Bisimulation Equivalence for First-Order Grammars

Petr Jančar
Techn. Univ. Ostrava, Czech Republic; url: http://www.cs.vsb.cz/jancar/

14 Feb 2012

***Abstract*—A self-contained proof of the decidability of bisimulation equivalence for first-order grammars is given. This provides an alternative for Sénizergues' decidability proof (1998, 2005) for nondeterministic pushdown automata with deterministic popping ε-steps, which generalized his decidability proof for the famous problem of language equivalence of deterministic pushdown automata (1997, 2001). One crucial novelty of the proof presented here is the framework of first-order terms. This framework seems to be more natural for the problem, allowing a presentation which should be transparent for general computer science audience. Though it seems that both the original proof and the proof presented here use the same ideas on an abstract level, the presented (substantially shorter) proof has not arisen as a translation of the original proof, and a detailed comparison would require an enormous technical work.**
For the deterministic (sub)case, a primitive recursive complexity upper bound was derived by Stirling (2002). Here we also suggest a simple way of presenting the ideas behind Stirling's proof in the framework of terms. The complexity result is also analyzed in more detail, which shows that the length of the shortest words witnessing nonequivalence is bounded by $tetr(2, g(n))$ where $tetr$ is the (nonelementary) operator of iterated exponentiation (tetration) and g is an elementary function of the input size.

I. Introduction

Language equivalence of deterministic pushdown automata (dpda), which lie at the heart of syntactic analysis of programming languages, is a famous problem in language theory. The decidability question for this problem was posed in the 1960s [8] (when language inclusion was easily seen as undecidable); then a series of works solving various subcases followed, until the question was finally answered positively by Sénizergues in 1997, with the full journal version [15]. G. Sénizergues was awarded Gödel prize in 2002 for this significant achievement.

Later Stirling [19] and subsequently also Sénizergues [16] provided technically simpler proofs than the original proof. A modified version, which showed a primitive recursive complexity upper bound, appeared as a conference paper by Stirling in 2002 [20]. Sénizergues also generalised the decidability result to bisimulation equivalence over the class of (nondeterministic) pushdown automata with only deterministic and popping ε-transitions [17]. Recall that bisimulation equivalence has been recognized as a fundamental behavioural equivalence in theory of concurrency and communication [13].

Both Sénizergues and Stirling followed their predecessors in presenting the topic in the framework of strict deterministic grammars, but even the above mentioned simplified proofs look rather long and technical; they do not seem to be broadly understood in the computer science community. It seems hard to get further insight, e.g. for trying to derive more results about the complexity (there is no known nontrivial lower bound for dpda-equivalence while the general bisimilarity problem is known to be exptime-hard [11]).

In this paper we choose the framework of first-order grammars, i.e. systems of (classical) first-order terms with finitely many root-rewriting rules, and reprove the above mentioned decidability results, adding a more detailed complexity analysis. It is not surprising that proofs about (d)pda can be done in this framework, since close relations between the frameworks of (d)pda, strict deterministic grammars and first-order schemes were recognized long ago, (cf., e.g., references in [6]) but the proofs here have not arisen as translations of the previous proofs, though they are surely inspired by them. More comments are added in the concluding section.

The related research is rich and active. There are comprehensive references in Sénizergues' and Stirling's papers to the prior research; for recent research on related complexity questions and on higher-order schemes, we can refer, e.g., to [3], [4], [7], [10], [12], [14] and the references therein.

To facilitate readability, the author's intention has been to use figures, examples and explanations of the ideas rather than excessive formalities, without compromising the rigour. The structure of the paper is almost clear from the subsection titles; we just add a few remarks. Section II sets "the stage". It recalls the notion of first-order terms and substitutions, considering *regular terms* (possibly infinite terms with only finitely many different subterms). The terms are then viewed as states of labelled transition systems, where transitions are generated by a first-order grammar, i.e. by a finite set of term root-rewriting rules. Dpda language equivalence naturally reduces to trace equivalence (i.e. a variant of language equivalence) for deterministic first-order grammars, and the semi-decidability of the negative case is immediate. Section III explains an algorithm (semi)deciding the positive case, based on a Prover-Refuter game for which soundness is clear from simple properties of trace equivalence $\sim$ and its "strata" $\sim_0$, $\sim_1$, ... Section IV explains a strategy of Prover which is shown to be complete and thus guarantees the termination of the algorithm. By a more closer look at the strategy, we derive an upper complexity bound in Section V. Section VI then explains the modifications needed to extend the decidability to bisimulation equivalence for general (nondetermininistic) first-order grammars; it is a generalization since bisimulation equivalence coincides with trace equivalence in deterministic labelled transition systems. (The above complexity argument does not apply in this general case.)

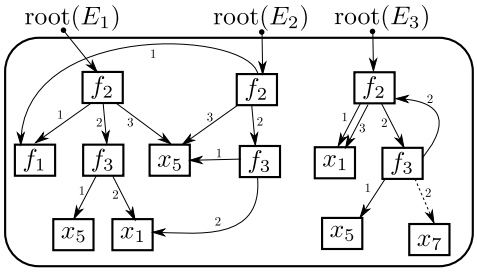

Fig. 1. Graph presentations of terms

II. Basic notions and simple observations

$\mathbb{N}$ denotes the set $\{0, 1, 2, \dots\}$ of natural numbers. For a set $\mathcal{A}$, by $\text{CARD}(\mathcal{A})$ we denote its cardinality (i.e. the number of elements when $\mathcal{A}$ is finite). $\mathcal{A}^*$ denotes the set of finite sequences of elements of $\mathcal{A}$, also called *words* (over $\mathcal{A}$). By $|w|$ we denote the *length* of $w \in \mathcal{A}^*$. If $w = uv$ then u is a *prefix* of w. The *empty sequence* is denoted ε (thus $|\varepsilon| = 0$).

First-order regular terms; substitutions

We recall the standard notion of *first-order terms* by examples, assuming a fixed countable set $\text{VAR} = \{x_1, x_2, x_3, \dots\}$ of *variables*. Given a set $\mathcal{F} = \{f_1, f_2, \dots, f_k\}$ of *function symbols* where each $f \in \mathcal{F}$ has $arity(f) \in \mathbb{N}$, an example of a *term over* $\mathcal{F}$ is $E_1 = f_2(f_1, f_3(x_5, x_1), x_5)$, when $k \geq 3$ and $arity(f_1) = 0$, $arity(f_2) = 3$, $arity(f_3) = 2$. We can recognize the syntactic tree of E_1 in Fig.1.

The figure suggests to define a *graph presentation* GP (of some terms) as a *finite graph* whose nodes are labelled with elements of $\mathcal{F} \cup \text{VAR}$. Each node labelled with f has m outgoing arcs labelled with $1, 2, \dots, m$ where $m = arity(f)$ (now ignore the dotted arc leading to x_7); the nodes labelled with variables x_i have no outgoing arcs (as well as the nodes labelled with nullary function symbols). A term E is presented by a graph GP when a node n is specified as the *root* of E; we refer to such a graph as to GP_E. A term can have more than one presentation; e.g., E_1, E_2 represented in Fig.1 are obviously the same terms.

Formally we view *terms as* the *partial mappings*

$$E : \mathbb{N}^* \to \mathcal{F} \cup \text{VAR}$$

where $\gamma\, i$ ($\gamma \in \mathbb{N}^*$, $i \in \mathbb{N}$) belongs to $\text{DOM}(E)$ iff $\gamma \in \text{DOM}(E)$ and $1 \leq i \leq arity(E(\gamma))$; we stipulate $arity(x_j) = 0$ for all $x_j \in \text{VAR}$. The expressions like $f_2(f_1, f_3(x_5, x_1), x_5)$ or x_{17} are thus viewed as representing partial mappings $\mathbb{N}^* \to \mathcal{F} \cup \text{VAR}$ (e.g., $E_1(\varepsilon) = f_2$, $E_1(\langle 2, 1\rangle) = x_5$).

Given a term E and $\gamma \in \text{DOM}(E)$, *the term* E_γ where $\text{DOM}(E_\gamma) = \{\delta \mid \gamma\delta \in \text{DOM}(E)\}$ and $E_\gamma(\delta) = E(\gamma\delta)$ is a *subterm* of E, *occurring in* E *at* γ, *in depth* $|\gamma|$.

A *term* E is *finite* if $\text{DOM}(E)$ is finite. A *term* is *regular* if it has only finitely many subterms; in other words, it has a finite graph presentation (possibly with cycles). E_3 in Fig.1 is an infinite regular term (e.g., $E_3(\langle 2, 2, 2, 2, 2, 2, 1\rangle) = x_1$).

Convention. By "terms" we further mean "regular terms". We do not consider the empty term with the empty domain; we might use a special nullary (function) symbol $\perp$ instead.

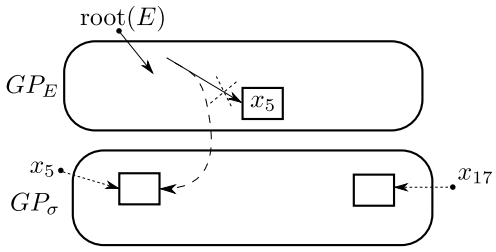

Fig. 2. Creating $\text{GP}_{E\sigma}$ from GP_E and GP_σ

We note that the equality of two (regular) terms can be efficiently checked, given their graph presentations: Given GP, for each node n of GP we consider the (sub)term rooted in n. We first partition all these terms according to the root-labels, and then we are refining the partition according to the (current) partition-classes of root-successors, until the stable partition (the fixpoint) is found.

By $\text{TERMS}_\mathcal{F}$ we denote the set of all (regular) terms over $\mathcal{F}$. A *substitution* σ is a mapping $\sigma : \text{VAR} \to \text{TERMS}_\mathcal{F}$ whose *support* $\text{SUPP}(\sigma) = \{x_i \mid \sigma(x_i) \neq x_i\}$ is *finite*. Our finite-support restriction allows to present any substitution σ by a finite graph GP where each $x_i \in \text{SUPP}(\sigma)$ has an associated node in GP, namely the root of $\sigma(x_i)$; we refer to GP_σ then. For a term E and a substitution σ, we define $E\sigma$ (the *term resulting from* E *by applying* σ) as expected: $\gamma \in \text{DOM}(E\sigma)$ iff either $\gamma \in \text{DOM}(E)$ and $E(\gamma) \notin \text{VAR}$, in which case $(E\sigma)(\gamma) = E(\gamma)$, or $\gamma = \gamma_1\gamma_2$, $\gamma_1 \in \text{DOM}(E)$, $E(\gamma_1) = x_j$, $\gamma_2 \in \text{DOM}(\sigma(x_j))$, in which case $(E\sigma)(\gamma) = (\sigma(x_j))(\gamma_2)$.

Fig.2 illustrates how to get a presentation of $E\sigma$ from presentations GP_E, GP_σ of E and σ, respectively: each arc leading to (a node labelled with) $x_i \in \text{SUPP}(\sigma)$ in GP_E is redirected to the node associated with x_i in GP_σ. Note that if $E = x_i$ then $E\sigma = x_i\sigma = \sigma(x_i)$.

By $E\sigma_1\sigma_2$ we mean $(E\sigma_1)\sigma_2$, but we also define the *composition of substitutions* $\sigma_1 \circ \sigma_2$, denoted just $\sigma_1\sigma_2$: for $\sigma = \sigma_1\sigma_2$ and $x_i \in \text{VAR}$ we have $\sigma(x_i) = (\sigma_1(x_i))\,\sigma_2$; thus $\text{SUPP}(\sigma_1\sigma_2) \subseteq \text{SUPP}(\sigma_1) \cup \text{SUPP}(\sigma_2)$. We can easily check $E\sigma_1\sigma_2 = (E\sigma_1)\sigma_2 = E\sigma$ where $\sigma = \sigma_1\sigma_2$; more generally, the *composition* is *associative*, i.e., $(\sigma_1\sigma_2)\sigma_3 = \sigma_1(\sigma_2\sigma_3)$.

Finally we note that E_3 in Fig.1 can be viewed as arising from (a finite term) $E = f_2(x_1, f_3(x_5, x_7), x_1)$ (represented like E_3 but using *the dotted arc* in Fig.1) by applying the substitution $\sigma' = \{(x_7, E)\}$, i.e. σ' where $\text{SUPP}(\sigma') = \{x_7\}$ and $\sigma'(x_7) = E$, repeatedly forever; hence $E_3 = E\sigma'\sigma'\sigma' \dots$. (To get GP_{E_3} from GP_E, each arc leading to x_7 is redirected to the root.) Note that the auxiliary variable x_7 could be replaced with any x_i not occuring in E_3.

Later we also refer to the *presentation size* $\text{PRESSIZE}(E)$, by which we mean the *size* of the *smallest graph presentation* of E; similarly for $\text{PRESSIZE}(\sigma)$. We can use any natural notion of *size* which takes also the indices of variables into account; e.g., we can take the number of nodes and arcs plus the bit-size of all labels. We thus have only finitely many terms E with $\text{PRESSIZE}(E) \leq b$, for any given bound $b \in \mathbb{N}$. Another natural property which we assume is $\text{PRESSIZE}(E\sigma) \leq \text{PRESSIZE}(E) + \text{PRESSIZE}(\sigma)$.

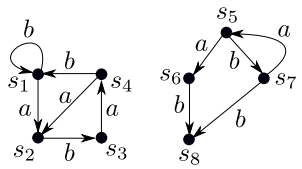

Fig. 3. A (finite) deterministic labelled transition system

(Deterministic) labelled transition systems; trace equivalence

A *labelled transition system*, an *LTS* for short, is a tuple $\mathcal{L} = (\mathcal{S}, \mathcal{A}, (\xrightarrow{a})_{a\in\mathcal{A}})$ where $\mathcal{S}$ is the set of *states*, $\mathcal{A}$ the set of *actions* and $\xrightarrow{a} \subseteq \mathcal{S}\times\mathcal{S}$ is the set of *transitions labelled with* $a \in \mathcal{A}$, called *a-transitions*. Fig.3 shows a finite LTS (in fact, a det-LTS as defined later). The relations $\xrightarrow{w} \subseteq \mathcal{S}\times\mathcal{S}$ for $w \in \mathcal{A}^*$ are defined as expected: $s \xrightarrow{\varepsilon} s$; if $s \xrightarrow{a} s'$ and $s' \xrightarrow{u} s''$ then $s \xrightarrow{au} s''$. In Fig. 3 we have, e.g., $s_1 \xrightarrow{bab} s_3$.

By writing $s \xrightarrow{w}$ we mean that s *enables* (a trace) $w \in \mathcal{A}^*$, i.e., $s \xrightarrow{w} s'$ for some s'. *Trace equivalence* $\sim$ on $\mathcal{S}$ and its "strata" $\sim_0$, $\sim_1$, $\sim_2$, ... are defined as follows:

$$s \sim t \text{ if } \forall w \in \mathcal{A}^* : s \xrightarrow{w} \Leftrightarrow t \xrightarrow{w},$$
$$\text{and for } k \in \mathbb{N}\text{: } s \sim_k t \text{ if } \forall w \in \mathcal{A}^{\leq k} : s \xrightarrow{w} \Leftrightarrow t \xrightarrow{w},$$

where $\mathcal{A}^{\leq k} = \{w \in \mathcal{A}^* \mid |w| \leq k\}$.

Proposition 1. *(1) $\sim$ and all $\sim_k$ are equivalence relations. (2) $\sim_0 = \mathcal{S}\times\mathcal{S}$. (3) $\sim_0 \supseteq \sim_1 \supseteq \sim_2 \supseteq \cdots$. (4) $\cap_{k\in\mathbb{N}} \sim_k = \sim$.*

This (trivial) proposition suggests to define the *equivalence-level* (*eq-level*) for each pair of states:

$$\text{EQLV}(s,t) = k \ (k \in \mathbb{N}) \text{ if } s \sim_k t \text{ and } s \not\sim_{k+1} t;$$
$$\text{EQLV}(s,t) = \omega \text{ if } s \sim t, \text{ also written as } s \sim_\omega t.$$

In Fig.3 we have, e.g., $\text{EQLV}(s_1, s_2) = 0$, $\text{EQLV}(s_1, s_5) = 2$, $\text{EQLV}(s_1, s_4) = \omega$. We take ω as an infinite number satisfying $n < \omega$ and $\omega - n = \omega + n = \omega$ for any $n \in \mathbb{N}$.

The next fact (for $k \in \mathbb{N}$) will be particularly useful.

Proposition 2. *If $\text{EQLV}(s,t) = k$ and $\text{EQLV}(s,s') \geq k+1$ then $\text{EQLV}(s',t) = k$ (since $s' \sim_k s \sim_k t$ and $s' \sim_{k+1} s \not\sim_{k+1} t$).*

Of special interest for us are *deterministic* LTSs, *det-LTSs* for short: $\mathcal{L} = (\mathcal{S}, \mathcal{A}, (\xrightarrow{a})_{a\in\mathcal{A}})$ is deterministic if for each $s \in \mathcal{S}$ and each $a \in \mathcal{A}$ there is at most one s' such that $s \xrightarrow{a} s'$. (Fig.3 depicts a finite det-LTS.)

Proposition 3. *In any det-LTS, $s \xrightarrow{w}$ entails a unique* path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots \xrightarrow{a_k} s_k$ *where $w = a_1 a_2 \ldots a_k$.*

For det-LTSs we easily observe that by performing the same action $a \in \mathcal{A}$ from s, t the eq-level can drop by at most one, and it does drop for some action when $\omega > \text{EQLV}(s,t) > 0$:

Proposition 4. *Given a* deterministic *LTS:*
(1) If $s \xrightarrow{w} s'$, $t \xrightarrow{w} t'$ then $\text{EQLV}(s',t') \geq \text{EQLV}(s,t) - |w|$.
(2) If $\text{EQLV}(s,t) = k \in \mathbb{N}$ then there is $w = a_1 a_2 \ldots a_k$ such that $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \cdots \xrightarrow{a_k} s_k$ and $t \xrightarrow{a_1} t_1 \xrightarrow{a_2} t_2 \xrightarrow{a_3} \cdots \xrightarrow{a_k} t_k$ where $\text{EQLV}(s_j, t_j) = k - j$ for $j = 1, 2, \ldots, k$.

In Point (2) we have $s_k \not\sim_1 t_k$, hence there is $a \in \mathcal{A}$ such that $s_k \xrightarrow{a}$, $\neg(t_k \xrightarrow{a})$ or vice versa; the word wa is then a *shortest nonequivalence-witness word* for the pair s, t.

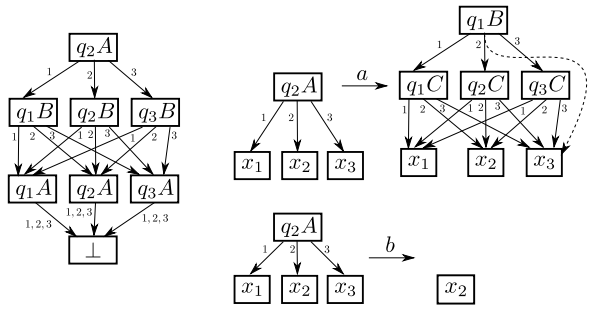

Fig. 4. Term representations of configurations and rules of (d)pda

(D)pda from a first-order term perspective

In the next subsection we formally introduce LTSs whose *states* are not "black dots" as in Fig.3 but *(regular) terms* E. Transitions $E_1 \xrightarrow{a} E_2$ will be determined by a finite set of (term root-rewriting) rules, i.e. by a (deterministic) first-order grammar. We will be interested in the *trace equivalence problem* asking if $E_1 \sim E_2$. Before precise definitions, we give a flavour of reasoning which shows that considering the classical *dpda language equivalence problem* from a "first-order term perspective" naturally leads to the notion of first-order grammars and *easily reduces to the above trace equivalence problem*.

A *pushdown automaton* (*pda*) can be viewed as a tuple $\mathcal{M} = (Q, \mathcal{A}, \Gamma, \Delta)$ of finite sets of *control states*, *actions* (also called *input letters*), *stack-symbols*, and (rewriting) *rules*, respectively. On the left in Fig.4 we can see a *term-representation of* the *configuration* $q_2ABA \in Q \times \Gamma^*$, assuming $Q = \{q_1, q_2, q_3\}$. So $Q \times \Gamma$ is the set of "function symbols", called now *nonterminals*, all with arity $\text{CARD}(Q)$. $\bot$ can be here viewed as a special "stack-bottom" nullary nonterminal. On the right we can see (the term representations of) two *rules*, one *pushing* ($q_2A \xrightarrow{a} q_1BC$) and one *popping* ($q_2A \xrightarrow{b} q_2$). (Ignore the dotted arc now.) For configurations $\mathcal{C}_1, \mathcal{C}_2$ (viewed as terms) we have $\mathcal{C}_1 \xrightarrow{a} \mathcal{C}_2$ iff there is a rule $E_1 \xrightarrow{a} E_2$ (in Δ) and a substitution σ such that $\mathcal{C}_1 = E_1\sigma$ and $\mathcal{C}_2 = E_2\sigma$. (We can check by term-representations that $q_2ABA \xrightarrow{a} q_1BCBA$, $q_2ABA \xrightarrow{b} q_2BA$.)

For a *deterministic pushdown automaton* (*dpda*) we thus get a det-LTS (with configurations as states) if we can get rid of ε-steps, i.e. of the rules like $q_2C \xrightarrow{\varepsilon} q_3$, written $[q_2C](x_1, x_2, x_3) \xrightarrow{\varepsilon} x_3$ in our term-representation. Note that a rule $[q_2C](x_1, x_2, x_3) \xrightarrow{\varepsilon}$ .. excludes the existence of another rule $[q_2C](x_1, x_2, x_3) \longrightarrow$ .., and it is standard to assume (i.e. safely transform the dpda so) that the *ε-rules* are *only popping*. The dotted arc in Fig.4 illustrates that we can get rid of all "unstable" nonterminals like $[q_2C]$, thus "swallowing" all potential ε-transitions in advance.

It is a routine to reduce the *dpda language equivalence problem* to the above mentioned trace equivalence problem, later formalized as TRACE-EQ-DET-G. We can use the following version: given $\mathcal{M}$ and configurations $\mathcal{C}, \mathcal{C}'$, are $\mathcal{C}, \mathcal{C}'$ language equivalent in the empty-stack acceptance sense? We do the above term-transformation of $\mathcal{M}$, also removing all unstable nonterminals like $[q_2C]$. Then for any pair $[qA], a$ ($a \in \mathcal{A}$)

$$r_1 : Ax_1 \xrightarrow{a} ABx_1 \qquad r_2 : Ax_1 \xrightarrow{b} x_1$$
$$r_3 : Bx_1 \xrightarrow{a} BAx_1 \qquad r_4 : Bx_1 \xrightarrow{b} x_1$$

Fig. 5. A det-first-order grammar $\mathcal{G} = (\{A, B\}, \{a, b\}, \{r_1, r_2, r_3, r_4\})$

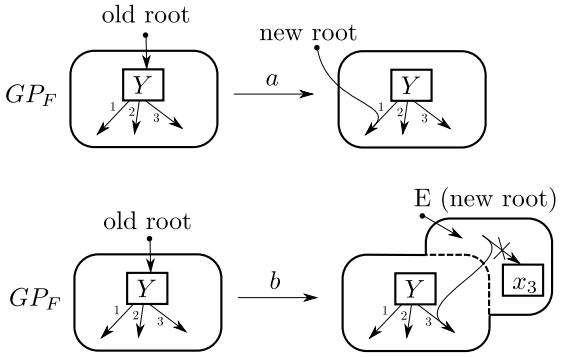

Fig. 6. Applying rules $Yx_1x_2x_3 \xrightarrow{a} x_1$ and $Yx_1x_2x_3 \xrightarrow{b} E$ to GP_F

for which there is no rule $[qA](x_1, \ldots, x_{\text{CARD}(Q)}) \xrightarrow{a} ..$ we add the rule $[qA](x_1, \ldots, x_{\text{CARD}(Q)}) \xrightarrow{a} [Loop]$ where $[Loop]$ is an added nullary nonterminal with rules $[Loop] \xrightarrow{b} [Loop]$ for all $b \in \mathcal{A}$. $\perp$ is unreachable after an added rule is used. (More details are in Appendix.)

(Det-) first-order grammars as generators of (det-)LTSs

Definition 5. *A* first-order grammar *is a tuple $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ where $\mathcal{N}$ is a finite set of ranked* nonterminals*, i.e. (function) symbols with arities, $\mathcal{A}$ is a finite set of* actions *(or terminals), and $\mathcal{R}$ is a finite set of* (root rewriting) rules *r of the form*

$$r : Yx_1x_2 \ldots x_m \xrightarrow{a} E \tag{1}$$

where $Y \in \mathcal{N}$, $arity(Y) = m$, $a \in \mathcal{A}$, and E is a finite *term over $\mathcal{N}$ in which each occurring variable is from the set $\{x_1, x_2, \ldots, x_m\}$. ($E = x_i$, where $1 \leq i \leq m$, is an example.) We put $\text{ACT}(r) = a$, thus defining the mapping $\text{ACT} : \mathcal{R} \to \mathcal{A}$. $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ is* deterministic*, a* det-first-order grammar*, if there is at most one rule (1) for each pair $Y \in \mathcal{N}$, $a \in \mathcal{A}$.*

Remark on notation. In the previous (classical term) notation, the rules would be written $r : f(x_1, x_2, \ldots, x_m) \xrightarrow{a} E$. Now $\mathcal{N}$ plays the role of former $\mathcal{F}$; we use Y to range over $\mathcal{N}$, and we omit parentheses. We might also use A, B for nonterminals, but E, F, G, H and T, U, V, W will always range over $\text{TERMS}_{\mathcal{N}}$ (using our fixed $\text{VAR} = \{x_1, x_2, \ldots\}$). $YG_1G_2 \ldots G_m$, $(Ax_1x_2x_3)\sigma = A\,\sigma(x_1)\sigma(x_2)\sigma(x_3)$, $F'\sigma_1\sigma_2$ are examples of notation which we use for terms (where σ's are substitutions). We consider $\perp$ as a special nullary nonterminal, with no rules; we use it in the example in Fig.9.

Fig.5 shows an example of a det-first-order grammar $\mathcal{G}$. This $\mathcal{G}$ is, in fact, very simple, we have $arity(Y) = 1$ for all $Y \in \mathcal{N}$ and the rules are thus of the form $Yx_1 \xrightarrow{a} Y_1Y_2 \ldots Y_\ell\, x_1$. (A more general example was given in Fig.4.) Our example grammar is thus, in fact, a context-free grammar in Greibach normal form, with no special starting symbol and with only leftmost derivations allowed, as the next definition shows (by using rules as *root-rewriting*). In fact, the definition takes all *(regular) terms as states*, though we allowed only *finite right-hand sides* (rhs) E *in rules* (1) for technical convenience.

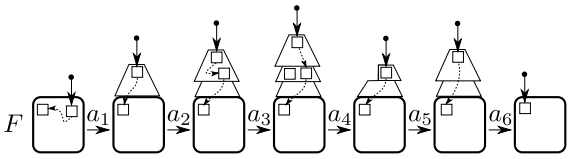

Fig. 7. A path in $\mathcal{L}^{\text{A}}_{\mathcal{G}}$

Definition 6. *A grammar $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ generates (the* rule based*) LTS $\mathcal{L}^{\text{R}}_{\mathcal{G}} = (\text{TERMS}_{\mathcal{N}}, \mathcal{R}, (\xrightarrow{r})_{r \in \mathcal{R}})$: for each rule $r : Yx_1x_2 \ldots x_m \xrightarrow{a} E$ (recall (1)) we have*

$$F \xrightarrow{r} H \text{ if there is a substitution } \sigma \text{ such that } F = (Yx_1 \ldots x_m)\sigma \text{ and } H = E\sigma.$$

(Note that σ with $\text{SUPP}(\sigma) = \emptyset$ yields $Yx_1 \ldots x_m \xrightarrow{r} E$.) For (the action-based*) LTS $\mathcal{L}^{\text{A}}_{\mathcal{G}} = (\text{TERMS}_{\mathcal{N}}, \mathcal{A}', (\xrightarrow{a})_{a \in \mathcal{A}'})$ we define $\mathcal{A}' = \mathcal{A} \cup \{a_{x_i} \mid x_i \in \text{VAR}\}$ where a_{x_i} is a unique (fresh) action attached to x_i. For $a \in \mathcal{A}'$ we have $F \xrightarrow{a} H$ if $F \xrightarrow{r} H$ for some $r \in \mathcal{R}$ with $\text{ACT}(r) = a$ or if $F = H = x_i$ and $a = a_{x_i}$.*

Remark and convention. In $\mathcal{L}^{\text{R}}_{\mathcal{G}}$ the variables x_i are examples of *dead terms* (not enabling any transition), like the term $\perp$. In $\mathcal{L}^{\text{A}}_{\mathcal{G}}$ we have $x_i \xrightarrow{a_{x_i}} x_i$ but we never use these special transitions in our reasoning; we only use the consequence that $x_i \not\sim_1 H$ if $H \neq x_i$ (in particular if $H = x_j$ for $j \neq i$).

Proposition 7. *$\mathcal{L}^{\text{R}}_{\mathcal{G}}$ is a det-LTS for any $\mathcal{G}$.*
$\mathcal{L}^{\text{A}}_{\mathcal{G}}$ is a det-LTS iff $\mathcal{G}$ is deterministic.

Fig.6 shows a way how to apply the rules (of some grammar) to graph presentations. To apply $r : Yx_1x_2x_3 \xrightarrow{b} E$ to GP_F, we first check if the root of F is (labelled with) Y. If yes then we add GP_E (the rhs of r) to GP_F (we "stack GP_E on top of GP_F"), the root of E becomes the new root, and every arc leading to x_i in GP_E is redirected to the i-th successor of the (old) root of F. In the case of a *sink rule* like $Yx_1x_2x_3 \xrightarrow{a} x_j$ ("popping" in pda terminology) we do, in fact, the same; the result is that the j-th successor of the (old) root in GP_F becomes the new root (it can be the old root in case of a loop). Fig.7 depicts a path in $LTS^{\text{A}}_{\mathcal{G}}$. We note that even if we successively "stack" many (finite) rhs $E_1, E_2, \ldots$ of used rules (or rather subterms of rhs), there can be always root-successors lying "deeply down", even in the initial (regular) term F. The figure also highlights that the current root is always connected to any future root which lies in the current graph.

The next fact holds in both $\mathcal{L}^{\text{R}}_{\mathcal{G}}$ and $\mathcal{L}^{\text{A}}_{\mathcal{G}}$. (Recall Fig.2).

Proposition 8. *If $E \xrightarrow{w} F$ then $E\sigma \xrightarrow{w} F\sigma$; hence if $E \xrightarrow{w} x_i$ then $E\sigma \xrightarrow{w} \sigma(x_i)$. If $E\sigma \xrightarrow{w}$ but $\neg(E \xrightarrow{w})$ then $w = uv$ where $E \xrightarrow{u} x_i$ for some $x_i \in \text{VAR}$ and $E\sigma \xrightarrow{u} \sigma(x_i) \xrightarrow{v}$.*

Convention. We further refer to $\mathcal{L}^{\text{A}}_{\mathcal{G}}$, if not said otherwise. Hence by writing $E \xrightarrow{w} F$ we mean $w \in \mathcal{A}^*$.

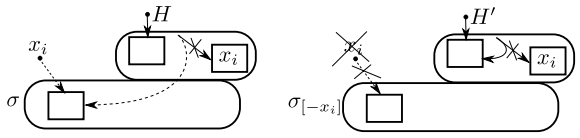

Fig. 8. $H\sigma$ and $H'\sigma_{[-x_i]} = (H\{(x_i, H)\}\{(x_i, H)\}\{(x_i, H)\}\cdots)\sigma$

Semidecidability of trace non-equivalence

Given $\mathcal{G}$ and a pair $E \not\sim F$, we can find a shortest word witnessing non-equivalence of E, F by systematic search. Hence the next lemma is obvious even in the general case, though we now concentrate on the deterministic case.

Problem TRACE-EQ-DET-G:

Input: a det-first-order grammar $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$, and (graph presentations of) two input terms T_{in}, U_{in}.
Question: is $T_{in} \sim U_{in}$ in $\mathcal{L}^{\mathrm{A}}_{\mathcal{G}}$?

Lemma 9. *There is an algorithm with the following property: it (halts and) computes* $\mathrm{EQLV}(T_{in}, U_{in})$ *for an instance* $\mathcal{G}, T_{in}, U_{in}$ *of* TRACE-EQ-DET-G *iff* $T_{in} \not\sim U_{in}$ *in* $\mathcal{L}^{\mathrm{A}}_{\mathcal{G}}$.
Thus the complement of TRACE-EQ-DET-G *is semidecidable.*

For later use we classify nonequivalent pairs E, F into two disjoint cases (recall Proposition 4(2)).

Proposition 10. *Assume a det-first-order grammar* $\mathcal{G}$.
If $\mathrm{EQLV}(E, F) = k \in \mathbb{N}$ *then we have either*
CASE 1: there is w, $|w| = k$, *such that* $E \xrightarrow{w} E'$, $F \xrightarrow{w} F'$ *and the roots of* E', F' *are nonterminals enabling different sets of actions (thus causing* $E'\sigma \not\sim_1 F'\sigma$ *for any* σ*), or*
CASE 2: there is no word required in CASE 1 but there is w, $|w| = k$, *such that* $E \xrightarrow{w} x_i$, $F \xrightarrow{w} H$ *or* $E \xrightarrow{w} H$, $F \xrightarrow{w} x_i$ *where* $H \neq x_i$. *(Recall that* $x_i \not\sim_1 H$ *then.)*

III. AN ALGORITHM DECIDING TRACE-EQ-DET-G

We aim to show the semidecidability of TRACE-EQ-DET-G, which will yield the decidability by Lemma 9. III-A shows some simple facts about the equivalences $\sim_k$ and $\sim$, and III-B introduces further technical prerequisites for the Prover-Refuter game (played for an instance $\mathcal{G}$, T_{in}, U_{in}) described in III-C; we also give a simple example of a play. In III-D we will easily observe the *soundness* of the P-R game, which means that Prover has no winning strategy if $T_{in} \not\sim U_{in}$. It will be also obvious that there is an algorithm which halts for $\mathcal{G}$, T_{in}, U_{in} iff Prover has a finite winning strategy. Hence the decidability of TRACE-EQ-DET-G will be established once we show the *completeness*, i.e. the existence of a finite winning strategy of Prover for every $T_{in} \sim U_{in}$; this is done in Sec.IV.

Convention. If not said otherwise, we assume a given det-first-order grammar $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ and refer to the det-LTS $\mathcal{L}^{\mathrm{A}}_{\mathcal{G}}$. (Recall that Prop.3 applies here.) By referring to a *path* $G \xrightarrow{w}$ (or $G \xrightarrow{w} G'$) we mean that w is enabled by G but we also refer to the unique sequence $G \xrightarrow{a_1} G_1 \xrightarrow{a_2} G_2 \xrightarrow{a_2} \cdots \xrightarrow{a_k} G_k$ ($G_k = G'$) where $w = a_1 a_2 \ldots a_k$.

A. Some properties of $\sim_k$ and $\sim$ (in the det-LTS $\mathcal{L}^{\mathrm{A}}_{\mathcal{G}}$)

For two substitutions $\sigma, \sigma' : \mathrm{VAR} \to \mathrm{TERMS}_{\mathcal{N}}$ we define

$$\sigma \sim_k \sigma' \text{ if } \sigma(x_i) \sim_k \sigma'(x_i) \text{ for all } x_i \in \mathrm{VAR}.$$

The following *congruence properties* are obvious, by recalling Prop.8 (and Fig.2, 6, 7), and noting that if $E \sim_k F$, $E \xrightarrow{u} x_i$, $|u| < k$ then $F \xrightarrow{u} x_i$.

Proposition 11. *(1) If* $E \sim_k F$ *then* $E\sigma \sim_k F\sigma$.
Hence $\mathrm{EQLV}(E, F) \leq \mathrm{EQLV}(E\sigma, F\sigma)$.
(2) If $\sigma \sim_k \sigma'$ *then* $E\sigma \sim_k E\sigma'$.
Hence $\mathrm{EQLV}(\sigma, \sigma') \leq \mathrm{EQLV}(E\sigma, E\sigma')$.

We now look more closely at Point (1).

Proposition 12. *If* $\mathrm{EQLV}(E, F) = k < \ell = \mathrm{EQLV}(E\sigma, F\sigma)$ *(where* $\ell \in \mathbb{N} \cup \{\omega\}$*) then there are some* $x_i \in \mathrm{SUPP}(\sigma)$, $H \neq x_i$, *and a word* w, $|w| = k$, *such that* $E \xrightarrow{w} x_i$, $F \xrightarrow{w} H$ *or* $E \xrightarrow{w} H$, $F \xrightarrow{w} x_i$*; moreover,* $\sigma(x_i) \sim_{\ell-k} H\sigma$.

Proof: If (E, F) belongs to CASE 1 of Proposition 10 then $\mathrm{EQLV}(E\sigma, F\sigma) = \mathrm{EQLV}(E, F) = k$. Hence any (E, F) satisfying the assumption belongs to CASE 2, so let us take x_i, H which are guaranteed there; if $H \notin \mathrm{VAR}$ then $x_i \in \mathrm{SUPP}(\sigma)$, and if $H = x_j$ then at least one of x_i, x_j belongs to $\mathrm{SUPP}(\sigma)$ (otherwise $\mathrm{EQLV}(E\sigma, F\sigma) = k$). The claim $\sigma(x_i) \sim_{\ell-k} H\sigma$ follows from Prop.4(1) (since $E \xrightarrow{w} x_i$, $F \xrightarrow{w} H$ implies $E\sigma \xrightarrow{w} \sigma(x_i)$, $F\sigma \xrightarrow{w} H\sigma$). ■

The next proposition (sketched in Fig.8) is later useful for decreasing the support of a substitution in an inductive argument (in Fig.13). We define $\sigma_{[-x_i]}$ as the substitution arising from σ by *removing* x_i *from the support* (if it is there):

$$\sigma_{[-x_i]}(x_i) = x_i \text{ and } \sigma_{[-x_i]}(x_j) = \sigma(x_j) \text{ for all } j \neq i.$$

Recall $E_3 = E\sigma'\sigma'\sigma' \ldots$ in Fig.1 where $\sigma' = \{(x_7, E)\}$. If $\sigma(x_7) \sim_k E\sigma$ (in some $\mathcal{L}^{\mathrm{A}}_{\mathcal{G}}$), whence $\sigma \sim_k \{(x_7, E)\}\,\sigma = \sigma'\sigma$, then Proposition 11(2) entails $\sigma(x_7) \sim_k E\sigma \sim_k E\sigma'\sigma \sim_k E\sigma'\sigma'\sigma \sim_k (E\sigma'\sigma'\cdots)\sigma = E_3\sigma$. Note that $E_3\sigma = E_3\sigma_{[-x_7]}$ since x_7 does not occur in E_3. We formalize this as follows (see Fig.8).

Proposition 13. *Assume* $H \neq x_i$ *and* $H' = H\sigma'\sigma'\cdots$ *where* $\sigma' = \{(x_i, H)\}$. *(*$\mathrm{GP}_{H'}$ *arises from* GP_H *by redirecting all incoming arcs of* x_i *to the root of* H*; hence* $H' = H$ *if* x_i *does not occur in* H*, in particular if* $H = x_j$, $j \neq i$.*)*
If $\sigma(x_i) \sim_k H\sigma$ *then* $\sigma(x_i) \sim_k H'\sigma_{[-x_i]}$ *and thus* $\sigma \sim_k \{(x_i, H')\}\,\sigma_{[-x_i]}$.

Proof: $H'\sigma = H'\sigma_{[-x_i]}$ since x_i does not occur in H'. If $H\sigma \sim H'\sigma$ then the claim is trivial. Otherwise ($H' \neq H$ and) $\mathrm{EQLV}(H\sigma, H'\sigma) > \mathrm{EQLV}(\sigma(x_i), H'\sigma)$ (since any w witnessing $H\sigma \not\sim H'\sigma$ must have a nonempty prefix u such that $H \xrightarrow{u} x_i$ and $H' \xrightarrow{u} H'$; see Fig.8). Hence $\mathrm{EQLV}(\sigma(x_i), H\sigma) = \mathrm{EQLV}(\sigma(x_i), H'\sigma)$ (by Prop.2). ■

B. k-distance regions (for deciding $T \sim_k U$)

We have implicitly noted (around Lemma 9) that we can decide whether $T \sim_k U$ (for $k \in \mathbb{N}$); a natural way is to construct the *k-distance region for* (T, U):

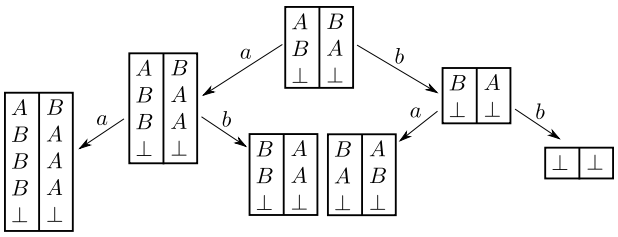

Fig. 9. The 2-distance region $\text{REG}(T, U, 2)$ for $(T, U) = (AB\bot, BA\bot)$

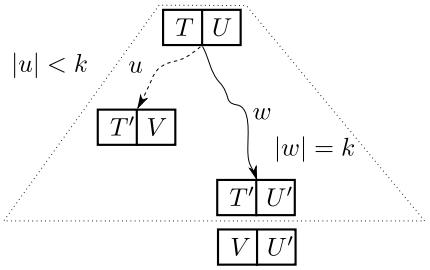

Fig. 10. Case 1 of left-balancing

$$\text{REG}(T,U,k) = \{\, (T',U') \mid T \xrightarrow{w} T', U \xrightarrow{w} U' \text{ for some } w, |w| \le k \,\}.$$

Fig.9 shows the 2-distance region for $(T, U) = (AB\bot, BA\bot)$, assuming our example grammar in Fig.5.

Note that $T \not\sim_k U$ iff there is $(T', U') \in \text{REG}(T, U, k-1)$ such that $T' \not\sim_1 U'$. We define the *least eq-level* for a set of pairs of terms (for a region $\text{REG}(T, U, k)$ in particular):

$$\text{for } R \subseteq \text{TERMS}_{\mathcal{N}} \times \text{TERMS}_{\mathcal{N}},\ R \neq \emptyset, \text{ we define}$$
$$\text{MINEL}(\mathcal{R}) = \min \{\, \text{EQLV}(T',U') \mid (T',U') \in \mathcal{R} \,\}.$$

The next proposition tells us that any least eq-level pair in $\text{REG}(T, U, k)$ must be in the bottom row in the figures like Fig.9 or Fig.10, if $T \not\sim U$. It follows trivially from Prop.4.

Proposition 14.
(1.) If $T \sim U$ then $T' \sim U'$ for all $(T', U') \in \text{REG}(T, U, k)$.
(2.) If $T \not\sim U$, $T \sim_k U$ and $(T', U') \in \text{REG}(T, U, k)$ satisfies $\text{EQLV}(T', U') = \text{MINEL}(\text{REG}(T, U, k))$ then $(T', U') \in \text{REG}(T, U, k) \setminus \text{REG}(T, U, k-1)$.

By Prop.14, 11(2) and 2 we easily derive the next proposition. It is useful to look at Fig.11 (which is fully used later), and imagine $\sigma = \{(x_1, V_1), (x_2, V_2)\}$, $\sigma' = \{(x_1, V_1'), (x_2, V_2')\}$.

Proposition 15. *Suppose that $T \sim_k U$ and for σ, σ' we have $\text{SUPP}(\sigma) = \text{SUPP}(\sigma')$ and $(\sigma(x_i), \sigma'(x_i)) \in \text{REG}(T, U, k-1)$ for each $x_i \in \text{SUPP}(\sigma)$.*
If $\text{EQLV}(T', U') = \text{MINEL}(\text{REG}(T, U, k))$ and $T' = G\sigma$ then $\text{EQLV}(G\sigma', U') = \text{EQLV}(T', U')$.

Note that the case depicted in Fig.10 is a special case: $G = x_1$, $\sigma(x_1) = T'$, $\sigma'(x_1) = V$.

C. Prover-Refuter game

We now describe the rules of the game between Prover (she) and Refuter (he).

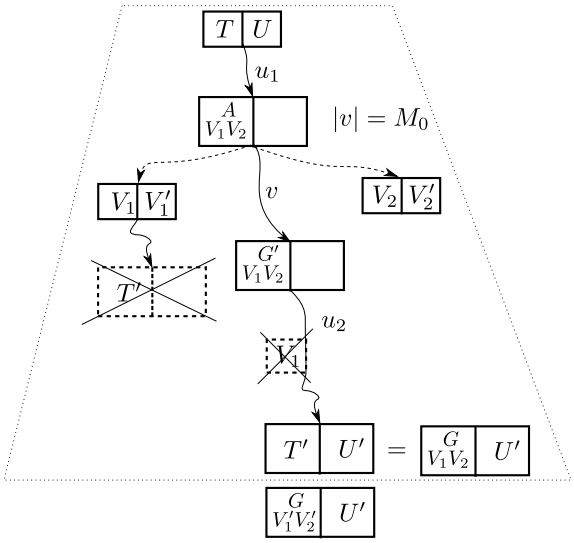

Fig. 11. Case 2 of left-balancing

PROVER-REFUTER GAME (P-R GAME)

1) A det-first-order grammar $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ is given.
2) Prover produces (by "guessing", say) a finite set BASIS of pairs of (graph presentations of regular) terms.
3) An input pair (T_{in}, U_{in}) is given.
4) Refuter chooses
   $(T_0, U_0) \in \text{STARTSET} = \{(T_{in}, U_{in})\} \cup \text{BASIS}$,
   and claims $\text{EQLV}(T_0, U_0) = \text{MINEL}(\text{STARTSET}) < \omega$.
5) For $i = 0, 1, 2, \ldots$, Phase i is performed, i.e.:
   a) Prover chooses $k > 0$, and $\text{REG}(T_i, U_i, k)$ is constructed; if $T_i \not\sim_k U_i$ then Prover loses (the play ends).
   b) Refuter chooses $(T_i', U_i') \in \text{REG}(T_i, U_i, k) \setminus \text{REG}(T_i, U_i, k-1)$ and w_i, $|w_i| = k$, such that $T_i \xrightarrow{w_i} T_i'$, $U_i \xrightarrow{w_i} U_i'$; if there is no such T_i', U_i', w_i (due to dead terms, hence $T_i \sim U_i$), Prover wins. Refuter claims that $\text{EQLV}(T_i', U_i') = \text{MINEL}(\text{REG}(T_i, U_i, k))$. (Recall Prop.14.)
   c) Prover produces (T_{i+1}, U_{i+1}) from (T_i', U_i') as follows:
      - either she puts $T_{i+1} = T_i'$, $U_{i+1} = U_i'$ (*no-change*),
      - or she *balances* (recall Prop.15 and Fig.11): if she finds σ, σ' such that $(\sigma(x_i), \sigma'(x_i)) \in \text{REG}(T, U, k-1)$ for all $x_i \in \text{SUPP}(\sigma) = \text{SUPP}(\sigma')$, and she presents T_i' as $G\sigma$ then she can (do a *left-balancing*, namely) put $T_{i+1} = G\sigma'$, and $U_{i+1} = U_i'$; symmetrically, if U_i' is $G\sigma'$ then she can (do a *right-balancing*, namely) put $T_{i+1} = T_i'$, and $U_{i+1} = G\sigma$.

      (Thus $\text{EQLV}(T_{i+1}, U_{i+1}) = \text{EQLV}(T_i', U_i')$ if Refuter's claim in 5.b is true. We have $T_{i+1} \sim U_{i+1}$ if $T_i \sim U_i$.)
   d) Prover *either contradicts Refuter's claims* by presenting a proof, i.e. a finite algorithmically verifiable sequence of deductions based on Propositions 2, 4, 11, 12, 13, in which case Prover wins, *or lets the play proceed* with Phase $i+1$.

If we switch Points 2) and 3) then we get the *weaker form of the game*; it is then clear that a play starts with a given

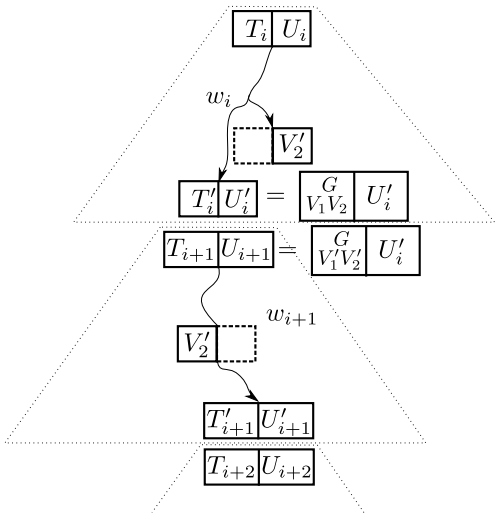

Fig. 12. A left balancing phase i followed by a no-change phase $i+1$

instance $\mathcal{G}, T_{in}, U_{in}$ of TRACE-EQ-DET-G. We use the above (stronger) form to stress that BASIS is determined by the grammar $\mathcal{G}$ (and is independent of T_{in}, U_{in}). We note that performing Point 5 in a play gives rise to a (finite or infinite) sequence of pairs

$$(T_1, U_1), (T_2, U_2), (T_3, U_3), \dots \tag{2}$$

which is eq-level decreasing (see Def.18(1)) if Refuter's claims are true; we have $T_j \sim U_j$ for all j if $T_0 \sim U_0$. An example of phases i and $i+1$ is depicted in Fig.12 (fully used later).

We can see that BASIS plays no role until possibly used in the final proof contradicting Refuter's claims. E.g., if (T_i, U_i) for $i > 0$ is shown to be a *basis-instance*, i.e. $(T_i, U_i) = (E\sigma, F\sigma)$ for some $(E, F) \in$ BASIS and some substitution σ, then this is a contradiction, since by Refuter's claims $\text{EQLV}(T_i, U_i) < \text{MINEL}(\text{STARTSET})$ (for $i > 0$) while $\text{EQLV}(E\sigma, F\sigma) \geq \text{EQLV}(E, F) \geq \text{MINEL}(\text{STARTSET})$ (by using Prop.11(1)). Another simple proof of contradiction is a *repeat*, i.e. getting $(T_j, U_j) = (T_i, U_i)$ for $j > i$.

Remark. We could surely make the game more flexible for Prover, adding her other sound possibilities but the above form is technically sufficient.

Example of a play of the Prover-Refuter game

We just show a simple play (not claiming anything particular).

1. $\mathcal{G}$ from Fig.5 is given.
2. Prover puts (guesses) $\text{BASIS} = \{(x_1, x_1), (Ax_1, Bx_1)\}$.
3. $(T_{in}, U_{in}) = (AB\bot, BA\bot)$ is given.
4. Refuter chooses $(T_0, U_0) = (T_{in}, U_{in}) = (AB\bot, BA\bot)$.

5a. Prover chooses $k = 2$ and constructs $\text{REG}(T_0, U_0, k)$ as in Fig.9, demonstrating that $T_0 \sim_k U_0$.

5b. Refuter chooses $(T_0', U_0') = (ABBB\bot, BAAA\bot)$, with $w_0 = aa$. ($T_0 \xrightarrow{w_0} T_0'$, $U_0 \xrightarrow{w_0} U_0'$.)

5c. Prover performs a left-balancing: she shows $(B\bot, A\bot) \in \text{REG}(T_0, U_0, k-1)$, puts $\sigma = \{(x_5, B\bot)\}$, $\sigma' = \{(x_5, A\bot)\}$, presents $T_0' = (ABBx_5)\sigma$, and puts $T_1 = (ABBx_5)\sigma'$; she thus defines $(T_1, U_1) = (ABBA\bot, BAAA\bot)$.

5d. Prover does not continue with Phase 1 but she derives a contradiction by assuming Refuter's claims are true as follows, denoting $\text{EQLV}(T_1, U_1) = \text{EQLV}(T_0', U_0') = \ell$: since the eq-levels of

$$(A\bot, B\bot),\ (AB\bot, BA\bot),\ (ABB\bot, BAA\bot)$$

are greater than ℓ, the eq-level of each of the following pairs (arising from (T_1, U_1) by successive subterm replacements) must be ℓ:
$(ABAB\bot, BAAA\bot)$ (we replaced $BA\bot$ in T_1),
$(ABAB\bot, BAAB\bot)$, $(ABAB\bot, BABA\bot)$,
$(ABAB\bot, BABB\bot)$, $(ABAB\bot, BBAA\bot)$,
$(ABAB\bot, BBAB\bot)$. But the last pair is an instance of the pair $(Ax_1, Bx_1) \in$ BASIS, a contradiction.

D. Soundness of the Prover-Refuter game

If $\{(T_{in}, U_{in})\} \cup$ BASIS contains a pair of non-equivalent terms then Refuter can be choosing so that his "least eq-level claims" (in 4. and 5.b) are true; thus the sequence (2) is eq-level decreasing and Prover loses eventually. This also applies to the weaker form of the P-R game (Points 2 and 3 switched).

Since BASIS is finite and Refuter always has finitely many choices when there is his turn, there is an obvious algorithmic aspect which we also capture in the next (soundness) lemma.

Lemma 16. *There is an algorithm with the following property: given a det-first order grammar $\mathcal{G}$ and T_{in}, U_{in}, it halts and produces some* BASIS *iff Prover can force her win for $\mathcal{G}$, T_{in}, U_{in} by using* BASIS *(in the weaker form of the game, say); in this case $T \sim U$ for all $(T, U) \in \{(T_{in}, U_{in})\} \cup$ BASIS.*

By combining with Lemma 9 we get an algorithm which decides TRACE-EQ-DET-G, on condition that for each det-first-order grammar $\mathcal{G}$ there exists a basis which is sufficient for forcing Prover's win for any $T_{in} \sim U_{in}$. This completeness is shown in Section IV, which will finish a proof of the next theorem.

Theorem 17. *Trace equivalence for det-first-order grammars (i.e., the problem* TRACE-EQ-DET-G*) is decidable.*

IV. COMPLETENESS OF THE PROVER-REFUTER GAME

IV-A shows that we get the completeness if there is $n \in \mathbb{N}$, $g : \mathbb{N} \to \mathbb{N}$ for any $\mathcal{G}$ such that Prover has a so-called (n, g)-strategy for each $T_0 \sim U_0$ (guaranteeing (n, g)-subsequences of sequences (2), as depicted on the left in Fig.13). IV-B then shows a "balancing strategy" for Prover which is an (n, g)-strategy.

A. "Stair-base" (n, g)-sequences are sufficient

We still assume a fixed det-first-order grammar $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ if not said otherwise.
We recall $\text{PRESSIZE}(E)$ (of a regular term E over $\mathcal{N}$), and put $\text{PRESSIZE}(E, F) = \text{PRESSIZE}(E) + \text{PRESSIZE}(F)$, say.

Definition 18.

1) A sequence $(T_1, U_1), (T_2, U_2), \dots, (T_\ell, U_\ell)$ *is* eq-level decreasing *if* $\omega > \text{EQLV}(T_1, U_1) > \text{EQLV}(T_2, U_2) > \dots > \text{EQLV}(T_\ell, U_\ell)$.

2) *A (finite or infinite) sequence ("head-pairs", or "heads")* $(E_1, F_1), (E_2, F_2), (E_3, F_3) \dots$
and a substitution σ *("tails") where* $\text{CARD}(\text{SUPP}(\sigma)) \leq n$ *constitute an* n-tail presentation *of the sequence* $(E_1\sigma, F_1\sigma), (E_2\sigma, F_2\sigma), (E_3\sigma, F_3\sigma), \dots$.

3) *For* $n \in \mathbb{N}$ *and a nondecreasing function* $g : \mathbb{N} \to \mathbb{N}$, $(T_1, U_1), (T_2, U_2), (T_3, U_3), \dots$ *is an* (n, g)-sequence *if it has an* n*-tail presentation given by tails* σ *and heads* $(E_1, F_1), (E_2, F_2), (E_3, F_3), \dots$
where $\text{PRESSIZE}(E_j, F_j) \leq g(j)$ *for* $j = 1, 2, 3, \dots$.
(Hence $(T_j, U_j) = (E_j\sigma, F_j\sigma)$ *for* $j = 1, 2, 3, \dots$.*)*

4) *Prover has an* (n, g)-strategy *for* $\mathcal{G}$ *if she can force that the sequence* $(T_1, U_1), (T_2, U_2), (T_3, U_3), \dots$ *arising in the phases* $0, 1, 2, \dots$ *(recall (2)) has an infinite* (n, g)*-subsequence in each play where* $T_0 \sim U_0$ *and the play does not finish with Prover's win in Point 5b or with a repeat. (The basis is irrelevant.)*

5) *Stipulating* $\max \emptyset = 0$*, we define the following finite number (Maximal Finite Equivalence Level) for any* $b \in \mathbb{N}$*:* $\text{MAXFEL}_b = \max\{\, \text{EQLV}(E, F) \mid E \not\sim F$ *and* $\text{PRESSIZE}(E, F) \leq b \,\}$.

The essence of the next lemma is the fact that the length of *eq-level decreasing* (n, g)-sequences is bounded by a number depending just on $\mathcal{G}, n, g$ (and independent of σ).

Lemma 19. *If Prover has an* (n, g)*-strategy for a det-first-order grammar* $\mathcal{G}$ *then there is some* BASIS *for* $\mathcal{G}$ *which is sufficient for Prover to force her win for all* $T_{in} \sim U_{in}$.

Proof: Suppose a fixed $\mathcal{G}$, and n, g guaranteed by the assumption. Consider a play of the P-R game in which $\mathcal{G}$ is given (in Point 1). Suppose Prover (in Point 2) produces $\text{BASIS} = \{(E, F) \mid E \sim F, \text{PRESSIZE}(E, F) \leq \mathcal{B}\}$ for some (large) bound $\mathcal{B} \in \mathbb{N}$. We can imagine that Prover has an unlimited computational power and "computes" (guesses) $\mathcal{B}$ and BASIS when knowing $\mathcal{G}, n, g$. Our reasoning below will put some conditions on $\mathcal{B}$ which Prover can "foresee" and which will guarantee that BASIS "handles" all $T_{in} \sim U_{in}$.

So assume that some $T_{in} \sim U_{in}$ is given in Point 3. (This could not be foreseen by Prover.) Refuter then necessarily chooses $T_0 \sim U_0$ in Point 4 (though claiming $T_0 \not\sim U_0$). Let Prover use her assumed (n, g)-strategy, and consider a moment (after a number of phases) when the so far constructed sequence $(T_1, U_1), (T_2, U_2), \dots$ has a "*long*" (n, g)-subsequence $(T_{i_1}, U_{i_1}) = (E_1\sigma, F_1\sigma)$, $(T_{i_2}, U_{i_2}) = (E_2\sigma, F_2\sigma)$, $\dots$, $(T_{i_\ell}, U_{i_\ell}) = (E_\ell\sigma, F_\ell\sigma)$; we write ℓ as $\ell_{(n,g)}$ for later use.

Prover can derive from Refuter's claims that

$$(E_1\sigma, F_1\sigma), (E_2\sigma, F_2\sigma), \dots, (E_{\ell_{(n,g)}}\sigma, F_{\ell_{(n,g)}}\sigma) \qquad (3)$$

is eq-level decreasing (though in reality $E_i\sigma \sim F_i\sigma$ for all i).

If $E_1 \sim F_1$ (which must be the case when $n = 0$, so when $\text{SUPP}(\sigma) = \emptyset$) then Prover can claim her win if she has chosen $\mathcal{B} \geq g(1)$ (so we have a first condition on $\mathcal{B}$, which Prover could "foreseee" in Point 2): in this case $(T_{i_1}, U_{i_1}) = (E_1\sigma, F_1\sigma)$ is a basis-instance (and Prover can claim her win).

Assume now $\text{EQLV}(E_1, F_1) = k \in \mathbb{N}$; note that $k \leq \text{MAXFEL}_{g(1)}$. Since $E_1\sigma \sim F_1\sigma$, by Prop.12 we know that

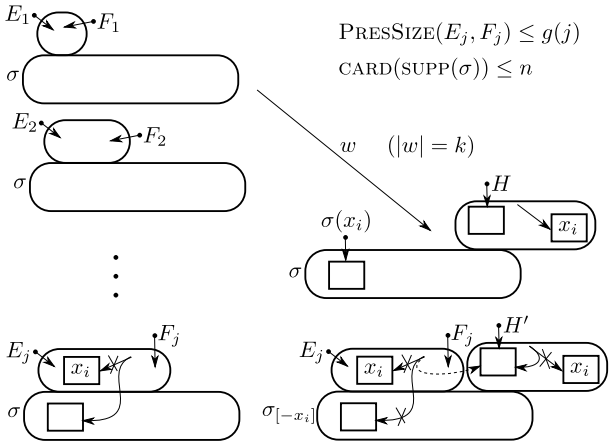

Fig. 13. An (n, g)-(sub)sequence (left); decreasing $\text{SUPP}(\sigma)$ by $\{(x_i, H')\}$

Prover can demonstrate $E_1\sigma \xrightarrow{w} \sigma(x_i)$ and $F_1\sigma \xrightarrow{w} H\sigma$ (or vice versa) for $x_i \in \text{SUPP}(\sigma)$, $H \neq x_i$ and $|w| = k$ (see Fig.13). Moreover, she derives $\text{EQLV}(\sigma(x_i), H\sigma) > \text{EQLV}(E_{s+1}\sigma, F_{s+1}\sigma) > \text{EQLV}(E_{s+2}\sigma, F_{s+2}\sigma) > \cdots$ for (the shift) $s = 1 + \text{MAXFEL}_{g(1)}$.

Using (deduction rules based on) Proposition 13, 11(2) and 2, Prover can demonstrate that in the pairs $(E_j\sigma, F_j\sigma)$, for $j = s{+}1, s{+}2, \dots$, she can replace σ with $\{(x_i, H')\}\sigma_{[-x_i]}$ where $\text{GP}_{H'}$ arises from GP_H by redirecting each incoming arc of x_i to the root (see Fig.13), *without affecting the eq-levels* of these pairs if Refuter's claims are true.

Note that $\text{PRESSIZE}(H)$ is surely bounded by $g(1) + \text{MAXFEL}_{g(1)} \cdot \text{STEPINC}$, where STEPINC can be taken as the size of the largest rhs in the rules of $\mathcal{G}$; it bounds the possible *one-step increase* of the presentation size when a rule is applied (recall Fig.6).

Prover thus demonstrates an $(n{-}1, g')$-sequence

$$(E'_1\sigma_{[-x_i]}, F'_1\sigma_{[-x_i]}), (E'_2\sigma_{[-x_i]}, F'_2\sigma_{[-x_i]}), \dots \qquad (4)$$

of length $\ell_{(n-1,g')} = \ell_{(n,g)} - (1 + \text{MAXFEL}_{g(1)})$ where $E'_j = E_{s+j}\{(x_i, H')\}$ and $F'_j = F_{s+j}\{(x_i, H')\}$; we note that $\text{PRESSIZE}(E'_j, F'_j)$ is surely bounded (by $g(s{+}j) + 2 \cdot \text{PRESSIZE}(H)$ and thus) by

$$g'(j) \text{ defined as} \qquad (5)$$

$$g(1{+}\text{MAXFEL}_{g(1)}{+}j){+}2{\cdot}(g(1){+}\text{MAXFEL}_{g(1)}{\cdot}\text{STEPINC}).$$

We can now reason for the sequence (4) as we did for the sequence (3): if $E'_1 \sim F'_1$ then Prover can claim her win if she has chosen $\mathcal{B} \geq g'(1)$ ("computing" g' from $\mathcal{G}$ and g). Otherwise Prover creates an $(n{-}2, g'')$-sequence of length $\ell_{(n-2,g'')} = \ell_{(n-1,g')} - (1 + \text{MAXFEL}_{g'(1)})$, etc. The iteration can happen at most n-times. Since n is known in Point 2, Prover could choose $\mathcal{B}$ sufficiently large to force her win (by creating a basis-instance with the eq-level less than $\text{MINEL}(\text{STARTSET})$ according to Refuter). ■

Remark. We note a "side-effect" of the above proof; though not important now, it will be used in the analogue of Lemma 19 for the general bisimilarity problem. Let us consider the following

recursive definition of $\ell_{(n,g)}$ (for all $n \in \mathbb{N}$, $g : \mathbb{N} \to \mathbb{N}$ when $\mathcal{G}$ is assumed):

$$\text{if } n = 0 \text{ then } \ell_{(n,g)} = \ell_{(0,g)} = 1 + \text{MAXFEL}_{g(1)}$$
$$\text{otherwise } \ell_{(n,g)} = \ell_{(n-1,g')} + (1 + \text{MAXFEL}_{g(1)})$$
$$\text{where } g' \text{ is defined as in (5).}$$

Using the reasoning in the proof of Lemma 19, by induction on n we can easily verify that $\ell_{(n,g)}$ provides an upper bound on the length of eq-level decreasing (n, g)-sequences. Computability of $\ell_{(n,g)}$ is another matter to which we return in the general bisimilarity case.

B. Balancing yielding an (n, g)-strategy

We first introduce some notions and make some observations which are used in describing a particular balancing strategy of Prover; then we show that this strategy is an (n, g)-strategy (for the assumed $\mathcal{G}$).

(Shortest) sink words; root-performability

In the general (nondeterministic) case, the following notions make better sense in the det-LTS $\mathcal{L}^{\text{R}}_{\mathcal{G}}$ (recall Def.6), but we can stick to our assumed det-LTS $\mathcal{L}^{\text{A}}_{\mathcal{G}}$ here.

Definition 20.

1) $w \in \mathcal{A}^*$ *is a* (Y, j)-sink-word, $1 \le j \le m = arity(Y)$, *if* $Yx_1 \dots x_m \xrightarrow{w} x_j$ *(hence* $YF_1 \dots F_m \xrightarrow{w} F_j$*).*
2) *A* path $F \xrightarrow{u}$ *(of length* $|u|$*) is* root-performable *if the root of* F *is* $Y \in \mathcal{N}$ *and* $Yx_1 \dots x_{arity(Y)} \xrightarrow{u}$*; hence no proper prefix of* u *is a* (Y, j)*-sink word then; if, moreover,* u *itself is not a* (Y, j)*-sink word then* $F \xrightarrow{u}$ *is* strongly root-performable.
3) *A* path $G \xrightarrow{w}$ sinks into depth k in $\text{DOM}(G)$ *(recall* $G : \mathbb{N}^* \to \mathcal{N} \cup \text{VAR}$*) if it sinks* along *some* $\gamma = i_1 i_2 \dots i_k \in \text{DOM}(G)$*, i.e. if* $w = w_1 w_2 \cdots w_k$ *and for each* ℓ*,* $1 \le \ell \le k$*, we have:* w_ℓ *is a* (Y, i_ℓ)*-sink word where* $Y = G(i_1 i_2 \dots i_{\ell-1})$*. (Hence* $G \xrightarrow{\varepsilon}$ *sinks into depth* 0*.)*

In Fig.6, a is a $(Y, 1)$-sink-word of length 1. In Fig.7, if $root(F) = A$ and the arc depicted in GP_F is labelled 2 then $a_1a_2a_3a_4a_5a_6$ is an $(A, 2)$-sink word. $F \xrightarrow{a_1a_2\dots a_j}$ is root-performable for all $j, 0 \le j \le 6$, but it is not strongly root-performable for $j = 6$.

It is useful to realize the next trivial fact (recall again Fig.6).

Proposition 21. *If* w *is a* (Y, j)*-sink-word then* $w = av$ *for a rule* $r : Yx_1 \dots x_m \xrightarrow{a} E$ *where* $E \xrightarrow{v}$ *sinks along some* $\gamma \in \text{DOM}(E)$ *where* $E(\gamma) = x_j$ *(so* $E \xrightarrow{v} x_j$*). (*$E = x_j$ *is a particular case.)*

It is thus clear that we can efficiently (by standard dynamic programming techniques) fix a *shortest* (Y, j)*-sink word* $\text{SSW}(Y, j)$ for each pair Y, j, $Y \in \mathcal{N}$, $1 \le j \le arity(Y)$ (in our assumed $\mathcal{G}$) for which there is such a word. If there is no (Y, j)-sink word (so the j-th successor of Y is nonexposable and thus irrelevant), we can safely decrease $arity(Y)$ and make the obvious corresponding modifications in the rules of $\mathcal{G}$. Hence we further assume $\text{SSW}(Y, j)$ for each Y, j, and put

$$M_0 = 1 + \max\{\, |\text{SSW}(Y, j)| \mid Y \in \mathcal{N}, 1 \le j \le arity(Y)\}. \quad (6)$$

M_0-sinking paths (no length-M_0 subpath is root-performable)

Definition 22. *We put* $\text{DEPTH}(E) = \max\{|\gamma| \mid \gamma \in \text{DOM}(E)\}$ *for* finite *terms* E*, and we define the* maximal one-step depth-increase*:*

$$\text{STEPDEPTHINC} = \max\{\text{DEPTH}(E) - 1 \mid E \text{ is the rhs of a rule in } \mathcal{G}\}.$$

A path $T \xrightarrow{w} T'$ *is* d-sinking, $d > 0$*, if there is no root-performable subpath of length* d *in* $T \xrightarrow{w} T'$*, i.e., we cannot write* $w = u_1 v u_2$, $|v| = d$*, so that* $T \xrightarrow{u_1} (Ax_1 \dots x_m)\sigma \xrightarrow{v} G'\sigma \xrightarrow{u_2} T'$ *where* $Ax_1 \dots x_m \xrightarrow{v} G'$*.*

In fact, we only use M_0-sinking paths. The path $T \xrightarrow{u_1 v u_2} T'$ in Fig.11 is not M_0-sinking.

The next fact is trivial; if we have a long M_0-sinking path $T \xrightarrow{w}$ then its prefix sinks deeply in $\text{DOM}(T)$ (increasing the sink-depth within every M_0 steps), which is possibly followed by a "growing end" shorter than M_0:

Proposition 23. *If a* path $T \xrightarrow{w} T'$ *is* M_0-sinking*, then we can write* $w = v_1 v_2$ *where* $T \xrightarrow{v_1} T_1 \xrightarrow{v_2} T'$, $|v_2| < M_0$, $T \xrightarrow{v_1} T_1$ *sinks into depth at least* $d = |w| \,div\, M_0$ *in* $\text{DOM}(T)$*, and* T' *can be written* $F\sigma$ *for a finite term* F *where* $\text{DEPTH}(F) \le 1 + (M_0 - 1) \cdot \text{STEPDEPTHINC}$ *and each* $\sigma(x_j)$ *(for* $x_j \in \text{SUPP}(\sigma)$*) is a subterm of* T_1 *and thus of* T*.*

(If $v_2 = \varepsilon$ then we take $F = x_1$ and $\sigma(x_1) = T_1 = T'$. Otherwise $T_1 = (Ax_1 \dots x_m)\sigma$, hence all $\sigma(x_j)$, $1 \le j \le m$, are subterms of T_1, and we have $Ax_1 \dots x_m \xrightarrow{v_2} F$.)

Figures 10 and 11 help to observe the next fact (saying that we can bound $\text{DEPTH}(G)$ in Fig.11).

Proposition 24. *Consider a region* $\text{REG}(T, U, k)$, $T \sim_k U$*, and* T', U', w, $|w| = k$*, such that* $T \xrightarrow{w} T'$, $U \xrightarrow{w} U'$*. Suppose there is no* $(T', V) \in \text{REG}(T, U, k-1)$ *and* $T \xrightarrow{w} T'$ *is not* M_0*-sinking. Take the last root-performable (sub)path of length* M_0 *in* $T \xrightarrow{w} T'$*, i.e., take the longest* u_1 *such that* $T \xrightarrow{u_1} (Ax_1 \dots x_m)\sigma \xrightarrow{v} G'\sigma \xrightarrow{u_2} T'$ *where* $Ax_1 \dots x_m \xrightarrow{v} G'$ *and* $|v| = M_0$*. Then there is* $(\sigma(x_j), V'_j) \in \text{REG}(T, U, k-1)$ *for each* $j, 1 \le j \le m$*, and* $G' \xrightarrow{u_2} G$ *(hence* $T' = G\sigma$*) where*
$\text{DEPTH}(G) \le 1 + (2M_0 - 1) \cdot \text{STEPDEPTHINC}$*.*

Proof: For each j, $1 \le j \le m$, we have $Ax_1 \dots x_m \xrightarrow{\text{SSW}(A,j)} x_j$, and $|\text{SSW}(A, j)| < M_0 = |v|$; hence there is a pair $(\sigma(x_j), V'_j)$ in $\text{REG}(T, U, k-1)$ (since $T \sim_k U$). We cannot have $G'\sigma \xrightarrow{u'} \sigma(x_j)$ for a prefix u' of u_2 since otherwise we had $u_2 = u'u''$, $\sigma(x_j) \xrightarrow{u''} T'$ and thus we had $(T', V) \in \text{REG}(T, U, k-1)$ (see Fig.11). We thus have $G' \xrightarrow{u_2} G$; moreover, $G' \xrightarrow{u_2} G$ is M_0-sinking (since we took the last root-performable path of length M_0). Since $\text{DEPTH}(G') \le 1 + M_0 \cdot \text{STEPDEPTHINC}$, with Prop.23 we easily verify the claim. ∎

Recall Fig.12 and suppose that $T_{i+1} \xrightarrow{w_{i+1}} T'_{i+1}$ is M_0-sinking. We then need that the "rest-head" G is "erased" which is guaranteed when we have an M_1-distance region for

$M_1 \geq M_0 \cdot (1 + \text{DEPTH}(G))$. Therefore we put

$$M_1 = M_0 \cdot (2 + (2M_0{-}1) \cdot \text{STEPDEPTHINC}). \quad (7)$$

(Restricted) left and right balancing steps

From now on we assume that Prover always chooses $k = M_1$ in Point 5a of the P-R game. We now also *restrict the allowed way of balancing*.

Imagine the game is in 5c in Phase i; we have $T_i \xrightarrow{w_i} T_i'$, $U_i \xrightarrow{w_i} U_i'$. If Prover wants to perform a *left balancing step*, she does the following (recall Fig.10 and Fig.11):

1) If there is some (T_i', V) in $\text{REG}(T_i, U_i, M_1{-}1)$, Prover (chooses one such pair and) puts $T_{i+1} = V$, $U_{i+1} = U_i'$.
2) Otherwise if there is a root-performable (sub)path of length M_0 in $T_i \xrightarrow{w_i} T_i'$, she takes the last one, getting $T_i \xrightarrow{u_1} (Ax_1 \dots x_m)\sigma \xrightarrow{v} G'\sigma \xrightarrow{u_2} T_i' = G\sigma$ as in Prop.24 ($u_1 v u_2 = w_i$). She defines $\sigma'(x_j) = V_j'$ for each $j, 1 \leq j \leq m$, where $(\sigma(x_j), V_j') \in \text{REG}(T, U, k{-}1)$, and puts $T_{i+1} = G\sigma'$, $U_{i+1} = U_i'$.
3) If none of 1, 2 applies, hence $T_i \xrightarrow{w_i} T_i'$ is M_0-sinking, then no left-balancing is possible.

In the cases 1 and 2, U_i is called the *balancing pivot* and (T_{i+1}, U_{i+1}) the *balancing result* (or the *bal-result*) of this balancing step. The *right balancing steps* are defined symmetrically (T_i is then the pivot).

Balancing strategy for Prover

We further assume that Prover behaves as follows in Phase i:

She balances, i.e. performs a left balancing or a right balancing step (as in 1 or 2 above), if possible but she cannot do a left (right) balancing if a right (left) balancing was done in Phase $i{-}1$; if balancing is (thus) not possible, Prover does no-change, i.e. puts $T_{i+1} = T_i'$, $U_{i+1} = U_i'$. (Prover thus cannot switch balancing sides in two consecutive phases; such a switch needs a separating no-change phase.)

Remark. We use a liberal notion of a strategy since it might leave some free choice to the player adhering to the strategy.

Pivots of a play are on a special pivot-path in $\mathcal{L}^{\text{A}}_{\mathcal{G}}$

Proposition 25. *(1.) If Prover balances in Phase i and Phase $i+1$, then we have $W \xrightarrow{w_i} W'$, $|w_i| = M_1$, for the respective pivots. ($W = U_i \xrightarrow{w_i} U_{i+1} = W'$ in the case of left balancing, and $W = T_i \xrightarrow{w_i} T_{i+1} = W'$ in the case of right balancing).*

(2.) If Prover balances in Phase i, with pivot W ($W = U_i$ or $W = T_i$), and does no-change in Phase $i+1$, then there are words v', v'' of length at most $2M_1$ such that $W \xrightarrow{v'} T_{i+2}$, $W \xrightarrow{v''} U_{i+2}$ (i.e., after Phase $i+1$ the terms on both sides are "shortly" reachable from the last pivot).

Proof: The first part is trivial. For the second part assume that left-balancing was done in Phase i and no change in Phase $i+1$ (as in Fig.12); hence $W = U_i$, and we have $W \xrightarrow{w_i w_{i+1}} U_{i+2}$ (where $|w_i w_{i+1}| = 2M_1$). Moreover, $T_{i+1} \xrightarrow{w_{i+1}} T_{i+2}$ is M_0-sinking, hence w_{i+1} has a prefix w such that $T_{i+1} \xrightarrow{w}$ sinks into depth at least $M_1 \, div \, M_0 = 2 + (2M_0{-}1) \cdot \text{STEPDEPTHINC}$ in $\text{DOM}(T_{i+1})$.

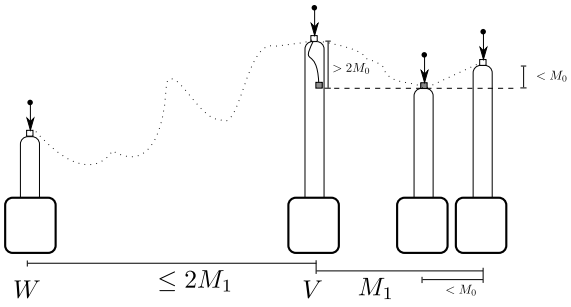

Fig. 14. (A prefix of) the path from a pivot W to the next pivot

If the balancing result is $(T_{i+1}, U_{i+1}) = (V, U_i')$ as in 1 (see Fig.10) then $W = U_i \xrightarrow{u} V \xrightarrow{w_{i+1}} T_{i+2}$, where $|u| < M_1$.

Now assume $(T_{i+1}, U_{i+1}) = (G\sigma', U_i')$ (Fig.11) where for each $x_j \in \text{SUPP}(\sigma')$ there is u_j', $|u_j'| < M_1$, such that $W = U_i \xrightarrow{u_j'} \sigma'(x_j)$; recall also that $\text{DEPTH}(G) \leq 1 + (2M_0{-}1) \cdot \text{STEPDEPTHINC}$ (by Prop.24).

We have chosen M_1 large enough so that $T_{i+1} = G\sigma' \xrightarrow{w}$ for the above mentioned prefix w of w_{i+1} sinks into greater depth than $\text{DEPTH}(G)$; hence there is a prefix w' of w_{i+1}, we put $w_{i+1} = w'w''$, such that $G\sigma' \xrightarrow{w'} \sigma'(x_j) \xrightarrow{w''} T_{i+2}$. Hence $W = U_i \xrightarrow{u_j'} \sigma'(x_j) \xrightarrow{w''} T_{i+2}$ ($|u_j w''| < 2M_1$). ■

Fig.14 shows a path from a pivot W to the next pivot. It starts with a (sub)path of length $\leq 2M_1$, corresponding *either* to (1) in Prop.25, in which case the length of this "starting path" is M_1 and the path finishes with the next pivot, *or* to (2) in Prop.25, in which case this starting path finishes with $V \in \{T_{i+2}, U_{i+2}\}$, for the appropriate i, depending on which side (left or right) the next pivot is. (In Fig.14 the starting path gives an impression of term-increasing but this is not true in general.) In the above case (2), the starting path, finishing in V, might be followed by a sequence of "follow-up" paths (where the last one finishes in the next pivot); each of these follow-up paths has length M_1 and is M_0-sinking. (Fig.14 depicts just one follow-up path.) Here our choice of M_1 guarantees (by Prop.23) that we have "term-sinking", in particular any path (of length M_1) in this follow-up sequence necessarily visits a subterm of V (in the ever greater depth in $\text{DOM}(V)$). (In Fig.14 we assume $\text{STEPDEPTHINC} = 1$.) We have thus shown:

Proposition 26. *The pivots of the balancing steps used in a play of P-R game (where Prover uses the above balancing strategy) are on a (finite or infinite) path*

$$W_1 \xrightarrow{v_1} W_2 \xrightarrow{v_2} W_3 \xrightarrow{v_3} \cdots \quad (8)$$

where each (sub)path $W_j \xrightarrow{v_j} W_{j+1}$ can be written $W_j \xrightarrow{v'} V \xrightarrow{v''} W_{j+1}$ so that $|v'| \leq 2M_1$ and $V \xrightarrow{v''} W_{j+1}$ is a sequence of "follow-up" (sub)paths of length M_1; each of these follow-up paths (of length M_1) is M_0-sinking, and thus visits a subterm of V.

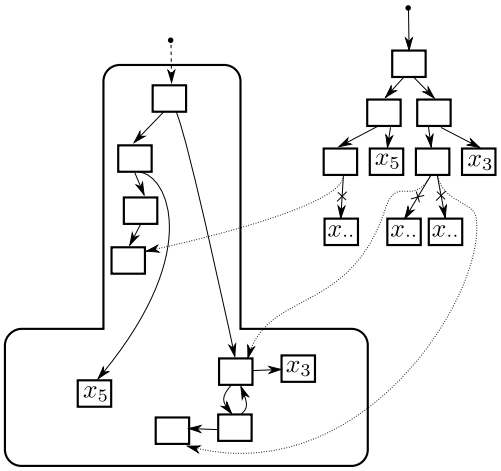

Fig. 15. A term V (with dotted root) presented as $(\text{TOP}_3^V)\sigma$

Balancing strategy is an (n, g)-strategy

In the proof of the next lemma (which finishes the proof of Theorem 17) we also use the following notion which we highlight for repeated trivial use: within d steps from V we can sink at most to depth d in $\text{DOM}(V)$.

Definition 27. *For a (regular) term V and $d \in \mathbb{N}$ we define TOP_d^V as the finite term with $\text{DEPTH}(\text{TOP}_d^V) \leq d$ which coincides with V on all $\gamma \in \text{DOM}(V)$, $|\gamma| < d$, but $\text{TOP}_d^V(\gamma)$ for $|\gamma| = d$ is a (fresh) variable x_{i_γ} unique to γ. (We assume a natural order on $D = \{\gamma \in \text{DOM}(V) \mid |\gamma| = d\}$ and we always use the smallest available index for i_γ.)*

By putting $\text{SUPP}(\sigma) = \{x_{i_\gamma} \mid \gamma \in D\}$ and $\sigma(x_{i_\gamma}) = V_\gamma$ (the subterm of V occurring at γ), we get the top-d presentation *of V, namely $V = (\text{TOP}_d^V)\sigma$.*

Fig.15 shows a top-3 presentation of some V. Note that we cannot use the variables in $\{V(\gamma) \mid |\gamma| < d\}$ as the variables in $\text{SUPP}(\sigma)$, but we can assume $\text{SUPP}(\sigma) \subseteq \{x_1, x_2, \ldots, x_{c^d}\}$ where $c = \max\{\, arity(Y) \mid Y \in \mathcal{N}\,\}$.

Lemma 28. *For any det-first-order grammar $\mathcal{G}$, the balancing strategy is an (n, g)-strategy (n, g are determined by $\mathcal{G}$).*

Proof: Assume an infinite play where $T_0 \sim U_0$, Prover uses the balancing strategy and there is no repeat. Prover then balances infinitely often: Otherwise for some $i \geq 0$ all $T_i \xrightarrow{w_i} T_{i+1}$, $T_{i+1} \xrightarrow{w_{i+1}} T_{i+2}$, ... are M_0-sinking (corresponding to the follow-up paths after V in Fig.14), and Prop.23 and our choice of M_1 easily yield that all T_j range over finitely many terms (since their presentation size is bounded). Similarly for U_j, so there would be a repeat.

The pivot path (8) of our assumed play is thus infinite, $W_1 \xrightarrow{v_1} W_2 \xrightarrow{v_2} \cdots$. If a term V' (not only a pivot) is visited infinitely often by (8) then we have $W_{i_1} = W_{i_2} = \cdots$ for infinitely many i_j: any particular visit of V' occurs in the path $W_j \xrightarrow{v_j} W_{j+1}$ for some j (V' is somewhere in the path in Fig.14), and $\text{PRESSIZE}(W_{j+1})$ can be obviously only boundedly bigger than $\text{PRESSIZE}(V')$. It is easy to check that the balancing results would be then infinitely often the same

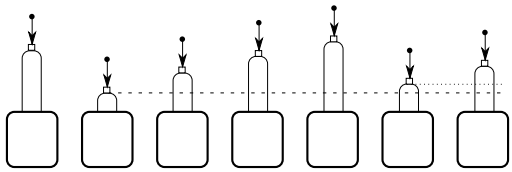

Fig. 16. First steps in path (8), the second term happens to be a "stair-base"

(recall Fig.10 and 11); we would have a repeat.

Hence there is a visit of a term $V = (Yx_1 \ldots x_m)\sigma'$ in (8) (a "stair-base", depicted as the second term in Fig.16) such that no subterm of V is visited later (so the rest of (8) is strongly root-performable); we can thus write (8) as

$$W_1 \xrightarrow{u} (Yx_1 \ldots x_m)\sigma' \xrightarrow{u'} H_1\sigma' \xrightarrow{v_{k+1}} H_2\sigma' \xrightarrow{v_{k+2}} \cdots$$

where $(Yx_1 \ldots x_m) \xrightarrow{u'} H_1 \xrightarrow{v_{k+1}} H_2 \xrightarrow{v_{k+2}} \cdots$, and $H_j\sigma' = W_{k+j}$ ($j = 1, 2, \ldots$) are the pivots after V. We can check (by Fig.14) that $\text{DEPTH}(H_j) \leq 1 + j \cdot 2M_1 \cdot \text{STEPDEPTHINC}$.

Recall that the balancing result with pivot $H_j\sigma'$ is composed from terms (like V, V'_j, U' in Fig.10, 11) reachable from the pivot by at most M_1 moves, possibly using a bounded "rest-head" G. Recalling Def.27 (and Fig.15), we write

$$V = (Yx_1 \ldots x_m)\sigma' = (\text{TOP}_{M_1}^V)\sigma = ((Yx_1 \ldots x_m)\sigma'')\sigma$$

where $\text{TOP}_{M_1}^V = (Yx_1 \ldots x_m)\sigma''$ and $\sigma' = \sigma''\sigma$. Moreover, we have $\text{SUPP}(\sigma) \subseteq \{x_1, x_2, \ldots, x_n\}$ where

$$n = c^{M_1} \text{ for } c = \max\{\, arity(Y) \mid Y \in \mathcal{N}\,\}. \qquad (9)$$

Hence $W_{k+j} = H_j\sigma''\sigma$, and the balancing result with W_{k+j} can be written $(E_j\sigma, F_j\sigma)$, where E_j, F_j are finite terms with $\text{DEPTH}(E_j)$, $\text{DEPTH}(F_j)$ bounded by $\text{DEPTH}(H_j)+M_1+M_1 \cdot \text{STEPDEPTHINC} + (1 + (2M_0-1) \cdot \text{STEPDEPTHINC})$ (recall Fig.10 and 11, and $\text{DEPTH}(G)$ in Prop.24).

This obviously gives some $g : \mathbb{N} \to \mathbb{N}$ (determined by $\mathcal{G}$) such that $\text{PRESSIZE}(E_j, F_j) \leq g(j)$. The bal-results related to pivots $W_{k+1}, W_{k+2}, W_{k+3}, \ldots$ thus constitute an infinite (n, g)-subsequence of the sequence $(T_1, U_1), (T_2, U_2), \ldots$ arising in the phases $0, 1, \ldots$ of our assumed play. ■

V. AN UPPER BOUND ON COMPLEXITY

Remark. The next section VI is independent of this section.

An *elementary function* $\mathbb{N}^k \to \mathbb{N}$ arises by a finite composition of constants, the elementary operations $+, -, \cdot, div$ and the exponential operator $\uparrow$, where $a \uparrow n = a^n$.

Convention. In our context, when we say that a *number* is *elementary*, we mean that there is an elementary function of the size of the underlying det-first-order grammar $\mathcal{G}$ which gives an upper bound on the number.

E.g., the numbers M_0, M_1 (see (6), (7)), as well as n in the proof of Lemma 28 (see (9)) are obviously elementary.

The first nonelementary (hyper)operator is *iterated exponentiation* $\uparrow\uparrow$, also called *tetration*: $a \uparrow\uparrow n = a \uparrow (a \uparrow (a \uparrow (\ldots a \uparrow a)\ldots))$ where $\uparrow$ is used n-times.

Our analysis will yield the following theorem, with an obvious algorithmic consequence.

Theorem 29. *For any triple $\mathcal{G}, T_0, U_0$ with the size* INSIZE *(of a standard presentation) where $\mathcal{G}$ is a det-first-order grammar and $T_0 \not\sim U_0$ we have* $\text{EQLV}(T_0, U_0) \leq 2 \uparrow\uparrow f(\text{INSIZE})$*, where f is an elementary function independent of $\mathcal{G}, T_0, U_0$.*

Corollary 30. *Trace equivalence for det-first-order grammars can be decided in time (and space) $O(2 \uparrow\uparrow g(\text{INSIZE}))$ for an elementary function g.*

An analogous claim holds for language equivalence of deterministic pushdown automata, as follows from the reduction shown in Section II.

We now aim to prove Theorem 29. V-A shows a stronger congruence property (in our deterministic case), which enables to show that there are no eq-level decreasing $(n, n{+}1)$-sequences introduced in V-B. V-C then shows that this yields a proof of Theorem 29.

We again assume a det-first-order grammar $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$, and the det-LTS $\mathcal{L}^{\text{A}}_{\mathcal{G}}$, if not said otherwise.

A. Stronger congruence properties

By Prop.11, $\sigma_1 \sim_{\ell+1} \sigma_2$ implies $E\sigma_1 \sim_{\ell+1} E\sigma_2$. Thus if $\text{EQLV}(E, F) = k \leq \ell = \text{EQLV}(E\sigma_1, F\sigma_1)$ and $\sigma_1 \sim_{\ell+1} \sigma_2$ then $\text{EQLV}(E\sigma_1, F\sigma_1) = \text{EQLV}(E\sigma_2, F\sigma_1) = \text{EQLV}(E\sigma_2, F\sigma_2)$ (using Prop.2). We now observe that for getting $\text{EQLV}(E\sigma_2, F\sigma_2) = \text{EQLV}(E\sigma_1, F\sigma_1)$ it is sufficient to have $\sigma_1 \sim_{\ell-k+1} \sigma_2$. (I.e., if we replace σ_1 with σ_2 on both sides simultaneously then the requirement is weaker.)

Proposition 31. *Assume* $\text{EQLV}(E, F) = k < \omega$ *and* $\text{EQLV}(E\sigma_1, F\sigma_1) = \ell$ *(hence $\ell \geq k$). If $\sigma_1 \sim_{\ell-k+1} \sigma_2$ then* $\text{EQLV}(E\sigma_2, F\sigma_2) = \text{EQLV}(E\sigma_1, F\sigma_1) = \ell$.

Proof: We recall Prop.8 and the fact that the assumption $E \sim_k F$ implies $E \xrightarrow{w} x_j$ iff $F \xrightarrow{w} x_j$ for any w, $|w| < k$. Hence if $E \xrightarrow{w} x_j$ and $F \xrightarrow{w} H$ where $H \neq x_j$ then $\ell{-}|w| \leq \ell{-}k$, $\sigma_1(x_j) \sim_{\ell-|w|} H\sigma_1$ (by Prop.4), and thus $\sigma_2(x_j) \sim_{\ell-|w|} H\sigma_2$; moreover, if $\text{EQLV}(\sigma_1(x_j), H\sigma_1) \leq \ell{-}k$ then $\text{EQLV}(\sigma_2(x_j), H\sigma_2) = \text{EQLV}(\sigma_1(x_j), H\sigma_1)$ (by Prop.2). We can thus verify that $\text{EQLV}(E\sigma_2, F\sigma_2) = \ell$. ∎

B. Recurrent-pattern sequences, called (n, t)-sequences

We introduce (n, t)-*sequences* where $t \in \mathcal{N}$ denotes that the sequence has 2^t elements=pairs. In contrast with (n, g)-sequences, the sizes play no role; we just capture a recurrent pattern. A crucial fact will be that there are no *eq-level decreasing* $(n, n{+}1)$-sequences. (Recall now 2 in Def.18.)

Definition 32. *A presentation consisting of σ (tails) and (E_1, F_1) (just one head-pair) is an* $(n, 0)$-presentation with the support set $\text{SUP} \subseteq \text{VAR}$ *(presenting the one-element sequence $(E_1\sigma, F_1\sigma)$) if* $\text{CARD}(\text{SUP}) \leq n$ *and* $\text{SUPP}(\sigma) \subseteq \text{SUP}$.

If $\mathcal{P}_1$ is an (n, t)-presentation with SUP*, consisting of σ and $(E_1, F_1), (E_2, F_2), \ldots, (E_{2^t}, F_{2^t})$, then for any σ',* $\text{SUPP}(\sigma') \subseteq \text{SUP}$*, the presentation $\mathcal{P}$ with tails σ and heads*

$$(E_1, F_1), \ldots, (E_{2^t}, F_{2^t}), (E_1\sigma', F_1\sigma'), \ldots, (E_{2^t}\sigma', F_{2^t}\sigma') \tag{10}$$

is an $(n, t{+}1)$ presentation with SUP.

*A sequence $(T_1, U_1), (T_2, U_2), \ldots, (T_m, U_m)$ is an (n, t)-*sequence *if it has an (n, t)-presentation (with some* SUP*).*

E.g., if the supports of $\sigma, \sigma_1, \sigma_2$ are subsets of SUP and $\text{CARD}(\text{SUP}) \leq n$ then σ (as tails) and the following pairs (heads) constitute an $(n, 2)$-presentation:

$$(E_1, F_1), (E_1\sigma_1, F_1\sigma_1), (E_1\sigma_2, F_1\sigma_2), (E_1\sigma_1\sigma_2, F_1\sigma_1\sigma_2). \tag{11}$$

(By adding 4 pairs which arise by prolonging both sides in the above pairs with σ_3 we would get an $(n, 3)$-presentation.)

Convention. We will implicitly assume that the heads $(E_1, F_1), (E_2, F_2), \ldots, (E_{2^t}, F_{2^t})$ in an (n, t)-presentation are, in fact, always given by (E_1, F_1) and some $\sigma_1, \sigma_2, \ldots, \sigma_t$ where $\text{SUPP}(\sigma_j) \subseteq \text{SUP}$ (for $j = 1, 2, \ldots, t$).

The above inductive definition easily yields (using Prop.11):

Proposition 33. *Given an (n, t)-presentation with* SUP*, for any head pair (E_j, F_j), $1 \leq j \leq 2^t$, there is some $\overline{\sigma}$,* $\text{SUPP}(\overline{\sigma}) \subseteq \text{SUP}$*, such that $(E_j, F_j) = (E_1\overline{\sigma}, F_1\overline{\sigma})$; hence* $\text{EQLV}(E_j, F_j) \geq \text{EQLV}(E_1, F_1)$.

Def.34 and Prop.35 serve for showing (an inductive proof of) Lemma 36. The essence is simple: in an eq-level decreasing $(n, t{+}1)$-sequence, (E_1, F_1) yields x_i, H as in Prop.12, and we can be replacing σ with $\{(x_i, H')\}\sigma_{[-x_i]}$ as in Prop.13, and more generally $\sigma'\sigma$ with $\{(x_i, H')\}(\sigma'\{(x_i, H')\})_{[-x_i]}\sigma_{[-x_i]}$, which in later pairs does not affect the eq-levels due to Prop.31. This allows to create an eq-level decreasing $(n{-}1, t)$-sequence from the original even pairs. We now formalize this.

Definition 34. *Let $\mathcal{P}_1$ and $\mathcal{P}$ be as in Def.32. (Hence $\mathcal{P}$ is an $(n, t{+}1)$ presentation with heads (10); $\mathcal{P}_1$ is its first half.)*

Let $x_i \in \text{SUP}$ (so $n > 0$) and $H \neq x_i$; we put

$$H' = H\{(x_i, H)\}\{(x_i, H)\} \ldots, \text{ and}$$
$$\sigma'' = \sigma'\{(x_i, H')\}.$$

Then the presentation induced by $\mathcal{P}$ and x_i, H *is the $(n{-}1, t)$-presentation with* $\text{SUP}' = \text{SUP} \setminus \{x_i\}$ *where the tails are $\sigma_{[-x_i]}$, the heads are $(E'_1, F'_1), (E'_2, F'_2), \ldots, (E'_{2^t}, F'_{2^t})$, and*

- *if $t = 0$ then $(E'_1, F'_1) = (E_1\sigma'', F_1\sigma'')$, and*
- *if $t \geq 1$ then $(E'_1, F'_1), \ldots, (E'_{2^{t-1}}, F'_{2^{t-1}})$ is the head sequence of the $(n{-}1, t{-}1)$-presentation induced by $\mathcal{P}_1$, x_i, H, and for $j = 1, 2 \ldots, 2^{t-1}$ we have $(E'_{2^{t-1}+j}, F'_{2^{t-1}+j}) = (E'_j\sigma''_{[-x_i]}, F'_j\sigma''_{[-x_i]})$.*

E.g., for (11) we get $E'_1 = E_1\sigma_1\{(x_i, H')\}$ and $E'_2 = E_1\sigma_1\{(x_i, H')\}(\sigma_2\{(x_i, H')\})_{[-x_i]}$.
If we make an $(n, 3)$-presentation from (11) by using $\sigma' = \sigma_3$ then in the induced sequence we would get $E'_3 = E_1\sigma_1\{(x_i, H')\}(\sigma_3\{(x_i, H')\})_{[-x_i]}$ and $E'_4 = E_1\sigma_1\{(x_i, H')\}(\sigma_2\{(x_i, H')\})_{[-x_i]}(\sigma_3\{(x_i, H')\})_{[-x_i]}$.

Proposition 35. *Assume an eq-level decreasing $(n, t{+}1)$-sequence ($t \geq 0$); then $n > 0$. Let the eq-levels of the pairs in the presented sequence be $\ell_1 > \ell_2 > \cdots > \ell_{2^{t+1}}$. Then there is an eq-level decreasing $(n{-}1, t)$-sequence, with eq-levels $\ell_2, \ell_4, \ell_6, \ldots, \ell_{2^{t+1}}$.*

Proof: Consider the assumed eq-level decreasing $(n, t{+}1)$-sequence in presentation $\mathcal{P}$ as in Def. 32 and 34. For any σ' where $\text{SUPP}(\sigma') = \emptyset$ we have $(E_1\sigma', F_1\sigma') = (E_1, F_1)$; hence there are obviously no eq-level decreasing $(0, 1)$-sequences, and our assumption indeed implies $n > 0$.

Since $(E_2, F_2) = (E_1\sigma_1, F_1\sigma_1)$ (for some σ_1, $\text{SUPP}(\sigma_1) \subseteq \text{SUP}$) and $\ell_1 > \ell_2 = \text{EQLV}(E_2\sigma, F_2\sigma) \geq \text{EQLV}(E_2, F_2) \geq \text{EQLV}(E_1, F_1)$ (recall Prop.11), we have $\ell_1 > k = \text{EQLV}(E_1, F_1)$. By Prop.12 and 13 we can fix some x_i, H such that $H \neq x_i$, $x_i \in \text{SUPP}(\sigma) \subseteq \text{SUP}$ and $\sigma \sim_{\ell_1 - k} \{(x_i, H')\}\sigma = \{(x_i, H')\}\sigma_{[-x_i]}$ (where $H' = H\{(x_i, H)\}\{(x_i, H)\}\dots$).

We now show, by induction on t, that the $(n{-}1, t)$-*presentation induced by* $\mathcal{P}$ *and* x_i, H presents an $(n{-}1, t)$-sequence which we search.

First note that for any $j \geq 2$ we have $\sigma \sim_{\ell_j - k + 1} \{(x_i, H')\}\sigma_{[-x_i]}$ (since $\ell_1 > \ell_j$). By Prop.33 and 31, we can replace the tails σ with $\{(x_i, H')\}\sigma_{[-x_i]}$ in all pairs but the first one, in particular in all second-half pairs, *without affecting the eq-levels* ($\ell_{2^t+1} > \ell_{2^t+2} > \dots > \ell_{2^{t+1}}$).

If $t = 0$ then the second pair $(E_1\sigma'\sigma, F_1\sigma'\sigma)$ thus yields $(E_1\sigma'\{(x_i, H')\}\sigma_{[-x_i]}, F_1\sigma'\{(x_i, H')\}\sigma_{[-x_i]})$.
We are done when recalling that the induced $(n{-}1, 0)$-presentation is given by tails $\sigma_{[-x_i]}$ and the head-pair $(E_1', F_1') = (E_1\sigma'\{(x_i, H')\}, F_1\sigma'\{(x_i, H')\})$.

If $t \geq 1$ then we first use the induction hypothesis for the first half, i.e. for the (n, t)-presentation $\mathcal{P}_1$; we get that the $(n{-}1, t{-}1)$-presentation induced by $\mathcal{P}_1$, x_i, H, i.e. $\sigma_{[-x_i]}$ and $(E_1', F_1'), \dots, (E_{2^{t-1}}', F_{2^{t-1}}')$, presents a sequence with eq-levels $\ell_2, \ell_4, \ell_6, \dots, \ell_{2^t}$.

Then we use the induction hypothesis for the modified second half, namely for the (n, t)-presentation with the same heads as in $\mathcal{P}_1$, i.e. $(E_1, F_1), \dots, (E_{2^t}, F_{2^t})$, but with tails $\sigma'\{(x_i, H')\}\sigma_{[-x_i]}$ (which replaced the original $\sigma'\sigma$ without affecting the eq-levels $\ell_{2^t+1}, \ell_{2^t+2}, \dots, \ell_{2^{t+1}}$). The induced $(n{-}1, t{-}1)$-presentation has the same heads as in the first-half case, namely $(E_1', F_1'), \dots, (E_{2^{t-1}}', F_{2^{t-1}}')$, but the tails are $(\sigma'\{(x_i, H')\}\sigma_{[-x_i]})_{[-x_i]} = (\sigma'\{(x_i, H')\})_{[-x_i]}\sigma_{[-x_i]}$. Now $\ell_{2^t+2}, \ell_{2^t+4}, \ell_{2^t+6}, \dots, \ell_{2^{t+1}}$ are the eq-levels of the presented sequence.

Since the two $(n{-}1, t{-}1)$-presentations, arising above by using the induction hypothesis twice, yield together the $(n{-}1, t)$-presentation induced by $\mathcal{P}$, x_i, H, we are done. ■

Prop.35 implies (by induction on n):

Lemma 36. *There are no eq-level decreasing $(n, n{+}1)$ sequences (for any n).*

C. Bounding shortest nonequivalence witnesses

Let us consider a play of a simplified version of the Prover-Refuter game. We assume that a det-first-order grammar $\mathcal{G}$ and a pair $T_0 \not\sim U_0$ are given, and the play starts from 5a (there is no basis since this is irrelevant now). We suppose that Refuter always chooses a least eq-level pair (so that his claims in 5b are true) and Prover follows the balancing strategy; Prover thus loses in the phase i where $i = \text{EQLV}(T_0, U_0)\, div\, M_1$. Our aim is to derive a bound (a function of the size of $\mathcal{G}$ and T_0, U_0) on $\text{EQLV}(T_0, U_0)$, a bound in the form in Theorem 29, called a *tetration bound* for short.

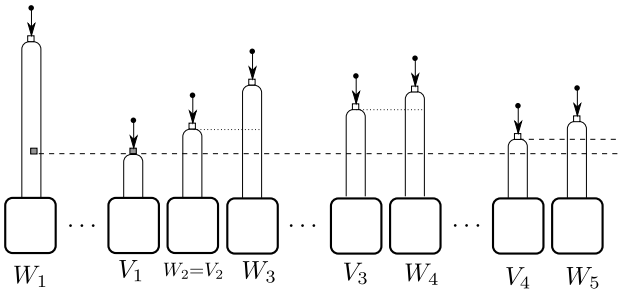

Fig. 17. $(1, 2, 3)$ and $(1, 4)$ are examples of stair sequences

We note easily that it suffices to derive a tetration bound on $\text{PRESSIZE}(T_j, U_j)$ for the pairs $(T_1, U_1), (T_2, U_2), \dots$ of the eq-level decreasing sequence (2) arising in the phases $0, 1, \dots$. Since we have no repeat, a bound on $\text{PRESSIZE}(T_j, U_j)$ yields an elementarily bigger bound on $\text{EQLV}(T_0, U_0)$.

If there is no balancing (since all relevant paths are M_0-sinking) then $\text{PRESSIZE}(T_j, U_j)$ is bounded by $\text{PRESSIZE}(T_0, U_0)$ increased with some elementary number. (Recall our convention that elementary numbers are bounded by elementary functions of the size of $\mathcal{G}$.)

We now suppose $k \geq 1$ balancing steps and write the finite pivot path (8) $W_1 \xrightarrow{v_1} W_2 \xrightarrow{v_2} \cdots$ as

$$W_1 \xrightarrow{v_{11}} V_1 \xrightarrow{v_{12}} W_2 \xrightarrow{v_{21}} V_2 \xrightarrow{w_{22}} W_3 \xrightarrow{v_{31}} \cdots V_{k-1} \xrightarrow{v_{k-1,2}} W_k \tag{12}$$

where v_{j1} is the longest prefix of v_j such that $W_j \xrightarrow{v_{j1}}$ sinks (into some depth in $\text{DOM}(W_j)$). We can have cases $v_{j1} = \varepsilon$ as well as cases $v_{j1} = v_j$. We also put $V_0 = W_1$ and $v_{02} = \varepsilon$. Fig.17 stresses that $V_j \xrightarrow{v_{j2}} W_{j+1}$ is strongly root-performable. Recall also that V_j is somewhere in the path depicted in Fig.14 but V there served for different aims.

Proposition 37. *If $V_j = (\text{TOP}_{M_1}^{V_j})\,\sigma$ then the bal-result related to the pivot W_{j+1} can be written $(E\sigma, F\sigma)$ where E, F are finite terms with depth bounded by $1 + M_1 + (3M_1 + 2M_0) \cdot \text{STEPDEPTHINC}$. Moreover, $\text{SUPP}(\sigma) \subseteq \{x_1, x_2, \dots, x_n\}$ where n is (the elementary number) defined by (9).*

Proof: Since $V_j \xrightarrow{v_{j2}} W_{j+1}$ is strongly root-performable, we have $V_j = (Yx_1x_2\dots x_m)\sigma'$, $W_{j+1} = H\sigma'$ where $Yx_1x_2\dots x_m \xrightarrow{v_{j2}} H$ and $\text{DEPTH}(H) \leq 1 + 2M_1 \cdot \text{STEPDEPTHINC}$ (recall Fig.14). The rest can be checked by recalling the balancing results (Fig.10 and 11) and Def.27. ■

It is easy to check that for getting a tetration bound on the size of T_j, U_j it suffices to get a tetration bound on $\text{DEPTH}(G)$ in strongly root-performable subpaths of (12) in the form $(Yx_1\dots x_m)\sigma \xrightarrow{u} G\sigma$ where $(Yx_1\dots x_m) \xrightarrow{u} G$. We now formalize that a large $\text{DEPTH}(G)$ can be only achieved by long "stair-sequences" of some V_j's, created from elementarily bounded "stairs"; if these stair-sequences were longer than a tetration bound then we would contradict Lemma 36.

Definition 38. *A pair (i,j), where $0 \le i < j \le k-1$ (k taken from (12)), is* a stair *if $V_i \xrightarrow{v_{i2}} \xrightarrow{v_{i+1}v_{i+2}} \cdots \xrightarrow{v_{j-1}} \xrightarrow{v_{j1}} V_j$ is strongly root-performable. A* stair (i,j) *is* simple *if there is no j', $i < j' < j$, such that both (i,j') and (j',j) are stairs.*

A subsequence $i_1, i_2, \ldots, i_r$, of $0,1,2,\ldots,k-1$ is a (simple) stair sequence *if (i_j, i_{j+1}) is a (simple) stair for each $j \in \{1,2,\ldots,r-1\}$.*

In Fig.17, examples of stairs are $(1,2)$, $(1,3)$, $(1,4)$, where $(1,2)$ and $(1,4)$ are simple stairs but $(1,3)$ is not simple. $1,2,3$ and $1,4$ are examples of simple stair-sequences.

The next proposition captures the fact that a simple stair represents just an "elementarily bounded increase".

A *substitution* σ is *finite* if $\sigma(x_i)$ is finite for all $x_i \in \text{VAR}$; $\text{DEPTH}(\sigma)$ is then $\max\{\text{DEPTH}(\sigma(x_i) \mid x_i \in \text{VAR}\}$.

Proposition 39. *If (i,j) is a simple stair and $V_i = (\text{TOP}^{V_i}_{M_1})\,\sigma$ then $V_j = (\text{TOP}^{V_j}_{M_1})\,\sigma'\sigma$ where σ' is finite and $\text{DEPTH}(\sigma') \le 2M_1 \cdot \text{STEPDEPTHINC}$. Moreover, $\text{SUPP}(\sigma') \subseteq \{x_1, x_2, \ldots, x_n\}$, n defined by (9); we can have $\text{SUPP}(\sigma') = \emptyset$.*

Proof: Suppose (i,j) is a simple stair and we have $V_i = (Yx_1 \ldots x_m)\sigma''\sigma$ where $(Yx_1 \ldots x_m)\sigma'' = \text{TOP}^{V_i}_{M_1}$. Then $(i, i+\ell)$ is a simple stair for any $\ell, 1 \le \ell \le j-i$, and we have $(Yx_1 \ldots x_m) \xrightarrow{v_{i2}} \xrightarrow{v_{i+1}} \cdots \xrightarrow{v_{i+\ell-1}} \xrightarrow{v_{i+\ell,1}} G_{i+\ell}$ where $V_{i+\ell} = G_{i+\ell}\sigma''\sigma$. For any $\ell, 1 \le \ell < j-i$, the path $G_{i+\ell} \xrightarrow{v_{i+\ell,2}} \xrightarrow{v_{i+\ell+1}} \cdots \xrightarrow{v_{j-1}} \xrightarrow{v_{j,1}} G_j$ is not strongly root-performable (since (i,j) is a simple stair), which by our definitions means that there is ℓ', $\ell < \ell' \le j-i$ such that $G_{i+\ell'}$ is a (proper) subterm of $G_{i+\ell}$. Repeating this reasoning for ℓ', etc., we deduce that G_j is a subterm of G_{i+1}, and thus $\text{DEPTH}(G_j) \le \text{DEPTH}(G_{i+1})$.

We can easily check $\text{DEPTH}(G_{i+1}) \le 1 + 2M_1 \cdot \text{STEPDEPTHINC}$, hence in $V_j = G_j\sigma''\sigma$ we have $\text{DEPTH}(G_j\sigma'') \le M_1 + 2M_1 \cdot \text{STEPDEPTHINC}$ and for any $\gamma \in \text{DOM}(G_j\sigma'')$ where $(G_j\sigma'')(\gamma) = x_i \in \text{SUPP}(\sigma)$ we have $|\gamma| \ge M_1$. Hence $V_j = (\text{TOP}^{V_j}_{M_1})\,\sigma'\sigma$ where σ' is finite and $\text{DEPTH}(\sigma') \le 2M_1 \cdot \text{STEPDEPTHINC}$. ∎

Proposition 40. *Assume $(Yx_1 \ldots x_m) \xrightarrow{u} G \notin \text{VAR}$ and $(Yx_1 \ldots x_m)\sigma \xrightarrow{u} G\sigma$ is a (strongly root-performable) subpath of (12). Then this subpath contains $V_{i_1}, V_{i_2}, \ldots, V_{i_r}$ for a simple stair-sequence $i_1, i_2, \ldots, i_r$ with $r = (\text{DEPTH}(G)\,div\,\text{STEP}) - 1$ where $\text{STEP} = (2M_1 + M_0) \cdot \text{STEPDEPTHINC}$.*

Proof: For any V_i *inside*, by which we mean inside the path $(Yx_1 \ldots x_m)\sigma \xrightarrow{u} G\sigma$, there is G_i such that $V_i = G_i\sigma$ and $(Yx_1 \ldots x_m) \xrightarrow{u'} G_i \xrightarrow{u''} G$ where $u'u'' = u$. If there is no V_i inside then obviously $\text{DEPTH}(G) < \text{STEP}$ (recall Fig.14). Let i' be such that $V_{i'}$ is inside and $\text{DEPTH}(G_{i'})$ is minimal; surely $\text{DEPTH}(G_{i'}) \le \text{STEP}$. Let i'' be the largest index such that $V_{i''}$ is inside; surely $\text{DEPTH}(G) \le \text{DEPTH}(G_{i''}) + \text{STEP}$. There must be a simple stair-sequence $i_1, i_2, \ldots, i_r$ such that $i_1 = i'$ and $i_r = i''$ (since either $i' = i''$, in which case $r = 1$, or (i', i'') is a stair).

Recalling Prop.39, we get $\text{DEPTH}(G_{i_r}) \le \text{DEPTH}(G_{i_1}) + (r-1) \cdot \text{STEP}$, and thus $\text{DEPTH}(G) \le (r+1) \cdot \text{STEP}$. ∎

Lemma 41. *There are elementary numbers $n, q \in \mathbb{N}$ (given by elementary functions of $size(\mathcal{G})$) with the following property: For any simple stair-sequence $(i_1, i_2, \ldots, i_r)$ in (12), of length $r = h(t)$ where the (nonelementary) function h is defined by the following recursive definition*

$$h(1) = q + 1 \text{ and } h(j+1) = h(j) \cdot (1 + q^{h(j)}),$$

the sequence of the balancing results related to pivots $W_{i_1+1}, W_{i_2+1}, \ldots, W_{i_r+1}$, contains an (n,t)-subsequence.

Proof: Let n be defined by (9) in the proof of Lemma 28, and let q be the number of possible pairs $(E\sigma', F\sigma')$ where (E,F) are finite terms arising as in Prop.37 and σ' is a finite substitution, $\text{SUPP}(\sigma') \subseteq \{x_1, x_2, \ldots, x_n\}$, arising as in Prop.39. These numbers n, q are obviously elementary.

For a simple stair-sequence $(i_1, i_2, \ldots, i_r)$ we present $V_{i_1} = (\text{TOP}^{V_{i_1}}_{M_1})\,\sigma$, $V_{i_2} = (\text{TOP}^{V_{i_2}}_{M_1})\,\sigma_1\sigma$, $V_{i_3} = (\text{TOP}^{V_{i_3}}_{M_1})\,\sigma_2\sigma_1\sigma$, …, $V_{i_r} = (\text{TOP}^{V_{i_r}}_{M_1})\,\sigma_{r-1}\sigma_{r-2}\ldots\sigma_1\sigma$, where σ_j are finite and bounded as σ' in Prop.39. Moreover, the supports of σ and σ_j ($j = 1,2,\ldots,r-1$) are subsets of $\text{SUP} = \{x_1, x_2, \ldots, x_n\}$.

The balancing results corresponding to $V_{i_1}, V_{i_2}, \ldots, V_{i_r}$ as in Prop.37 can be written as follows:
$(E_1\sigma, F_1\sigma)$, $(E_2\sigma_1\sigma, F_2\sigma_1\sigma)$, $(E_3\sigma_2\sigma_1\sigma, F_3\sigma_2\sigma_1\sigma)$, …, $(E_r\sigma_{r-1}\sigma_{r-2}\ldots\sigma_1\sigma, F_r\sigma_{r-1}\sigma_{r-2}\ldots\sigma_1\sigma)$.

We now establish the claim by induction on t, using the pigeonhole principle. If $t = 1$ then $r = q + 1$ and we necessarily get $(E_i, F_i) = (E_j, F_j)$ for some $i < j$; hence the i-th pair and the j-th pair constitute an $(n,1)$-sequence: the tails are $\overline{\sigma} = \sigma_{i-1}\sigma_{i-2}\ldots\sigma_1\sigma$ and the heads are (E_i, F_i), $(E_i\sigma', F_i\sigma')$ where $\sigma' = \sigma_{j-1}\sigma_{j-2}\ldots\sigma_i$.

As the induction hypothesis we take that for $r = h(t)$ we are guaranteed that among the above bal-results there are 2^t pairs, with indices $j_1 < j_2 < \cdots < j_{2^t}$, which constitute an (n,t)-presentation with tails $\overline{\sigma} = \sigma_{j_1-1}\sigma_{j_1-2}\ldots\sigma_1\sigma$. When considering $r = h(t+1)$, we partition $(i_1, i_2, \ldots, i_r)$, and the coresponding bal-results, into $1 + q^{h(t)}$ segments of length $h(t)$. The i-th segment induces the sequence (E_{s+1}, F_{s+1}), $(E_{s+2}\sigma_{s+1}, F_{s+2}\sigma_{s+1})$, $(E_{s+3}\sigma_{s+2}, F_{s+3}\sigma_{s+2})$, …, $(E_{s+h(t)}\sigma_{s+h(t)-1}, F_{s+h(t)}\sigma_{s+h(t)-1})$ where $s = (i-1)\cdot h(t)$; the number of possible induced sequences is surely bounded by $q^{h(t)}$. Hence there are $i < j$ such that the i-th segment and the j-th segment induce the same sequence. By the induction hypothesis, the i-th segment gives rise to an (n,t)-presentation with some tails $\overline{\sigma}$, and the j-th segment gives rise to an (n,t)-presentation with the same heads but with tails $\sigma'\overline{\sigma}$ (for some σ'). We thus get an $(n,t+1)$-presentation with tails $\overline{\sigma}$. ∎

It is a routine to verify that $h(t)$ can be bounded with $2 \uparrow\uparrow g(t)$ for an elementary function g. Hence $h(n+1)$ (n being the elementary number (9)) yields a tetration bound on the length of (simple) stair sequences in (12). We thus have a tetration bound on $\text{DEPTH}(G)$ in Prop.40; this yields a tetration bound on $\text{PRESSIZE}(T_j, U_j)$, and thus a tetration bound on $\text{EQLV}(T_0, U_0)$. We have finished a proof of Theorem 29.

VI. BISIMILARITY FOR FIRST-ORDER GRAMMARS

Trace equivalence coincides with bisimulation equivalence (bisimilarity) in deterministic LTSs. For (general) first-order grammars trace equivalence is undecidable (like language equivalence for nondeterministic pushdown automata) but bisimilarity is still decidable. This can be shown in principle in the way we used for det-first order grammars, with some modifications. Below we define bisimilarity and then discuss the respective modifications of Sections II, III, IV. We also note why the way of deriving the complexity bound in Section V does not apply in the general case.

Bisimulation equivalence (bisimilarity)

We recall a variant of the standard definition of bisimulation equivalence, denoted $\sim$, and of its strata $\sim_k$, $k \in \mathbb{N}$.

Given an LTS $\mathcal{L} = (\mathcal{S}, \mathcal{A}, (\stackrel{a}{\longrightarrow})_{a\in\mathcal{A}})$, we say that $B \subseteq \mathcal{S} \times \mathcal{S}$ *covers* $(s,t) \in \mathcal{S} \times \mathcal{S}$ if for any $s \stackrel{a}{\longrightarrow} s'$ there is $t \stackrel{a}{\longrightarrow} t'$ such that $(s',t') \in B$, and for any $t \stackrel{a}{\longrightarrow} t'$ there is $s \stackrel{a}{\longrightarrow} s'$ such that $(s',t') \in B$. B *covers* $B' \subseteq \mathcal{S} \times \mathcal{S}$ if B covers each $(s,t) \in B'$. B is a *bisimulation* if B covers B. States s,t are *bisimilar*, $s \sim t$, if there is a bisimulation B containing (s,t).

We put $\sim_0 = \mathcal{S} \times \mathcal{S}$. For $k \geq 1$, $\sim_k \subseteq \mathcal{S} \times \mathcal{S}$ is the set of all pairs covered by $\sim_{k-1}$. Note that $s \not\sim_1 t$ iff s,t enable different sets of actions iff there is no B which covers (s,t). If s,t are dead, not enabling any transition, then $\emptyset$ covers (s,t).

We can easily check that if $\mathcal{L}$ is a det-LTS then $\sim$ and $\sim_k$ coincide with the trace equivalence and its strata (defined in Section II), but bisimilarity is finer than trace equivalence for general LTSs which we consider now.

We first note that Prop.1 holds for our new $\sim_k$ and $\sim$ if we restrict ourselves to *image-finite LTSs* to get $\cap_{k\in\mathbb{N}} \sim_k = \sim$; i.e., we assume that for each $s \in \mathcal{S}$ and each $a \in \mathcal{A}$ there are only finitely many s' such that $s \stackrel{a}{\longrightarrow} s'$ (in which case $\cap_{k\in\mathbb{N}} \sim_k$ is a bisimulation). Note that $\mathcal{L}^{\rm A}_{\mathcal{G}}$ is image-finite for any first-order grammar $\mathcal{G}$.

We define $\mathrm{EQLV}(s,t)$ in the same way as after Prop.1 (now with respect to the new $\sim_k$, $\sim$) and note that Prop.2 keeps holding. Prop.4 does not hold for general LTSs, the eq-level can drop by more than 1 in one step, but we note:

Proposition 42. *(1) If $s \sim_1 t$ then there is B which covers (s,t) and $\mathrm{EQLV}(s',t') \geq \mathrm{EQLV}(s,t) - 1$ for all $(s',t') \in B$. (2) If $s \not\sim t$ and B covers (s,t) then there is $(s',t') \in B$ such that $\mathrm{EQLV}(s',t') < \mathrm{EQLV}(s,t)$.*

We now consider the problem BISIM-EQ-G:

Input: a first-order grammar $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$, and (graph presentations of) two input terms T_{in}, U_{in}.
Question: is $T_{in} \sim U_{in}$ in $\mathcal{L}^{\rm A}_{\mathcal{G}}$? ($\sim$ denotes bisimilarity)

The bisimilarity problem in the LTSs generated by (nondeterministic) pushdown automata can be reduced to BISIM-EQ-G in the way sketched in Sec.II on condition that ε-steps are popping and deterministic, i.e. if also here a rule $[q_2C](x_1,x_2,x_3) \stackrel{\varepsilon}{\longrightarrow} ..$ excludes the existence of another rule $[q_2C](x_1,x_2,x_3) \longrightarrow ...$ (As shown in [9], with popping but nondeterministic ε-steps bisimilarity becomes undecidable.)

An analogue of Lemma 9 for BISIM-EQ-G holds as well; this follows from the (inductive) observation that $\sim_k$ is decidable, for any $k \in \mathbb{N}$. (Recall that $\mathcal{L}^{\rm A}_{\mathcal{G}}$ is image finite and $\mathcal{A}$ is finite.)

We now introduce a notion of *k-distance covers* related to $\mathrm{REG}(T,U,k)$ from Subsec.III-B, in the setting of a given LTS $\mathcal{L} = (\mathcal{S}, \mathcal{A}, (\stackrel{a}{\longrightarrow})_{a\in\mathcal{A}})$. We say that B is a *minimal cover of* B' if B covers B' but no proper subset of B covers B'.

A mapping

$\mathcal{C} : \{0,1,\ldots,k\} \to 2^{\mathcal{S}\times\mathcal{S}}$ is a *k-distance cover of* (s,t) if $\mathcal{C}(0) = \{(s,t)\}$ and
$\mathcal{C}(j)$ is a minimal cover of $\mathcal{C}(j-1)$, for $j = 1,2,\ldots,k$.

Remark. A k-distance cover corresponds to a strategy for Defender (which might still leave some free choice) in the standard bisimulation game between Attacker and Defender; this strategy guarantees that Defender will not lose within k rounds.

We easily observe the next facts:

Proposition 43. *(1) If $\mathcal{C}$ is a k-distance cover of (s,t) then $s' \sim_1 t'$ for all $(s',t') \in \bigcup_{0\leq j\leq k-1} \mathcal{C}(j)$.*
(2) $s \sim_k t$ iff there is a k-distance cover of (s,t).

Proposition 44. *Assume $\mathcal{C}$ is a k-distance cover of (E,F) in $\mathcal{L}^{\rm A}_{\mathcal{G}}$ for a given first-order grammar $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$. (Recall Def.6, Prop.7 and Prop.3.) Then we have:*

1) *If $(E',F') \in \mathcal{C}(j)$ then there are $u, u' \in \mathcal{R}^*$ where $|u| = |u'| = j$ such that $E \stackrel{u}{\longrightarrow} E'$ and $F \stackrel{u'}{\longrightarrow} F'$ in $\mathcal{L}^{\rm R}_{\mathcal{G}}$.*
2) *If $(E',F') \in \mathcal{C}(j_1)$ and $u = r_1 r_2 \ldots r_{j_2} \in \mathcal{R}^*$, $j_1 + j_2 \leq k$, and $E' \stackrel{u}{\longrightarrow} E''$ then there is $u' = r'_1 r'_2 \ldots r'_{j_2} \in \mathcal{R}^*$ such that $\mathrm{ACT}(r_i) = \mathrm{ACT}(r'_i)$ for $i = 1,2,\ldots,j_2$, $F' \stackrel{u'}{\longrightarrow} F''$, and $(E'',F'') \in \mathcal{C}(j_1 + j_2)$. (Symmetrically for $F' \stackrel{u}{\longrightarrow} F''$.)*

Though the eq-level drop (in Prop.42(2)) can be more than 1 in one step, we still have the following analogue of Prop.14 (which follows from Prop.42 and Prop.43).

Proposition 45. *:*
(1) If $s \sim t$ then for each $k \in \mathbb{N}$ there is a k-distance cover $\mathcal{C}$ of (s,t) such that $s' \sim t'$ for all $(s',t') \in \bigcup_{0\leq j\leq k} \mathcal{C}(j)$.
(2) If $s \not\sim t$ and $\mathcal{C}$ is a k-distance cover for (s,t) then any least eq-level pair in $\bigcup_{0\leq j\leq k} \mathcal{C}(j)$ is in $\mathcal{C}(k) \setminus \bigcup_{0\leq j\leq k-1} \mathcal{C}(j)$.

We note that each pair (s,s) has a k-distance cover $\mathcal{C}$ where $\mathcal{C}(j) = \{(s',s') \mid s \stackrel{w}{\longrightarrow} s', |w| = j\}$. Considering now $\mathcal{L}^{\rm A}_{\mathcal{G}}$ for a first-order grammar $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$, it is straightforward to verify Prop.11. Prop.12 is modified as follows:

Proposition 46. *If $\mathrm{EQLV}(E,F) = k < \ell = \mathrm{EQLV}(E\sigma, F\sigma)$ (where $\ell \in \mathbb{N} \cup \{\omega\}$) then there are $x_i \in \mathrm{SUPP}(\sigma)$, $H \neq x_i$, and a word $w \in \mathcal{A}^*$, $|w| \leq k$, such that $E \stackrel{w}{\longrightarrow} x_i$, $F \stackrel{w}{\longrightarrow} H$ or $E \stackrel{w}{\longrightarrow} H$, $F \stackrel{w}{\longrightarrow} x_i$, and $\sigma(x_i) \sim_{\ell-k} H\sigma$.*

Proof: We consider E, F, σ in the assumption and a $(k+1)$-cover $\mathcal{C}$ of $(E\sigma, F\sigma)$ such that $G' \sim_{\ell-j} G''$ for all

$(G', G'') \in \mathcal{C}(j)$, $0 \leq j \leq k{+}1$; its existence follows from Prop.42(1). If there is no $(G', G'') \in \bigcup_{0\leq j\leq k} \mathcal{C}(j)$ where $\{G', G''\} = \{\sigma(x_i), H\sigma\}$ and x_i, H satisfy the claim (for some w, $|w| = j \leq k$, where $(G', G'') \in \mathcal{C}_j$) then we can obviously use $\mathcal{C}$ to construct a $(k{+}1)$-cover of (E, F), which is impossible. ∎

Remark. A difference wrt the deterministic case is that for $\sigma_1 \neq \sigma_2$ such that $\text{EQLV}(E, F) = k$ and $k < \ell_1 = \text{EQLV}(E\sigma_1, F\sigma_1)$, $k < \ell_2 = \text{EQLV}(E\sigma_2, F\sigma_2)$ we can have that the pairs x_i, H satisfying the claim for σ_1 might differ from those satisfying the claim for σ_2 (even if $\ell_1 = \ell_2$). Another difference is that we might get a case x_i, H, w with $|w| < k$ for any $(k{+}1)$-cover described in the proof of Prop.46; that's why Prop.31 does not hold here. The mentioned differences cause that the complexity argument from Sec.V does not go through for the problem BISIM-EQ-G.

We can easily verify that Prop.13 keeps holding (for bisimilarity on general first-order grammars). Prop.15 is just reformulated as follows (recall also Prop.45, the analogue of Prop.14):

Proposition 47. *Suppose $\mathcal{C}$ is a k-distance cover of (T, U) and for σ, σ' we have $\text{SUPP}(\sigma) = \text{SUPP}(\sigma')$ and $(\sigma(x_i), \sigma'(x_i)) \in \bigcup_{0\leq j\leq k-1} \mathcal{C}(j)$ for each $x_i \in \text{SUPP}(\sigma)$.*
If $\text{EQLV}(T', U') = \text{MINEL}(\bigcup_{0\leq j\leq k} \mathcal{C}(j))$ and $T' = G\sigma$ then $\text{EQLV}(G\sigma', U') = \text{EQLV}(T', U')$.

We now describe the Prover-Refuter game in the more general setting of bisimilarity for (general) first-order grammars.

PROVER-REFUTER GAME (for bisimilarity)

1) A first-order grammar $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ is given.
2) Prover produces (by "guessing", say) a finite set BASIS of pairs of (graph presentations of regular) terms.
3) An input pair (T_{in}, U_{in}) is given.
4) Refuter chooses $(T_0, U_0) \in \text{STARTSET} = \{(T_{in}, U_{in})\} \cup \text{BASIS}$, and claims $\text{EQLV}(T_0, U_0) = \text{MINEL}(\text{STARTSET}) < \omega$.
5) For $i = 0, 1, 2, \ldots$, Phase i is performed, i.e.:
   a) Prover chooses $k > 0$ and presents a k-distance cover $\mathcal{C}_i$ of (T_i, U_i); if she is unable, she loses.
   b) Refuter chooses $(T_i', U_i') \in \mathcal{C}_i(k) \setminus \bigcup_{0\leq j\leq k-1} \mathcal{C}_i(j)$ and $w_i' = r_1' r_2' \ldots r_k' \in \mathcal{R}^*$, $w_i'' = r_1'' r_2'' \ldots r_k'' \in \mathcal{R}^*$ so that for each $j, 1 \leq j \leq k$, we have $\text{ACT}(r_j') = \text{ACT}(r_j'')$, $T_i \xrightarrow{r_1' r_2' \ldots r_j'} T_{ij}$, $U_i \xrightarrow{r_1'' r_2'' \ldots r_j''} U_{ij}$ where $(T_{ij}, U_{ij}) \in \mathcal{C}_i(j)$; moreover, $(T_{ik}, U_{ik}) = (T_i', U_i')$. (Recall Prop.44.) If Refuter is unable to do this then Prover wins (we have $T_i \sim U_i$ due to dead terms).
   Refuter claims that $\text{EQLV}(T_i', U_i') = \text{MINEL}(\bigcup_{0\leq j\leq k} \mathcal{C}_i(j))$. (Recall Prop.45.)
   c) Prover produces (T_{i+1}, U_{i+1}) from (T_i', U_i') as follows:
      - either she puts $T_{i+1} = T_i'$, $U_{i+1} = U_i'$ (*no-change*),
      - or she *balances* (recall Prop.47 and Fig.11): if she finds σ, σ' such that $(\sigma(x_i), \sigma'(x_i)) \in \bigcup_{0\leq j\leq k-1} \mathcal{C}_i(j)$ for all $x_i \in \text{SUPP}(\sigma) = \text{SUPP}(\sigma')$, and she presents T_i' as $G\sigma$ then she can (do a *left-balancing*, namely) put $T_{i+1} = G\sigma'$, and $U_{i+1} = U_i'$; symmetrically, if U_i' is $G\sigma'$ then she can (do a *right-balancing*, namely) put $T_{i+1} = T_i'$, and $U_{i+1} = G\sigma$.

      (Thus $\text{EQLV}(T_{i+1}, U_{i+1}) = \text{EQLV}(T_i', U_i')$ if Refuter's claim in 5b is true. If $T_i \sim U_i$ and Prover has chosen $\mathcal{C}_i$ as in Prop.45(1) then $T_{i+1} \sim U_{i+1}$.)
   d) Prover *either contradicts Refuter's claims* by presenting an algorithmically verifiable proof, in which case Prover wins, *or lets the play proceed* with Phase $i{+}1$.

Lemma 16 obviously holds in the general case as well; if we get completeness then we will finish a proof of the next theorem, generalizing Theorem 17:

Theorem 48. *Bisimulation equivalence for first-order grammars (i.e., the problem* BISIM-EQ-G*) is decidable.*

We have mentioned in Subsection IV-B that the notions like (Y, j)-sink words, $\text{SSW}(Y, j)$, etc. make better sense in the det-LTS $\mathcal{L}^{\text{R}}_{\mathcal{G}}$ when considering the general case where $\mathcal{L}^{\text{A}}_{\mathcal{G}}$ is not deterministic.

It is a routine to go through Subsection IV-B and verify that all reasoning holds when we refer to $\mathcal{L}^{\text{R}}_{\mathcal{G}}$; a path $T \xrightarrow{w} T'$ (like, e.g., in Prop.23) is thus meant to be the unique path in $\mathcal{L}^{\text{R}}_{\mathcal{G}}$ determined by T and $w \in \mathcal{R}^*$. Prop.24 is now reformulated as follows:

Proposition 49. *Let $\mathcal{C}$ be a k-distance cover of (T, U) and let $T \xrightarrow{w'} T'$, $U \xrightarrow{w''} U'$ ($w', w'' \in \mathcal{R}^*$) be as described in Point 5b of the P-R game (where $T = T_i$, $w' = w_i'$ etc.). Suppose there is no $(T', V) \in \bigcup_{0\leq j\leq k-1} \mathcal{C}(j)$ and $T \xrightarrow{w'} T'$ is not M_0-sinking. Take the last root-performable (sub)path of length M_0 in $T \xrightarrow{w'} T'$, i.e., take the longest u_1 such that $T \xrightarrow{u_1} (Ax_1 \ldots x_m)\sigma \xrightarrow{v} G'\sigma \xrightarrow{u_2} T'$ where $Ax_1 \ldots x_m \xrightarrow{v} G'$ and $|v| = M_0$. Then there is $(\sigma(x_j), V_j') \in \bigcup_{0\leq j\leq k-1} \mathcal{C}(j)$ for each $j, 1 \leq j \leq m$, and $G' \xrightarrow{u_2} G$ (hence $T' = G\sigma$) where $\text{DEPTH}(G) \leq 1 + (2M_0{-}1) \cdot \text{STEPDEPTHINC}$.*

The previous proof goes through (when recalling Prop.44).

Also the pivot path (associated to a play) can be now considered as a path in the det-LTS $\mathcal{L}^{\text{R}}_{\mathcal{G}}$. It is a routine to verify Lemma 28 for general first-order grammars if the balancing strategy also requires that Prover uses only equivalent pairs in each $\mathcal{C}_i$ when starting from $T_0 \sim U_0$; hence Prover always chooses from the M_1-distance covers guaranteed by Prop.45(1). (Prover does not demonstrate that she uses such covers, the strategy is non-effective in this sense).

It remains to check the analogue of Lemma 19. Now the reasoning becomes more sophisticated; the reason lies in the fact that Prop.46 in the general case is weaker than Prop.12 in the deterministic case (where Prop.4 helped).

Let us consider a play of the P-R game where a first-order grammar $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ is given in Point 1. For Point 2 we again suppose that Prover has an unlimited computational power, and also knows n, g for which she has an (n, g)-strategy for $\mathcal{G}$ (as defined by Def.18). The idea is that Prover

demonstrates a relative bound $\ell_{(n,g)}$ on the length of eq-level decreasing (n,g)-sequences (captured by Prop.50), which she can use to contradict Refuter's claims.

We imagine that Prover announces n, g in Point 2. In fact, we know that g induced by the balancing strategy is computable, so Prover can provide a Turing machine computing g though for our aims just a finite table with sufficiently many values $g(1), g(2), \ldots$ would suffice.

Now Prover chooses (sufficiently large) $\mathcal{B} \in \mathbb{N}$ and partitions the set $\{(E,F) \mid \textsc{PresSize}(E,F) \leq \mathcal{B}\}$ into two disjoint classes, putting

$$\{(E,F) \mid \textsc{PresSize}(E,F) \leq \mathcal{B}\} = \textsc{NonEq} \cup \textsc{GuessEq}$$

where for each $(E,F) \in \textsc{NonEq}$ she demonstrates $E \not\sim F$ by computing the finite $\textsc{EqLv}(E,F)$ (using the algorithm from an analogue of Lemma 9). The intention is that Prover puts only equivalent pairs in GUESSEQ but the following arguments also count with the possibility that some nonequivalent pairs were included in GUESSEQ. For any $b \leq \mathcal{B}$ we put

$$\textsc{MaxFEL}'_b = \max\{\, \textsc{EqLv}(E,F) \mid (E,F) \in \textsc{NonEq} \text{ and } \textsc{PresSize}(E,F) \leq b \,\}.$$

We thus have $\textsc{MaxFEL}'_b \leq \textsc{MaxFEL}_b$ (recall Def.18(5)); if $E \sim F$ for each $(E,F) \in \textsc{GuessEq}$ then $\textsc{MaxFEL}'_b = \textsc{MaxFEL}_b$.

Recalling the remark after the proof of Lemma 19, we now let Prover demonstrate the computation of $\ell_{(n,g)}$ using the following recursive definition

$$\text{if } n = 0 \text{ then } \ell_{(n,g)} = \ell_{(0,g)} = 1 + \textsc{MaxFEL}'_{g(1)}$$
$$\text{otherwise } \ell_{(n,g)} = \ell_{(n-1,g')} + (1 + \textsc{MaxFEL}'_{g(1)})$$
$$\text{where } g' \text{ is defined as in (5) but } \textsc{MaxFEL}_{g(1)} \text{ is replaced with } \textsc{MaxFEL}'_{g(1)}.$$

We denote $g = g_n$ and note that computing $\ell_{(n,g)} = \ell_{(n,g_n)}$ comprises computing $\ell_{(n-1,g_{n-1})}$, $\ell_{(n-2,g_{n-2})}$, $\ldots$, $\ell_{(0,g_0)}$ (by recursive calls) where for $i \in \{n-1, n-2, \ldots, 0\}$ we have

$$g_i(j) = g_{i+1}(1 + \textsc{MaxFEL}'_{g_{i+1}(1)} + j) + 2 \cdot (g_{i+1}(1) + \textsc{MaxFEL}'_{g_{i+1}(1)} \cdot \textsc{StepInc}).$$

During that computation, also the values $\textsc{MaxFEL}'_{g_n(1)}$, $\textsc{MaxFEL}'_{g_{n-1}(1)}$, $\ldots$, $\textsc{MaxFEL}'_{g_0(1)}$ are computed; this supposes that $g_i(1) \leq \mathcal{B}$ for all $i = n, n-1, \ldots, 0$. Since Prover knows n, g, she could have chosen $\mathcal{B}$ sufficiently large so that the computation is really performable and can be thus demonstrated.

Proposition 50. *The length of any eq-level decreasing (n,g)-sequence where the eq-level of the first pair is less than* $\textsc{MinEL}(\textsc{GuessEq})$ *is bounded by the above computed $\ell_{(n,g)}$.*

Proof: By induction on i we prove the claim for (i, g_i)-sequences and $\ell_{(i,g_i)}$, $i = 0, 1, \ldots, n$. We thus assume an (i,g_i)-presentation with tails σ ($\textsc{card}(\textsc{supp}(\sigma)) \leq i$) and heads (E_1,F_1), (E_2,F_2), $\ldots$, (E_ℓ,F_ℓ) (where $\textsc{PresSize}(E_j,F_j) \leq g_i(j)$ for $j = 1,2,\ldots,\ell$) for which the presented sequence is eq-level decreasing and $\textsc{MinEL}(\textsc{GuessEq}) > \textsc{EqLv}(E_1\sigma, F_1\sigma)$; recall that $\textsc{EqLv}(E_1\sigma, F_1\sigma) \geq \textsc{EqLv}(E_1,F_1)$. Hence $(E_1,F_1) \notin \textsc{GuessEq}$; since $\textsc{PresSize}(E_1,F_1) \leq g_i(1) \leq \mathcal{B}$, we have $(E_1,F_1) \in \textsc{NonEq}$, and thus $\textsc{EqLv}(E_1,F_1) \leq \textsc{MaxFEL}'_{g_i(1)}$. If $i = 0$ (so $(E_j\sigma, F_j\sigma) = (E_j,F_j)$) then necessarily $\ell \leq 1 + \textsc{MaxFEL}'_{g_0(1)}$.

If $i > 0$ and $\ell > 1 + \textsc{MaxFEL}'_{g_i(1)}$ then by Prop.46 there are x_i, H such that $x_i \in \textsc{supp}(\sigma)$, $H \neq x_i$, $\textsc{PresSize}(H) \leq \textsc{PresSize}(E_1,F_1) + \textsc{EqLv}(E_1,F_1) \cdot \textsc{StepInc} \leq g_i(1) + \textsc{MaxFEL}'_{g_i(1)} \cdot \textsc{StepInc}$, and $\sigma(x_i) \sim_t H\sigma$ where $t = \textsc{EqLv}(E_1\sigma, F_1\sigma) - \textsc{MaxFEL}'_{g_i(1)}$. (The *difference wrt the deterministic case*, where we had Prop.4, is that we have no direct means to demonstrate that $\textsc{MaxFEL}'_{g_i(1)}$ bounds the eq-level drop for any particular x_i, H; *just the existence of such x_i, H is guaranteed*.)

Using this guaranteed x_i, H, we could proceed as in the proof of Lemma 19 (recall Fig.13) and create the $(i-1, g_{i-1})$-sequence

$$(E'_1\sigma_{[-x_i]}, F'_1\sigma_{[-x_i]}), (E'_2\sigma_{[-x_i]}, F'_2\sigma_{[-x_i]}), \ldots$$

of length $\ell - (1 + \textsc{MaxFEL}'_{g_i(1)})$ where $E'_j = E_{s+j}\{(x_i, H')\}$, $F'_j = F_{s+j}\{(x_i,H')\}$, for $s = (1 + \textsc{MaxFEL}'_{g_i(1)})$ and $H' = H\{(x_i,H)\}\{(x_i,H)\}\{(x_i,H)\}\ldots$. This is an eq-level decreasing $(i-1, g_{i-1})$-sequence with the eq-level of the first pair surely less than $\textsc{MinEL}(\textsc{GuessEq})$, and by the induction hypothesis we can deduce $\ell \leq (1 + \textsc{MaxFEL}'_{g_i(1)}) + \ell_{(i-1,g_{i-1})}$. ∎

Therefore the mere demonstration of an (n,g)-subsequence of length $1 + \ell_{(n,g)}$ in the sequence $(T_1,U_1), (T_2,U_2), \ldots$ can serve as a contradiction for Refuter's claims, together with the demonstrated computation of $\ell_{(n,g)}$ based on $\textsc{GuessEq}$ which is viewed as BASIS.

VII. CONCLUSION

It seems technically difficult to compare the proof(s) presented here with the previous proofs in detail. The abstract idea is surely the same, and can be captured as follows. If two states (configurations, terms) are nonequivalent then there is a "shortest" finite witness of this fact; let us think of a finite word in the deterministic case. Such a witness can be found and verified. If the states are equivalent then we aim to show that however we start to build a supposed shortest witness (of nonequivalence), we will fail (demonstrate a failure) in finite time. This is shown by using a measure for "equivalence-level" of pairs of states which must be decreasing for the pairs along a real shortest witness. We thus can use the pairs which have bigger eq-levels, assuming we really build a shortest witness, for "balancing", i.e. making the structured states in the considered pairs close to each other, as objects with "bounded differing heads" but "the same (unbounded) tails". If this is done with some care, we get (a repeat of pairs or) a (sub)sequence of eq-level decreasing pairs with the same tails for which the heads grow in a controlled way. The last point is to demonstrate that if the length of such (sub)sequence is large enough then this contradicts the shortest witness assumption. In this paper, the idea of regular terms and of the basis helped

to establish this smoothly. For the complexity bound, we have not used the basis but we have looked at the structure of the above growing heads in more detail, and the analysis has been lead a bit further than Stirling's.

Hence essentially the proofs presented here are (further) simplifications of the previous proofs. It should be the readers from computer science community who decide if and how far this simplification is valuable.

Remark. Related to this is also an unclear status of papers by V. Yu. Meitus which were written in Russian but translated to English. He claims to have given the first proof for dpda language equivalence but this has been obviously not accepted by the computer science community.

The notion of a finite basis determined by the grammar (a basis is in fact, computable, but "after the fact", by using the established decidability) seems new in this context; it seems like a natural object for further exploration. In fact, it can be viewed as a generalization of the bisimulation base (or basis) used previously for context-free processes (generated by first-order grammars with arity-1 nonterminals); this line of research started with [1] and further developments can be found in [5].

The complexity is puzzling, the nonelementary upper bound induced by [20], which has been more specified here (in Theorem 29) seems unbelievably high and does not apply to bisimilarity; Sénizergues showed a more reasonable bound for a subclass in [18]. We can also note PSPACE-completeness of bisimilarity for ε-steps free one-counter automata [2] and NL-completeness of language equivalence for their deterministic version [3].

As already mentioned, there is no known nontrivial lower bound for dpda-equivalence while the general bisimilarity problem is known to be exptime-hard [11]. A slight generalization, namely allowing nondeterministic popping ε-steps when considering bisimilarity on (nondeterministic) pushdown automata, leads to undecidability [9].

References

[1] J. Baeten, J. Bergstra, and J. Klop, "Decidability of bisimulation equivalence for processes generating context-free languages," *J.ACM*, vol. 40, no. 3, pp. 653–682, 1993.

[2] S. Böhm, S. Göller, and P. Jančar, "Bisimilarity of one-counter processes is PSPACE-complete," in *CONCUR 2010 - Concurrency Theory*, ser. LNCS, vol. 6269. Springer-Verlag, 2010, pp. 177–191.

[3] S. Böhm and S. Göller, "Language equivalence of deterministic real-time one-counter automata is NL-complete," in *MFCS 2011*, ser. LNCS, vol. 6907. Springer-Verlag, 2011, pp. 194–205.

[4] C. H. Broadbent, A. Carayol, C.-H. L. Ong, and O. Serre, "Recursion schemes and logical reflection," in *LICS 2010*. IEEE Computer Society, 2010, pp. 120–129.

[5] O. Burkart, D. Caucal, F. Moller, and B. Steffen, "Verification on infinite structures," in *Handbook of Process Algebra*, J. Bergstra, A. Ponse, and S. Smolka, Eds. North-Holland, 2001, pp. 545–623.

[6] B. Courcelle, "Recursive applicative program schemes," in *Handbook of Theoretical Computer Science, vol. B*, J. van Leeuwen, Ed. Elsevier, MIT Press, 1990, pp. 459–492.

[7] W. Czerwiński and S. Lasota, "Fast equivalence-checking for normed context-free processes," in *Proc. FSTTCS'10*, ser. LIPIcs, vol. 8. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.

[8] S. Ginsburg and S. A. Greibach, "Deterministic context free languages," *Information and Control*, vol. 9, no. 6, pp. 620–648, 1966.

[9] P. Jančar and J. Srba, "Undecidability of bisimilarity by Defender's forcing," *J. ACM*, vol. 55, no. 1, 2008.

[10] S. Kiefer, A. S. Murawski, J. Ouaknine, B. Wachter, and J. Worrell, "On the complexity of the equivalence problem for probabilistic automata," *CoRR,*, vol. abs/1112.4644, 2011, to appear in Proc. FoSSaCS 2012.

[11] A. Kučera and R. Mayr, "On the complexity of checking semantic equivalences between pushdown processes and finite-state processes," *Inf. Comput.*, vol. 208, no. 7, pp. 772–796, 2010.

[12] I. Lanese, J. A. Pérez, D. Sangiorgi, and A. Schmitt, "On the expressiveness and decidability of higher-order process calculi," *Inf. Comput.*, vol. 209, no. 2, pp. 198–226, 2011.

[13] R. Milner, *Communication and concurrency*. Prentice-Hall, Inc., 1989.

[14] S. Salvati and I. Walukiewicz, "Krivine machines and higher-order schemes," in *ICALP(2) 2011*, ser. Lecture Notes in Computer Science, vol. 6756. Springer, 2011, pp. 162–173.

[15] G. Sénizergues, "L(A)=L(B)? Decidability results from complete formal systems," *Theoretical Computer Science*, vol. 251, no. 1–2, pp. 1–166, 2001, (A preliminary version appeared at ICALP'97.).

[16] ——, "L(A)=L(B)? a simplified decidability proof," *Theoretical Computer Science*, vol. 281, no. 1–2, pp. 555–608, 2002.

[17] ——, "The bisimulation problem for equational graphs of finite out-degree," *SIAM J.Comput.*, vol. 34, no. 5, pp. 1025–1106, 2005, (A preliminary version appeared at FOCS'98.).

[18] G. Sénizergues, "The equivalence problem for t-turn dpda is co-NP," in *ICALP 2003*, ser. LNCS, vol. 2719. Springer, 2003, pp. 478–489.

[19] C. Stirling, "Decidability of DPDA equivalence," *Theoretical Computer Science*, vol. 255, no. 1–2, pp. 1–31, 2001.

[20] ——, "Deciding DPDA equivalence is primitive recursive," in *Proc. ICALP'02*, ser. LNCS, vol. 2380. Springer-Verlag, 2002, pp. 821–832.

Appendix

Pushdown automata via first-order grammars

Here we give a more formal description of the reduction sketched in Fig.4. We first show how language equivalence for deterministic pushdown automata can be reduced to bisimulation equivalence on deterministic first-order grammars (which coincides with trace equivalence in this case).

A *deterministic pushdown automaton* (*dpda*) is a tuple $\mathcal{M} = (Q, \mathcal{A}, \Gamma, \Delta)$ consisting of finite sets Q of (control) states, $\mathcal{A}$ of actions (or terminals), Γ of stack symbols, and Δ of transition rules. For each pair pA, $p \in Q$, $A \in \Gamma$, and each $a \in \mathcal{A} \cup \{\varepsilon\}$, Δ contains at most one rule of the type $pA \stackrel{a}{\longrightarrow} q\alpha$, where $q \in Q$, $\alpha \in \Gamma^*$. Moreover, any pair pA is (exclusively) either *stable*, i.e. having no rule $pA \stackrel{\varepsilon}{\longrightarrow} q\alpha$, or *unstable*, in which case there is (one rule $pA \stackrel{\varepsilon}{\longrightarrow} q\alpha$ and) no rule $pA \stackrel{a}{\longrightarrow} q\alpha$ with $a \in \mathcal{A}$.

A dpda $\mathcal{M}$ generates a labelled transition system $(Q \times \Gamma^*, \mathcal{A} \cup \{\varepsilon\}, (\stackrel{a}{\longrightarrow})_{a \in \mathcal{A} \cup \{\varepsilon\}})$ where the states are configurations $q\alpha$ ($q \in Q$, $\alpha \in \Gamma^*$). We view a rule $pA \stackrel{a}{\longrightarrow} q\alpha$ as $pAx \stackrel{a}{\longrightarrow} q\alpha x$ (for a formal variable x), inducing $pA\beta \stackrel{a}{\longrightarrow} q\alpha\beta$ for every $\beta \in \Gamma^*$. The transition relation is extended to words $w \in \mathcal{A}^*$ as usual; we note that $p\alpha \stackrel{w}{\longrightarrow} q\beta$ can comprise more than $|w|$ basic steps, due to possible "silent" ε-moves. Each configuration $p\alpha$ has its associated *language* $L(p\alpha) = \{w \in \mathcal{A}^* \mid p\alpha \stackrel{w}{\longrightarrow} q\varepsilon \text{ for some } q\}$. The *dpda language equivalence problem* is: given a dpda $\mathcal{M}$ and two configurations $p\alpha$, $q\beta$, is $L(p\alpha) = L(q\beta)$?

Remark. It is straightforward to observe that this setting is equivalent to the classical problem of language equivalence between deterministic pushdown automata with accepting states. First, the disjoint union of two dpda's is a dpda. Second, for languages $L_1, L_2 \subseteq \Sigma^*$ we have $L_1 = L_2$ iff

$L_1 \cdot \{\$\} = L_2 \cdot \{\$\}$, for an endmarker $\$ \notin \Sigma$; so restricting to prefix-free deterministic context-free languages, accepted by dpda via empty stack, does not mean losing generality.

Each dpda $\mathcal{M}$ can be transformed by a standard polynomial-time algorithm so that all ε-transitions are popping, i.e., of the type $pA \stackrel{\varepsilon}{\longrightarrow} q$, while $L(pA\alpha)$, for stable pA, keep unchanged. (A principal point is that a rule $pA \stackrel{\varepsilon}{\longrightarrow} qB\alpha$ where $qB \stackrel{a_1}{\longrightarrow} q_1\beta_1$, ..., $qB \stackrel{a_k}{\longrightarrow} q_k\beta_k$ can be replaced with rules $pA \stackrel{a_j}{\longrightarrow} q_j\beta_j\alpha$; unstable pairs pA enabling only an infinite sequence of ε-steps are determined and removed.)

It is also harmless to assume that for each stable pA and each $a \in \mathcal{A}$ we have one rule $pA \stackrel{a}{\longrightarrow} q\alpha$ (since we can introduce a 'loop' state q_{L} with rules $q_{\text{L}}A \stackrel{a}{\longrightarrow} q_{\text{L}}A$ for all $A \in \Gamma, a \in \mathcal{A}$, and for every 'missing' rule $pA \stackrel{a}{\longrightarrow}$ .. we add $pA \stackrel{a}{\longrightarrow} q_{\text{L}}A$). $L(p\alpha)$ are unchanged by this transformation. Then $w \in \mathcal{A}^*$ is not enabled in $p\alpha$ iff $w = uv$ where $p\alpha \stackrel{u}{\longrightarrow} q\varepsilon$ (for some q), so $u \in L(p\alpha)$, and $v \neq \varepsilon$. This reduces language equivalence to trace equivalence:

$$L(p\alpha) = L(q\beta) \;\text{ iff }\; \forall w \in \mathcal{A}^* : p\alpha \stackrel{w}{\longrightarrow} \Leftrightarrow\; q\beta \stackrel{w}{\longrightarrow}.$$

Proposition 51. *The dpda language equivalence problem is polynomial-time reducible to the deterministic first-order grammar bisimulation equivalence problem.*

Proof: (Recall Fig.4.) Assume an (ε-popping) dpda $\mathcal{M} = (Q, \mathcal{A}, \Gamma, \Delta)$ transformed as above (so trace equivalence coincides with language equivalence). We define the first-order grammar $\mathcal{G}_{\mathcal{M}} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ where $\mathcal{N} = \{pA \mid pA \text{ is } \textit{stable}\} \cup \{\bot\}$; each $X = pA$ gets arity $m = |Q|$, and $\bot$ is a special nullary nonterminal not enabling any action. A dpda configuration $p\alpha$ is transformed to the term $\mathcal{T}(p\alpha)$ defined inductively by rules 1.,2.,3. below, assuming $Q = \{q_1, q_2, \ldots, q_m\}$.

1) $\mathcal{T}(q\varepsilon) = \bot$.
2) If $qA \stackrel{\varepsilon}{\longrightarrow} q_i$ (qA is unstable) then $\mathcal{T}(qA\beta) = \mathcal{T}(q_i\beta)$.
3) If qA is stable then $\mathcal{T}(qA\beta) = X\,\mathcal{T}(q_1\beta)\ldots\mathcal{T}(q_m\beta)$ where $X = qA$.
4) $\mathcal{T}(q_i x) = x_i$.

Rule 4. is introduced to enable the smooth transformation of a dpda rule $pA \stackrel{a}{\longrightarrow} q\alpha$, where $a \in \mathcal{A}$, rewritten in the form $pAx \stackrel{a}{\longrightarrow} q\alpha x$, to the $\mathcal{G}_{\mathcal{M}}$-rule $\mathcal{T}(pAx) \stackrel{a}{\longrightarrow} \mathcal{T}(q\alpha x)$, i.e. to $Yx_1 \ldots x_m \stackrel{a}{\longrightarrow} \mathcal{T}(q\alpha x)$, where $Y = pA$. Thus $\mathcal{R}$ in $\mathcal{G}_{\mathcal{M}}$ is defined (with no ε-moves). We observe easily: if $pA\alpha \stackrel{\varepsilon}{\longrightarrow} q\alpha$ (recall that ε-steps are popping) then $\mathcal{T}(pA\alpha) = \mathcal{T}(q\alpha)$; if $pA\alpha \stackrel{a}{\longrightarrow} q\beta\alpha$ ($a \in \mathcal{A}$, pA stable) then $\mathcal{T}(pA\alpha) \stackrel{a}{\longrightarrow} \mathcal{T}(q\beta\alpha)$. This also implies: if $p\alpha \stackrel{w}{\longrightarrow} q\varepsilon$ then $\mathcal{T}(p\alpha) \stackrel{w}{\longrightarrow} \bot$. Thus

$$L(p\alpha) = L(q\beta) \;\text{ iff }\; \big(\forall w \in \mathcal{A}^* : p\alpha \stackrel{w}{\longrightarrow} \Leftrightarrow\; q\beta \stackrel{w}{\longrightarrow}\big) \;\text{ iff }\; \mathcal{T}(p\alpha) \sim \mathcal{T}(q\beta).$$

We note that $\mathcal{T}(q\alpha)$ can have (at most) $1 + m + m^2 + m^3 + \cdots + m^{|\alpha|}$ subterm-occurrences, but the natural finite graph presentation of $\mathcal{T}(q\alpha)$ has at most $1 + m(|\alpha| - 1) + 1$ nodes and can be obviously constructed in polynomial time. ■

A *nondeterministic pushdown automaton where ε-steps are popping and deterministic* is defined as the dpda above, with stable and unstable configurations, but with the following relaxation for stable ones: there can be more rules of the type $pA \stackrel{a}{\longrightarrow}$ .. for $a \in \mathcal{A}$. The bisimilarity problem for such automata (solved in [17]) can be reduced to bisimilarity for (general) first-order grammars in the way analogous to the proof of Prop.51.