{N}$ there is some $\rho' \in \mathfrak{N}$ such that

$$\bar{\rho}, \rho \approx_{l'} \bar{\rho}', \rho' \text{ and } \text{len}(\pi) \leq C + 4^{l'} \cdot n \cdot 4^l \exp(p(|N|)) \text{ for all } \pi \in \text{RA}_{l'}(\bar{\rho}', \rho').$$

Proof. As in the 2-NPT case, we distinguish a local and a global case. Let φ witness that $\bar{\rho} \approx_l \bar{\rho}'$.

- If $\text{RA}_{l'}(\rho) \cap \text{RA}_l(\bar{\rho}) \neq \emptyset$ let ρ_0 be maximal in this set. We define $\rho'_0 := \varphi(\rho_0)$. There are runs $\rho_0 \prec \rho_1 \prec \rho_2 \prec \dots \prec \rho_m$ which form the set $\text{RA}_{l'}(\rho) \cap \{\pi : \rho_0 \preceq \pi\}$. We now inductively construct $\rho'_0 \prec \rho'_1 \prec \rho'_2 \prec \dots \prec \rho'_m$ such that $\rho' := \rho'_m$ satisfies the claim.

Therefore, we copy the edge connecting ρ_i with ρ_{i+1}. If $\rho_i \curvearrowright \rho_{i+1}$ we can construct a run $\rho'_{i+1} \notin \mathrm{RA}_l(\bar\rho)$ such that $\rho'_i \curvearrowright \rho_{i+1}$. Using Lemmas 7.2 and 7.3 we can ensure that $\rho'_{i+1} = \rho'_i \circ \pi$ with $\mathrm{len}(\pi)$ bounded by $n \cdot 4^l \cdot \exp(p(|\mathcal{N}|))$. Since $m \leq 4^{l'}$ the claim follows by iterative use of this observation.

- If $\mathrm{RA}_{l'}(\rho) \cap \mathrm{RA}_l(\bar\rho) = \emptyset$, we proceed analogous to Lemma 6.10. Let ρ_0 be the minimal element of $\mathrm{RA}_{l'}(\rho)$. Let q be its final state and a be its final topmost stack entry. Let m be the number of occurrences of elements in $\mathrm{RA}_l(\bar\rho)$ that end in state q and with final topmost symbol a. Due to $\bar\rho \approx_l \bar\rho'$, m is also the number of elements in $\mathrm{RA}_l(\bar\rho')$ that end in state q and with final topmost symbol a. Due to Lemmas 7.2 and 7.3 there are $m+1$ elements of size at most $n \cdot 4^l \cdot \exp(p(|\mathcal{N}|))$ that end in state q and end with topmost stack entry a. By pigeonhole principle, we can set ρ'_0 to be one of these such that $\rho'_0 \notin \mathrm{RA}_l(\bar\rho')$. Now, we proceed as in the local case. Let $\rho_0 \prec \rho_1 \prec \rho_2 \prec \dots \prec \rho_k$ be the enumeration of $\mathrm{RA}_{l'}(\rho)$. We define runs $\rho'_0 \prec \rho'_1 \prec \rho'_2 \prec \dots \prec \rho'_k$ such that ρ_i is connected to ρ_{i+1} via the same edge as ρ'_i to ρ'_{i+1}. Due to Lemmas 7.2 and 7.3, ρ_{i+1} can be chosen such that the run connecting ρ'_i with ρ'_{i+1} is bounded by $n \cdot 4^l \cdot \exp(p(|\mathcal{N}|))$. Since $k < 4^{l'}$ we conclude that $\rho' := \rho'_k$ has length at most $4^{l'} \cdot n \cdot 4^l \cdot \exp(p(|\mathcal{N}|))$.

□

Corollary 7.6. FO *model checking on* 1*-NPT can be solved by an alternating Turing machine in* 2*-EXPTIME.*

Proof. Let $\varphi \in \mathrm{FO}_r$ and $\mathcal{N}$ some 1-PS. Iterated use of the previous lemmas shows that Duplicator has a winning strategy in the r round game on $\mathrm{NPT}((N))$ and $\mathrm{NPT}((N))$ where he chooses elements of size bounded by $r \cdot \exp(\exp(q(r))) \cdot \exp(p(|\mathcal{N}|))$ for some fixed polynomials p and q. □

8. Conclusion

In this paper we extended the notion of a nested pushdown tree and developed the hierarchy of higher-order nested pushdown trees. These are higher-order pushdown trees enriched by a jump relation that makes corresponding push and pop operations (of the highest level) visible. This new hierarchy is an intermediate step between the hierarchy of pushdown trees and the hierarchy of collapsible pushdown graphs in the sense that it contains expansions of higher-order pushdown trees and is uniformly first-order interpretable in the class of collapsible pushdown graphs. We hope that further study of this hierarchy helps to clarify the relationship between the hierarchies defined by higher-order pushdown systems and by collapsible pushdown systems. We have shown the decidability of the first-order model checking on the first two levels of the nested pushdown tree hierarchy. The algorithm is obtained from the analysis of restricted strategies in Ehrenfeucht-Fraïssé games on nested pushdown trees of level 2. The game analysis gets tractable due to the theory of *relevant ancestors* in combination with shrinking constructions for level 2 pushdown systems. It is open whether this approach extends to higher-levels. The theory of relevant ancestors generalises to all levels of the hierarchy and provides an understanding of the Ehrenfeucht-Fraïssé games. Unfortunately, we do not have any kind of pumping or shrinking constructions for pushdown systems of level 3 or higher.[5] The development of such shrinking construction

[5]See [19] for a counter-example to the wrong pumping lemma in [3].

may yield the necessary bridge between the theory of relevant ancestors of n-NPT and the dynamic-small-witness property needed for the development of a model checking algorithm. Furthermore, better shrinking construction may imply elementary complexity bounds for the FO model checking on 2-NPT. Another open question concerns the modal μ-calculus model checking on this new hierarchy. Note that the interpretation of n-NPT in collapsible pushdown graphs of level $n+1$ is almost modal but for the reversal of the jump edges in comparison to the collapse edges used for their simulation. We conjecture that modal μ-calculus is decidable on the whole hierarchy of nested pushdown trees.

References

[1] R. Alur, S. Chaudhuri, and P. Madhusudan. Languages of nested trees. In *Proc. 18th International Conference on Computer-Aided Verification*, volume 4144 of *LNCS*, pages 329–342. Springer, 2006.

[2] Y. Bar-Hillel, M. Perles, and E. Shamir. On formal properties of simple phrase-structure grammars. *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung*, 14(2):143–172, 1961.

[3] A. Blumensath. On the structure of graphs in the Caucal hierarchy. *Theoretical Computer Science*, 400:19–45, 2008.

[4] C. H. Broadbent. Private communication. September 2010.

[5] A. Carayol. Regular sets of higher-order pushdown stacks. In *MFCS 05*, pages 168–179, 2005.

[6] A. Carayol and S. Wöhrle. The Caucal hierarchy of infinite graphs in terms of logic and higher-order pushdown automata. In *Proceedings of the 23rd Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2003*, volume 2914 of *LNCS*, pages 112–123. Springer, 2003.

[7] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fund. Math.*, 49:129–141, 1960/1961.

[8] J. Ferrante and C.W. Rackoff. *The Computational Complexity of Logical Theories*. Springer-Verlag, Berlin, 1979.

[9] R. Fraïssé. Sur quelques classifications des systèmes de relations. *Publications Scientifiques de l'Université d'Alger, série A*, 1:35–182, 1954.

[10] M. Hague, A. S. Murawski, C-H. L. Ong, and O. Serre. Collapsible pushdown automata and recursion schemes. In *LICS '08: Proceedings of the 2008 23rd Annual IEEE Symposium on Logic in Computer Science*, pages 452–461, 2008.

[11] Takeshi Hayashi. On derivation trees of indexed grammars. *Publ. RIMS, Kyoto Univ.*, 9:61–92, 1973.

[12] A. Kartzow. FO model checking on nested pushdown trees. In *MFCS 09*, volume 5734 of *LNCS*, pages 451–463. Springer, 2009.

[13] A. Kartzow. Collapsible pushdown graphs of level 2 are tree-automatic. In *STACS 10*, volume 5 of *LIPIcs*, pages 501–512. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2010.

[14] A. Kartzow. *First-Order Model Checking On Generalisations of Pushdown Graphs*. PhD thesis, Technische Universität Darmstadt, 2011. Unpublished, submitted in December 2010.

[15] T. Knapik, D. Niwinski, and P. Urzyczyn. Higher-order pushdown trees are easy. In *FOSSACS'02*, volume 2303 of *LNCS*, pages 205–222. Springer, 2002.

[16] A. N. Maslov. The hierarchy of indexed languages of an arbitrary level. *Sov. Math., Dokl.*, 15:1170–1174, 1974.

[17] A. N. Maslov. Multilevel stack automata. *Problems of Information Transmission*, 12:38–43, 1976.

[18] D. E. Muller and P. E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theor. Comput. Sci.*, 37:51–75, 1985.

[19] Pawel Parys. The pumping lemma is incorrect? unpublished, June 2010.

[20] Pawel Parys. Collapse Operation Increases Expressive Power of Deterministic Higher Order Pushdown Automata. In Thomas Schwentick and Christoph Dürr, editors, *28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011)*, volume 9 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 603–614, Dagstuhl, Germany, 2011. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.