ELSEVIER

Theoretical Computer Science 255 (2001) 1–31

Theoretical Computer Science

www.elsevier.com/locate/tcs

Fundamental Study

Decidability of DPDA equivalence

Colin Stirling

Division of Informatics, University of Edinburgh, J.C. Maxwell Bldg., Mayfield Road, Edinburgh EH9 3JZ, UK

Received June 1999; revised July 2000; accepted August 2000
Communicated by A. Salomaa

Abstract

A proof of decidability of equivalence between deterministic pushdown automata is presented using a mixture of methods developed in concurrency and language theory. The technique appeals to a tableau proof system for equivalence of configurations of strict deterministic grammars.

Keywords: DPDA; Decidability; Language equivalence; Bisimulation; Fundamental study

Contents

1. The DPDA problem

Ingredients of pushdown automata with ε-transitions are a finite set of states P, a finite set of stack symbols S, a finite alphabet A and a finite family of basic transitions,

E-mail address: cps@dcs.ed.ac.uk (C. Stirling).

0304-3975/01/$ - see front matter
PII: S0304-3975(00)00389-3

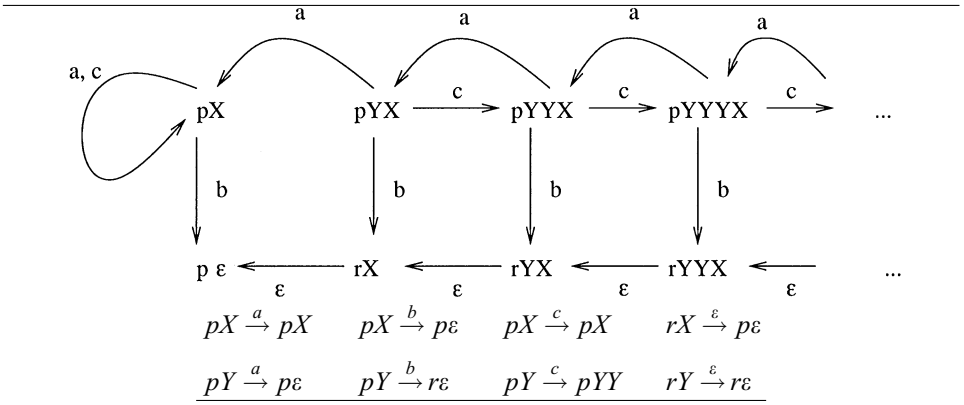

Fig. 1. A DPDA.

each of the form $pS \xrightarrow{a} q\alpha$ where p, q are states, $a \in \mathsf{A} \cup \{\varepsilon\}$, S is a stack symbol and α is a sequence of stack symbols. A configuration of an automaton is any expression $p\alpha$, $p \in \mathsf{P}$ and $\alpha \in \mathsf{S}^*$ whose behaviour is determined by the basic transitions together with the following prefix rule, where $\beta \in \mathsf{S}^*$:

if $pS \xrightarrow{a} q\alpha$ then $pS\beta \xrightarrow{a} q\alpha\beta$.

The language accepted by a configuration $p\alpha$ is $\{w \in \mathsf{A}^* : \exists q \in \mathsf{P}.\ p\alpha \xrightarrow{w} q\varepsilon\}$ where the extended transitions for words are defined as expected. Note that ε-transitions are swallowed in the usual fashion. Acceptance is by empty stack (and not by final state, see [6]).

A deterministic pushdown automaton, DPDA, has restrictions on its basic transitions

if $pS \xrightarrow{a} q\alpha$ and $pS \xrightarrow{a} r\beta$ then $q = r$ and $\alpha = \beta$,

if $pS \xrightarrow{\varepsilon} q\alpha$ and $pS \xrightarrow{a} r\lambda$ then $a = \varepsilon$.

Moreover, one can assume that in a basic transition $pS \xrightarrow{a} q\alpha$ the length of α is less than 3, and that ε-transitions can only pop the stack: if $pS \xrightarrow{\varepsilon} q\alpha$ then $\alpha = \varepsilon$. One consequence of the restrictions is that the language accepted by a configuration is prefix-free: if w is accepted then no proper prefix of w is accepted. Thus if ε is accepted then no other word is. In the following, we assume DPDAs which do not accept the language $\{\varepsilon\}$.[1] Fig. 1 depicts a simple DPDA whose basic transitions are listed under the diagram.

The DPDA decidability problem was first posed in 1966 [2]. Is there an effective procedure for deciding whether or not two configurations of a DPDA accept the same

[1] Classical DPDAs have final states, but for any language L recognised by a configuration of such a DPDA there is a configuration of a DPDA which does not accept ε with empty stack acceptance for the language $L\$$ where \$ is an endmarker.

language?[2] Why is the decision question so difficult to answer, despite all the intensive work on the problem over the past 30 years? Because one needs to expose the right structure. It appears that the notation of pushdown configurations, although simple, is not rich enough. Attempts to prove the result (such as Valiant's technique) examine differences between stack lengths and potentially equivalent configurations. This method showed decidability of equivalence for real-time DPDAs which have no ε-transitions [8]. But when there are ε-transitions it is possible for configurations of arbitrary size to be equivalent. For example, configurations pY^nX and pY^mX of Fig. 1 are equivalent for all m and n.

Finally, Sénizergues [9] showed that the problem is decidable. However, his proof is very intricate and when spelt out in full is over 70 pages long [10]. It exposes structure within a DPDA by representing configurations as boolean rational series, and he develops an interesting, albeit intricate, algebraic theory of their linear combinations. Equivalence between configurations is captured within a deduction system. The equations within the proof system have associated weights. Higher level strategies (transformations) are defined which guide proof. A novel feature is that these strategies depend upon differences between weights of their associated equations. Decidability is achieved by showing that two configurations are equivalent iff there is a finite proof of this fact. An especially formidable ingredient is the termination proof, that there is a *finite* proof of a true equation.

We provide a different proof of decidability, which is essentially a simplification of Sénizergues's proof. It utilises a mixture of techniques developed in concurrency theory and within language theory. From concurrency theory we employ methods developed for showing decidability of bisimulation equivalence for subsets of process calculi which are infinite state, and in particular tableaux proof systems [7, 12]. From language theory we utilise strict deterministic grammars, which were introduced by Harrison and Havel [4] because they are equivalent to DPDA. In effect we build a process calculus whose processes are derived from determinising strict grammars. These processes are essentially "associates", in the sense of [5]. However we endow them with algebraic structure. The DPDA equivalence problem is then equivalent to the bisimulation equivalence problem between these processes.

We prove that two processes are equivalent iff there is a finite tableau proof of this fact. However, the notion of tableau proof rule is richer than that used in concurrency theory. We utilise conditional proof rules which involve distances between premises. Essentially, this is a rationalisation of Sénizergues's use of weights, and the idea was developed from trying to understand his proof. We do not employ explicit weights within the proof system because there is already such a notion given by the bisimulation approximants. We believe that this makes the decidability proof much more manageable, especially the termination proof. However, the tableau proof rules are

[2] As the disjoint union of two DPDAs is a DPDA, we can assume the decision question over configurations of a single DPDA instead of between two different DPDAs.

essentially Sénizergues's strategies. The rule UNF is the strategy T_A in [10], BAL is T_B, and CUT is a variant of T_C, which uses a different "decomposition" mechanism (based on the notions of "unifier" and auxiliary nonterminals as developed previously in [1, 12]).

The paper is intended to be entirely self-contained, and all important proofs are presented. Examples are also interspersed throughout to aid understanding. In the next section strict deterministic grammars are introduced, and in Section 3 they are extended with auxiliary symbols. The tableau proof system is presented in Section 4, and its correctness is shown in Section 5.

2. Strict deterministic grammars

An ε-free context-free grammar in 3-Greibach normal form consists of a finite family N of nonterminals, a finite alphabet A and a finite family of basic transitions, each of the form $X \stackrel{a}{\rightarrow} \alpha$ where $X \in \mathsf{N}$, $a \in \mathsf{A}$ and $\alpha \in \mathsf{N}^*$ such that its length, $|\alpha|$, is less than 3. A simple configuration is a sequence of nonterminals whose behaviour is determined by the basic transitions and the prefix rule: if $X \stackrel{a}{\rightarrow} \alpha$ then $X\beta \stackrel{a}{\rightarrow} \alpha\beta$ where $\beta \in \mathsf{N}^*$. The language of a simple configuration α is the set of words $\{w \in \mathsf{A}^* : \alpha \stackrel{w}{\rightarrow} \varepsilon\}$. However, we shall also consider composite configurations which are finite families of simple configurations $\{\alpha_1, \ldots, \alpha_n\}$. We shall write a set as a sum form $\alpha_1 + \cdots + \alpha_n$. A degenerate case is the empty sum which we write as $\emptyset$. The language of a sum configuration is just the union of the languages of the components.

We are interested in a restricted family of context-free grammars, the *strict deterministic grammars* [4, 5]. Assume a context-free grammar (in 3-Greibach normal form). Let $\equiv$ be a partition of its nonterminals N. We extend $\equiv$ to sequences of nonterminals, $\alpha \equiv \beta$ iff $\alpha = \beta$ or there is a δ such that $\alpha = \delta X \alpha_1$ and $\beta = \delta Y \beta_1$ and $X \equiv Y$ and $X \neq Y$. Some simple properties of $\equiv$ are as follows.

Fact 1. (1) $\alpha\beta \equiv \alpha$ *iff* $\beta = \varepsilon$.
(2) $\alpha \equiv \beta$ *iff* $\delta\alpha \equiv \delta\beta$.
(3) *If* $\alpha \equiv \beta$ *and* $\gamma \equiv \delta$ *then* $\alpha\gamma \equiv \beta\delta$.
(4) *If* $\alpha \equiv \beta$ *and* $\alpha \neq \beta$ *then* $\alpha\gamma \equiv \beta\delta$.
(5) *If* $\alpha\gamma \equiv \beta\delta$ *and* $|\alpha| = |\beta|$ *then* $\alpha \equiv \beta$.

The partition $\equiv$ on N is *strict* if the basic transitions obey the following two conditions:

$$\text{if } X \stackrel{a}{\rightarrow} \alpha \text{ and } Y \stackrel{a}{\rightarrow} \delta \text{ and } X \equiv Y \text{ then } \alpha \equiv \delta,$$

$$\text{if } X \stackrel{a}{\rightarrow} \alpha \text{ and } Y \stackrel{a}{\rightarrow} \alpha \text{ and } X \equiv Y \text{ then } X = Y.$$

A context-free grammar is strict deterministic if there exists a strict partition of its nonterminals. We now examine some properties of strict deterministic grammars (which

are also shown in [5]). First, the strictness conditions generalise to words w, which is an instance of the following more general result.

Proposition 1. (1) *If* $\alpha \stackrel{w}{\rightarrow} \alpha'$ *and* $\beta \stackrel{w}{\rightarrow} \beta'$ *and* $\alpha \equiv \beta$ *then* $\alpha' \equiv \beta'$.
(2) *If* $\alpha \stackrel{w}{\rightarrow} \alpha'$ *and* $\beta \stackrel{w}{\rightarrow} \alpha'$ *and* $\alpha \equiv \beta$ *then* $\alpha = \beta$.

Proof. In both cases the proof is by induction on $|w|$. For the base case of 1 $|w| = 0$. In which case $\alpha \stackrel{w}{\rightarrow} \alpha$ and $\beta \stackrel{w}{\rightarrow} \beta$ and by assumption $\alpha \equiv \beta$. For the inductive step, assume $w = aw'$ and $\alpha \stackrel{a}{\rightarrow} \alpha_1 \stackrel{w'}{\rightarrow} \alpha'$ and $\beta \stackrel{a}{\rightarrow} \beta_1 \stackrel{w'}{\rightarrow} \beta'$. Therefore $\alpha = X\delta$ and $X \stackrel{a}{\rightarrow} \delta_1$ and $\alpha_1 = \delta_1\delta$ and $\beta = Y\gamma$ and $Y \stackrel{a}{\rightarrow} \gamma_1$ and $\beta_1 = \gamma_1\gamma$. Because $X\delta \equiv Y\gamma$ it follows from Fact 1.5 that $X \equiv Y$ and therefore $\delta_1 \equiv \gamma_1$ by the first condition of the definition of strictness. There are two cases to consider. First, $\delta_1 = \gamma_1$, and therefore by condition 2 of being strict $X = Y$ and therefore because $X\delta \equiv Y\gamma$ it follows from Fact 1.2 that $\delta \equiv \gamma$ and therefore $\delta_1\delta \equiv \gamma_1\gamma$ from Fact 1.3. Now the required result, $\alpha' \equiv \beta'$, follows by the induction hypothesis because $\alpha_1 \equiv \beta_1$ and $|w'| < |w|$. Second, $\delta_1 \neq \gamma_1$, and therefore because $\delta_1 \equiv \gamma_1$ it now follows from Fact 1.4 that $\delta_1\delta \equiv \gamma_1\gamma$. The required result now follows as in the first case.

The base case for 2 is $|w| = 0$. Therefore $\alpha' = \alpha$ and $\alpha' = \beta$ and therefore $\alpha = \beta$. For the inductive step assume $w = aw'$ and $\alpha \stackrel{a}{\rightarrow} \alpha_1 \stackrel{w'}{\rightarrow} \alpha'$ and $\beta \stackrel{a}{\rightarrow} \beta_1 \stackrel{w'}{\rightarrow} \alpha'$. As in the case of the proof of 1, $\alpha = X\delta$ and $X \stackrel{a}{\rightarrow} \delta_1$ and $\alpha_1 = \delta_1\delta$ and $\beta = Y\gamma$ and $Y \stackrel{a}{\rightarrow} \gamma_1$ and $\beta_1 = \gamma_1\gamma$. We can use the same argument as above to show that $\alpha_1 \equiv \beta_1$ and therefore by the induction hypothesis because $\alpha_1 \stackrel{w'}{\rightarrow} \alpha'$ and $\beta_1 \stackrel{w'}{\rightarrow} \alpha'$ it follows that $\alpha_1 = \beta_1$. Therefore $\delta_1\delta = \gamma_1\gamma$. However $\delta_1 \equiv \gamma_1$. By Fact 1.1 it is not possible for $\alpha \equiv \alpha X\lambda$. Therefore $\delta_1 = \gamma_1$ and $\delta = \gamma$. But also $X \stackrel{a}{\rightarrow} \delta_1$ and $Y \stackrel{a}{\rightarrow} \gamma_1$ and so by the second condition of being strict $X = Y$. □

The next result shows that if $\alpha \equiv \beta$ then their languages are prefix disjoint (part 2) and if also $\alpha \neq \beta$ then their languages are disjoint (part 3).

Proposition 2. (1) *If* $\alpha \stackrel{w}{\rightarrow} \varepsilon$ *and* $\alpha \equiv \beta$ *and* $\alpha \neq \beta$ *then not* $(\exists\gamma.\, \beta \stackrel{w}{\rightarrow} \gamma)$.
(2) *If* $\alpha \equiv \beta$ *and* $\alpha \stackrel{u}{\rightarrow} \varepsilon$ *and* $u = vaw$ *then not* $(\beta \stackrel{v}{\rightarrow} \varepsilon)$.
(3) *If* $\alpha \equiv \beta$ *and* $\alpha \neq \beta$ *then* $\{u : \alpha \stackrel{u}{\rightarrow} \varepsilon\} \cap \{v : \beta \stackrel{v}{\rightarrow} \varepsilon\} = \emptyset$.

Proof. To show 1 assume $\alpha \stackrel{w}{\rightarrow} \varepsilon$ and $\alpha \equiv \beta$ and $\alpha \neq \beta$ and $\beta \stackrel{w}{\rightarrow} \gamma$. By Proposition 1.1 $\varepsilon \equiv \gamma$ and therefore by Fact 1.1 $\gamma = \varepsilon$, but then by Proposition 1.2 $\alpha = \beta$, contrary to assumption (2) and (3) are now immediate corollaries. □

Our main concern is with a subset of composite configurations. A composite configuration $\beta_1 + \cdots + \beta_n$ is *admissible* if $\beta_i \equiv \beta_j$ for each pair of components β_i and β_j. The empty sum, $\emptyset$, is therefore admissible. In [5] admissible configurations are called "associates". The following is a simple corollary of Proposition 1, that "reachability"' under any word preserves admissibility.

Fact 2. *If* $\{\beta_1,\ldots,\beta_n\}$ *is admissible then for any* w, $\{\beta' : \beta_i \xrightarrow{w} \beta',\ 1 \leqslant i \leqslant n\}$ *is admissible.*

There is a standard transformation of a pushdown automaton into a language equivalent context-free grammar (whose nonterminals are triples $[pSq]$ where p and q are states and S is a stack symbol and whose language is the set of words w such that $pS \xrightarrow{w} q\varepsilon$), see for instance [3, 6]. Harrison and Havel introduced strict deterministic grammars because they are the transformations of ε-free DPDA [4]. They also show the converse, that any strict deterministic grammar can be transformed back into an ε-free DPDA. We now describe the transformation of an ε-free DPDA into an equivalent 3-Greibach normal form strict deterministic grammar.

Assume an ε-free DPDA. For every pair of states p, q and stack symbol S introduce a nonterminal $[pSq]$, whose language is $\{w \in \mathsf{A}^* :\ pS \xrightarrow{w} q\varepsilon\}$. To ensure this the basic transitions for $a \in \mathsf{A}$ are translated: $pS \xrightarrow{a} q\varepsilon$ becomes $[pSq] \xrightarrow{a} \varepsilon$, $pS \xrightarrow{a} qT$ becomes the family for each r, $[pSr] \xrightarrow{a} [qTr]$, and $pS \xrightarrow{a} qTU$ becomes the family for each r and p', $[pSr] \xrightarrow{a} [qTp'][p'Ur]$. Erase all ε-nonterminals (if $pS \xrightarrow{\varepsilon} q\varepsilon$ then $[pSq]$ is an ε-nonterminal) from the right-hand side of any transition. Next, delete all transitions involving redundant nonterminals (those which accept no words). It is easy to check that the partition $\equiv$ relating pairs $[pSq]$ and $[pSr]$ is strict.[3] A configuration $pS_1S_2\ldots S_n$ of the DPDA is transformed into the following admissible configuration, where the summation is over all p_i, $1 \leqslant i \leqslant n$:

$$\sum [pS_1p_1][p_1S_2p_2]\ldots[p_{n-1}S_np_n]$$

after all ε-nonterminals are erased and all components involving redundant nonterminals are removed. The proof that the transformation preserves language equivalence is straightforward, and instead of reproducing it we illustrate the transformation on the DPDA of Fig. 1.

Initially, $\{[pXp],[pXr],[rXp],[rXr],[pYp],[pYr],[rYp],[rYr]\}$ is the set of nonterminals. The basic transitions are translated as follows:

$$[pXp] \xrightarrow{a} [pXp], \qquad [pXr] \xrightarrow{a} [pXr], \qquad [pXp] \xrightarrow{b} \varepsilon,$$

$$[pXp] \xrightarrow{c} [pXp], \qquad [pXr] \xrightarrow{c} [pXr],$$

$$[pYp] \xrightarrow{a} \varepsilon, \qquad [pYr] \xrightarrow{b} \varepsilon, \qquad [pYp] \xrightarrow{c} [pYp][pYp],$$

$$[pYp] \xrightarrow{c} [pYr][rYp], \quad [pYr] \xrightarrow{c} [pYp][pYr], \quad [pYr] \xrightarrow{c} [pYr][rYr].$$

There are two ε-nonterminals, $[rXp]$ and $[rYr]$, which are erased from the right-hand side of any transition: the transition $[pYr] \xrightarrow{c} [pYr][rYr]$ is changed to $[pYr] \xrightarrow{c} [pYr]$. There are also three redundant nonterminals $[pXr]$, $[rXr]$ and $[rYp]$. All transitions

[3] By determinism, for instance, if $q \neq r$ and $pS \xrightarrow{a} q\varepsilon$ then $[pSq] \xrightarrow{a} \varepsilon$ but not ($[pSr] \xrightarrow{a} \beta$), and if $pS \xrightarrow{a} p'T$ then $[pSq] \xrightarrow{a} [p'Tq]$ and $[pSr] \xrightarrow{a} [p'Tr]$.

involving these nonterminals are removed. This reduces the transitions to the following set:

$$[pXp] \xrightarrow{a} [pXp], \quad [pXp] \xrightarrow{b} \varepsilon, \quad [pXp] \xrightarrow{c} [pXp],$$
$$[pYp] \xrightarrow{a} \varepsilon, \quad [pYr] \xrightarrow{b} \varepsilon, \quad [pYp] \xrightarrow{c} [pYp][pYp],$$
$$[pYr] \xrightarrow{c} [pYp][pYr], \quad [pYr] \xrightarrow{c} [pYr].$$

The final set of nonterminals is $\{[pXp],[pYp],[pYr]\}$ and the partition is into the sets $\{\{[pXp]\},\{[pYp],[pYr]\}\}$. The configuration $pYYX$ of the DPDA becomes the following admissible configuration $[pYp][pYp][pXp]+[pYp][pYr]+[pYr]$.

The DPDA language equivalence problem reduces to[4] the problem of language equivalence between admissible configurations of a strict deterministic grammar. Transitions of a DPDA are deterministic whereas there is constrained nondeterminism in the case of a strict grammar (for example, $[pYr]$ above has two distinct c-transitions). Therefore, we now determinise a strict grammar by defining deterministic transition relations between admissible configurations. The idea is as in process calculi that one builds transitions from a composite process out of transitions of its components. First, the basic transitions are determinised by coalescing all the basic transitions of a nonterminal with the same label. If $X \xrightarrow{a} \alpha_1$ and $\ldots$ and $X \xrightarrow{a} \alpha_n$ then form the single transition $X \xrightarrow{a} \alpha_1+\cdots+\alpha_n$. By Proposition 1 $\alpha_1+\cdots+\alpha_n$ is admissible. We also assume that if X has no a-transitions then $X \xrightarrow{a} \emptyset$. Consequently for each nonterminal X and each $a \in \mathsf{A}$ there is a single transition rule $X \xrightarrow{a} \sum \alpha_j$. For instance the rule for $[pYr]$ and c above becomes $[pYr] \xrightarrow{c} [pYp][pYr]+[pYr]$. The transition rule for admissible configurations, the prefix rule, is then as follows:

$$\text{if } X_i \xrightarrow{a} \sum \alpha_{ij} \text{ then } \sum X_i\beta_i \xrightarrow{a} \sum\sum \alpha_{ij}\beta_i.$$

By Proposition 1 the resulting configuration is admissible.

Example 1. The determinised strict grammar of the example above, assuming A is $[pXp]$, B is $[pYp]$ and C is $[pYr]$ and $B \equiv C$, has the following transitions:

$$A \xrightarrow{a} A, \quad A \xrightarrow{b} \varepsilon, \quad A \xrightarrow{c} A,$$
$$B \xrightarrow{a} \varepsilon, \quad B \xrightarrow{b} \emptyset, \quad B \xrightarrow{c} BB,$$
$$C \xrightarrow{c} \emptyset, \quad C \xrightarrow{b} \varepsilon, \quad C \xrightarrow{c} BC + C.$$

The transition $BBA + BC + C \xrightarrow{c} BBBA + BBC + BC + C$ corresponds exactly to $pYYX \xrightarrow{c} pYYYX$ of Fig. 1.

[4] Is in fact equivalent to.

Example 2. Assume nonterminals A, B, A', B' where $A \equiv B$ and $A' \equiv B'$. The transitions are as follows, where we omit the $\emptyset$ cases:

$$A \xrightarrow{a} \varepsilon, \quad A' \xrightarrow{a} \varepsilon, \quad B \xrightarrow{b} \varepsilon, \quad B' \xrightarrow{b} \varepsilon,$$
$$A \xrightarrow{c} AA, \quad A' \xrightarrow{c} A'A', \quad B \xrightarrow{c} BB, \quad B' \xrightarrow{c} B'B'.$$

The grammar is strict deterministic, and for instance, $AAA + BB \xrightarrow{c} AAAA + BBB$.

The extended transition relation $\xrightarrow{w}$, $w \in \mathsf{A}^*$, between admissible configurations of a determinised strict grammar is defined as expected. Consequently, the language accepted by an admissible configuration $\beta_1 + \cdots + \beta_k$ is the set of words $\{w : \beta_1 + \cdots + \beta_k \xrightarrow{w} \varepsilon\}$. Two admissible configurations are equivalent if they accept the same language. Language equivalence coincides here with bisimulation equivalence.

3. Recursive nonterminals and shapes

Assume a fixed determinised strict grammar in 3-Greibach normal form without redundant nonterminals. We use $\alpha, \beta, \ldots$ to range over sequences of nonterminals and E, F, G, ... to range over admissible configurations. The size of an admissible configuration $E = \beta_1 + \cdots + \beta_n$, written $|E|$, is the length of its longest sequence of nonterminals, $\max\{|\beta_j|: \ 1 \leqslant j \leqslant n\}$ [5]. For each n there are only finitely many admissible configurations of size n.

We assume a fixed total ordering on the alphabet A. From this we define a total ordering on words, $u < v$ if $|u| < |v|$ or $|u| = |v|$ and u is lexicographically less than v. If $u < v$ we say that u is shorter than v. For each nonterminal X there is a unique shortest word u such that $X \xrightarrow{u} \varepsilon$. We let $w(X)$ denote this word and we let the *norm* of X be its length. An important measure is M which is the maximum norm of the grammar:

$$\mathrm{M} = \max\{|w(X)|: \ X \text{ is a nonterminal}\}.$$

The notion of norm extends to admissible configurations. The norm of E is the length of the smallest word u such that $E \xrightarrow{u} \varepsilon$ [6]. Infinitely many different admissible configurations can have the same norm (and this is one reason why the decision problem is difficult).

Although the starting point is a fixed strict deterministic grammar we shall extend it with auxiliary nonterminals, ranged over by V, each of which has an associated definition $V \stackrel{\text{def}}{=} H$. We say that $(V_1, \ldots, V_n)$ is a family of *recursive nonterminals* if for each $i : 1 \leqslant i \leqslant n$

1. either $V_i \stackrel{\text{def}}{=} \beta_{i1} V_{i1} + \cdots + \beta_{im} V_{im}$ where each $\beta_{ij} \neq \varepsilon$ and

[5] We let $|\emptyset| = 0$.

[6] We assume that the norm of $\emptyset$ is ∞.

(a) no β_{ij} contains auxiliary nonterminals, and if $j \neq k$ then $\beta_{ij} \neq \beta_{ik}$,
(b) $\beta_{i1} + \cdots + \beta_{im}$ is admissible and each $V_{ij} \in \{V_1, \ldots, V_n\}$.

2. or $V_i \stackrel{\text{def}}{=} V_j$ and $j \leqslant i$ and $V_j \stackrel{\text{def}}{=} V_j$.

Auxiliary nonterminals play an important role in the decidability proof. However, their occurrence in an admissible configuration is severely restricted. They can only appear as a final element in a sequence of nonterminals. Admissibility is extended to such families of sequences as follows. A configuration which is a singleton V is admissible, and $\beta_1 V_1' + \cdots + \beta_k V_k'$ is admissible if the head $\beta_1 + \cdots + \beta_k$ is admissible and each β_j is distinct and does not contain auxiliary nonterminals, and there is a family of recursive nonterminals $(V_1, \ldots, V_n)$ such that each V_i' is one of the V_j's. We assume that $|V| = 1$ for each recursive nonterminal V. The transition relation is extended to the wider class of admissible configurations with the following rule for any $w \in \mathsf{A}^*$:

$$\text{if } E \xrightarrow{w} V_i \text{ and } V_i \stackrel{\text{def}}{=} H \text{ then also } E \xrightarrow{w} H.$$

We use the notation $E{\cdot}u$ for "the result of E after the word u" which is the configuration F such that $E \xrightarrow{u} F$, which can be $\emptyset$. If E does not contain recursive nonterminals then $E \cdot u$ is unique. If E does contain recursive nonterminals then we ensure it is unique by stipulating that if $E \xrightarrow{u} V_i$ and $V_i \stackrel{\text{def}}{=} H$ then $E \cdot u = H$. Consequently, when $E \cdot u = V_i$, it follows that $V_i \stackrel{\text{def}}{=} V_i$. We shall appeal to some obvious properties of $E \cdot u$.

Fact 3. (1) $(E \cdot uv) = (E \cdot u) \cdot v$.
(2) *If* $(E \cdot u) = \emptyset$ *then* $(E \cdot uv) = \emptyset$.
(3) *If* $(E \cdot u) = \varepsilon$ *or* V *then* $(E \cdot ua) = \emptyset$.

If E does not contain recursive nonterminals then its language is the set of words $\{w : E \cdot w = \varepsilon\}$. If E contains recursive nonterminals then its language is the set $\{w : (E \cdot w) = V_i \text{ for some } V_i\}$. A recursive nonterminal V_i such that $V_i \stackrel{\text{def}}{=} V_i$ is a terminating nonterminal. The norm of E is still the length of the smallest word accepted by E.

Two configurations E and F are equivalent, written $E \sim F$, if they accept the same language and, when applicable, agree on terminating recursive nonterminals: that is, for any u and V_i, $E \cdot u = V_i$ iff $F \cdot u = V_i$. A configuration E "rejects" word u iff $E \cdot u = \emptyset$. An admissible configuration E is either a set $\{\beta_1, \ldots, \beta_n\}$ or $\{\beta_1 V_1, \ldots, \beta_n V_n\}$ where in both cases each $\beta_i \equiv \beta_j$ and $\beta_i \neq \beta_j$ for each i and $j \neq i$. Moreover we assume that if $n > 0$ then such a configuration has a finite norm.

Proposition 3. $E \sim F$ *iff for all words* w
1. $(E \cdot w) = \emptyset$ *iff* $(F \cdot w) = \emptyset$, *and*
2. $(E \cdot w) = V_i$ *iff* $(F \cdot w) = V_i$.

Proof. Assume $E \sim F$, but $E \cdot w = \emptyset$ and $F \cdot w = F' \neq \emptyset$. By the normed constraint, there is a word v such that $F' \cdot v = \varepsilon$ or V_i. Hence $F \cdot wv = \varepsilon$ or V_i but $(E \cdot wv) = \emptyset$

which contradicts that $E \sim F$. Assume 1 and 2 hold but $E \not\sim F$. Let w be the smallest distinguishing word for E and F. Assume $E \cdot w = \varepsilon$ and $F \cdot w = F' \neq \varepsilon$. If $F' = \emptyset$ then 1 fails. If $F' \neq \emptyset$ then either $F' = V_i$ which contradicts 2 or there is an a such that $F' \cdot a \neq \emptyset$. However $E \cdot wa = \emptyset$. □

Later we shall use Proposition 3 as the criterion for equivalence of admissible configurations. Equivalence can also be "approximated". For $n \geqslant 0$ we say that E and F are n-equivalent, written $E \sim_n F$, iff for all words w whose length $|w| \leqslant n$

$$(E \cdot w) = \emptyset \text{ iff } (F \cdot w) = \emptyset \text{ and } (E \cdot w) = V_i \text{ iff } (F \cdot w) = V_i.$$

Note that for each n it is decidable whether $E \sim_n F$. The following clearly holds.

Fact 4. *$E \sim F$ iff $E \sim_n F$ for all $n \geqslant 0$.*

If E and F are admissible and $E \cup F$ is admissible then we let $E + F$ represent this configuration. Note that E or F could be $\emptyset$. We also introduce sequential composition, $EF = \{\beta\gamma : \beta \in E \text{ and } \gamma \in F\}$. The following result provides admissibility well-formedness conditions for composition. [7]

Proposition 4. *Assume $E_1 + \cdots + E_n$ is admissible and $E_i \cap E_j = \emptyset$ for each i and $j \neq i$ and no E_i contains recursive nonterminals or is ε:*
1. *If for each $i : 1 \leqslant i \leqslant n$, G_i is admissible then $E_1G_1 + \cdots + E_nG_n$ is admissible.*
2. *If $E_1G_1 + \cdots + E_nG_n$ is admissible and $E_i \neq \emptyset$ then G_i is admissible.*

Proof. Assume each G_i, $1 \leqslant i \leqslant n$, is admissible. Consider any pair δ and δ' in $E_1G_1 + \cdots + E_nG_n$. We show that $\delta \equiv \delta'$. Suppose $\delta \in E_iG_i$ and $\delta' \in E_jG_j$. Therefore $\delta = \beta\gamma$ and $\beta \in E_i$ and $\gamma \in G_i$, and $\delta' = \beta'\gamma'$ and $\beta' \in E_j$ and $\gamma' \in G_j$. By assumption β and β' are not ε. If $i \neq j$ then $\beta \equiv \beta'$ and $\beta \neq \beta'$, and so $\beta\gamma \equiv \beta'\gamma'$ by Fact 1.4. If $i = j$ and $\beta \neq \beta'$ then the same argument shows $\beta\gamma \equiv \beta'\gamma'$. Otherwise $\beta = \beta'$. However $\gamma \equiv \gamma'$ and so by Fact 1.2 $\beta\gamma \equiv \beta'\gamma'$. To prove 2, assume $E_1G_1 + \cdots + E_nG_n$ is admissible. Assume γ and γ' are in G_i. We show $\gamma \equiv \gamma'$. Consider any $\beta \in E_i$ (and by assumption $\beta \neq \varepsilon$). By assumption $\beta\gamma \equiv \beta\gamma'$, and therefore by Fact 1.2 $\gamma \equiv \gamma'$. □

This result is used repeatedly in the decidability proof. It allows one to view an admissible configuration as having a variety of different "shapes", when common subterms are collected together using $+$, sequential composition and $\emptyset$. Throughout the rest of the paper when we write an admissible configuration as $E_1G_1 + \cdots + E_nG_n$ we assume the following:

1. $E_1 + \cdots + E_n$ is admissible and $E_i \cap E_j = \emptyset$ for each i and $j \neq i$.
2. No E_i contains recursive nonterminals or is ε.
3. No G_i is $\emptyset$.

[7] We assume in this result that either none of the "tails" G_i contain recursive nonterminals, or they all contain recursive nonterminals drawn from some family.

We now describe two simple consequences of Proposition 4. The first is that we can redefine recursive nonterminals as follows. $(V_1,\ldots,V_n)$ is a family of recursive nonterminals if for each $i: 1 \leqslant i \leqslant n$

1. either $V_i \stackrel{\text{def}}{=} V_j$ and $j \leqslant i$,
2. or $V_i \stackrel{\text{def}}{=} H_1V_1 + \cdots + H_nV_n$.

This shape is guaranteed by letting H_j consist of all the β such that βV_j is a component of the definition of V_i. The second consequence is that substitutivity of subterms preserves admissibility. If $E_1G_1 + \cdots + E_nG_n$ is admissible and each H_i for $i: 1 \leqslant i \leqslant n$ is admissible[8] then $E_1H_1 + \cdots + E_nH_n$ is admissible.

An admissible configuration E therefore has many different "shapes". However the size of E does not depend on the presentation. We now go one step further. If E contains recursive nonterminals then we allow presentations of E in which recursive nonterminals are substituted by their definitions, provided that the size is not increased. If βV_i is a component of E and $V_i \stackrel{\text{def}}{=} V_j$ then we allow E to be presented with component βV_j instead of βV_i. If βV_i is a component of E and $V_i \stackrel{\text{def}}{=} H_1V_1 + \cdots + H_nV_n$ and $|E| \geqslant |\beta H_1V_1 + \cdots + \beta H_nV_n|$ then we allow E to be presented with component βV replaced by $\beta H_1V_1 + \cdots + \beta H_nV_n$. By Proposition 4 this representation preserves admissibility. It also preserves n-equivalence (and therefore equivalence).

Fact 5. (1) *If* $V_i \stackrel{\text{def}}{=} V_j$ *then* $\beta V_i \sim_n \beta V_j$ *for all* $n \geqslant 0$.
(2) *If* $V_i \stackrel{\text{def}}{=} H_1V_1 + \cdots + H_nV_n$ *then* $\beta V_i \sim_n \beta H_1V_1 + \cdots + \beta H_nV_n$.

This additional flexibility does not compromise the fact that it is simple to decide whether two presentations E and E' are of the same admissible configuration.

A special presentation of an admissible configuration is in "head normal form". We say that E is in n-head form for $n \geqslant 1$ if E is $\beta_1G_1 + \cdots + \beta_kG_k$ where each β_i is distinct, different from ε and $\beta_1 + \cdots + \beta_k$ is admissible, and when E does not contain recursive nonterminals then either $|\beta_i| = n$ or $|\beta_i| < n$ and $G_i = \varepsilon$, and when E does contain recursive nonterminals then either $|\beta_i| = n$ and $|G_i| > 0$ or $|\beta_i| < n$ and G_i is a recursive nonterminal V. In this case we say that the β_i's are the "heads" and the G_i's are the "tails". Any admissible configuration E such that $|E| \geqslant 2$ has an n-head form.

Proposition 5. *Assume* $E = \beta_1G_1 + \cdots + \beta_kG_k$ *is in n-head form*:

1. *If* $\beta_i \cdot w = \varepsilon$ *then for all* $j \neq i$, $(\beta_j \cdot w) = \emptyset$ *and* $E \xrightarrow{w} G_i$.
2. *If* $(\beta_i \cdot w)$ *is different from* ε *and* $\emptyset$ *then* $(E \cdot w) = (\beta_1 \cdot w)G_1 + \cdots + (\beta_k \cdot w)G_k$.
3. *If* $|\beta_i| = m$ *and* $|w| < m$ *and* $(\beta_i \cdot w) \neq \emptyset$ *then* $m - |w| \leqslant |\beta_i \cdot w| \leqslant m + |w|$.

[8] And that either none of the H_i contain recursive nonterminals, or they all contain recursive nonterminals drawn from some family.

Proof. Conditions 1 and 2 are simple consequences of Proposition 2, and 3 follows because the grammar is in 3-Greibach normal form. □

Equivalence between configurations which contain recursive nonterminals is more intensional than simply accepting the same language. We have also included the condition that they must agree on terminating nonterminals. One reason is that n-equivalence (and hence equivalence) preserves refinement. $(V_1', \ldots, V_n')$ is said to refine the family $(V_1, \ldots, V_n)$ iff the following two conditions hold:

$$\text{if } V_i \stackrel{\text{def}}{=} H_1 V_1 + \cdots + H_n V_n \text{ then } V_i' \stackrel{\text{def}}{=} H_1 V_1' + \cdots + H_n V_n',$$

$$\text{if } V_i \stackrel{\text{def}}{=} V_j \text{ and } V_i' \stackrel{\text{def}}{=} H \text{ then } V_j' \stackrel{\text{def}}{=} H.$$

A refined family agrees on the definitions of nonterminating nonterminals and preserves equality of definitions, but may contain fewer terminating nonterminals.

Proposition 6. *Assume* $E = E_1 V_1 + \cdots + E_n V_n$, $F = F_1 V_1 + \cdots + F_n V_n$, $(V_1', \ldots, V_n')$ *refines* $(V_1, \ldots, V_n)$, $E' = E_1 V_1' + \cdots + E_n V_n'$ *and* $F' = F_1 V_1' + \cdots + F_n V_n'$. *If* $E \sim_n F$ *then* $E' \sim_n F'$.

Proof. Assume $E \sim_n F$. Suppose $|w| \leqslant n$ and $E' \cdot w = \emptyset$ but $F' \cdot w \neq \emptyset$. Consider the longest prefix w' of w such that $F \cdot w' \neq \emptyset$. If $w' = w$ then $E \cdot w \neq \emptyset$ and therefore $E' \cdot w \neq \emptyset$ which is a contradiction. So w' is a proper prefix of w, and so $F \cdot w' = V_i$ and $V_i \stackrel{\text{def}}{=} V_i$. Hence $F' \cdot w' = H$ where $V_i' \stackrel{\text{def}}{=} H$. But then also $E' \cdot w' = H$ and so $E' \cdot w \neq \emptyset$. The argument is similar if $E' \cdot w = V_i'$ and $F' \cdot w \neq V_i'$. □

Finally, we collect together a variety of routine results about equivalence, approximation and congruence which will be used later.

Fact 6. (1) $E \sim F$ *iff for all* $u \in {}^*$, $E \cdot u \sim F \cdot u$.
(2) *If* $m \leqslant n$ *then* $E \sim_n F$ *iff for all* $u \in {}^*$, $|u| = m$, $E \cdot u \sim_{n-m} F \cdot u$.
(3) *If* $E \sim_n F$ *and* $0 \leqslant m < n$ *then* $E \sim_m F$.
(4) *If* $E \sim E'$ *and* $F \sim F'$ *then* $E + F \sim E' + F'$.
(5) *If* $E \sim_n E'$ *and* $F \sim_n F'$ *then* $E + F \sim_n E' + F'$.
(6) *If* $E \sim_n F$ *and* $F \not\sim_n G$ *then* $E \not\sim_n G$.
(7) *If* $EF \sim G$ *and* $F \sim F'$ *then* $EF' \sim G$.
(8) *If* $EF \sim G$ *and* $E \sim E'$ *then* $E'F \sim G$
(9) *If* $EF \sim_n G$ *and* $|E| > 0$ *and* $F \sim_{n-1} F'$ *then* $EF' \sim_n G$.

4. The tableau proof system

Consider trying to show that $E \sim F$. One approach is goal directed. Start with the goal $E = F$ (to be understood as "is $E \sim F$?") and then reduce it to subgoals. Keep reducing to further subgoals until one reaches either obviously true subgoals (such as

$$\textit{UNF}\quad \frac{E = F}{E \cdot a_1 = F \cdot a_1 \cdots E \cdot a_k = F \cdot a_k}\,\mathsf{A} = \{a_1, \ldots, a_k\}$$

Fig. 2. The rule UNF.

$G = G$) or obviously false subgoals (such as $G = H$ when $G = \emptyset$ and $H \neq \emptyset$). This naive technique is now described more formally in terms of tableaux.

A tableau proof system consists of rules which allow one to reduce goals to subgoals. We appeal to two kinds of tableau proof rule, "simple" and "conditional". A simple rule has the following form:

$$\frac{\text{Goal}}{\text{Subgoal}_1 \ldots \text{Subgoal}_n}\,\text{C}$$

The antecedent Goal is reduced to the consequent Subgoals, provided that the condition C holds. Next is a conditional rule

$$\frac{\begin{array}{c}\text{Goal}_1\\ \vdots\\ \text{Goal}_k\\ \vdots\\ \text{Goal}\end{array}}{\text{Subgoal}}\,\text{C}$$

where Goal is the current goal which reduces to Subgoal provided that the goals $\text{Goal}_1, \ldots, \text{Goal}_k$ occur above Goal on the path between it and the root (starting goal) and provided that the side condition C holds.

One builds a proof tree starting from an initial goal and repeatedly applying the rules. There is also the important notion of when a goal is a final goal. Final goals are classified as either "successful" or "unsuccessful". A successful *tableau* proof for Goal is a finite proof tree whose root is Goal and all of whose leaves are successful final goals, and all of whose inner subgoals are the result of an application of one of the rules.

In the case of the tableau proof system that we now present goals and subgoals are all of the form $E = F$ where E and F are admissible configurations which may contain recursive nonterminals. It is our intention to show that $E \sim F$ iff there is a successful tableau proof for $E = F$. There are just four tableau proof rules and they are presented each in turn.

4.1. UNF

There is one simple rule UNF, for unfold, presented in Fig. 2. A goal $E = F$ reduces to the subgoals $E \cdot a = F \cdot a$ for each $a \in$. UNF obeys local completeness and soundness. Completeness is that if the goal is true, $E \sim F$, then so are all the subgoals, $E \cdot a \sim F \cdot a$,

which follows from Fact 6.1. Soundness is that if all the subgoals are true then so is the goal, or equivalently if the goal is false then so is at least one of the subgoals. A finer version uses approximants, which provide a measure of how false a goal $E = F$ is. Consider the smallest n such that $E \not\sim_n F$. In the case of UNF if the goal is false at $n+1$, $E \not\sim_{n+1} F$, then at least one of the subgoals is false at n, $E \cdot a \not\sim_n F \cdot a$, which follows from Fact 6.2.

Example 3. Below is an illustration of an application of UNF where A, B and C are from Example 1:

$$\frac{BA + C = BBA + BC + C}{A = BA + C\varepsilon = \varepsilon BBA + BC + C = BBBA + BBC + BC + C}$$

The three subgoals are a result of the goal after a, b and c. □

If $E' = F'$ is a subgoal which is the result of m consecutive applications of UNF (and no other rule) from the goal $E = F$ then there is a word u such that $|u| = m$ and $E' = (E \cdot u)$ and $F' = (F \cdot u)$. In this circumstance, we say that u is the "associated" word with this sequence of applications of UNF.

4.2. *BAL(L) and BAL(R)*

The next two rules allow goals to be reduced to "balanced" subgoals. If E has shape $E_1G_1 + \cdots + E_nG_n$ and F has a similar shape $F_1G_1 + \cdots + F_nG_n$ then the imbalance between E and F with these shapes is $\max\{|E_i|, |F_i| : 1 \leqslant i \leqslant n\}$. If the imbalance is 0 then the configurations are identical. The balance rules are conditional and are presented in Fig. 3.

We explain how BAL(L) can reduce imbalance. For ease of exposition we only consider the case when the goals do not contain recursive nonterminals.[9] $E_1H_1 + \cdots + E_kH_k = F'$ is the result of m consecutive applications of UNF from $X_1H_1 + \cdots + X_kH_k = F$, where the left hand configuration is in 1-head normal form. Assume that u is the word associated with the sequence of applications of UNF between the top and bottom goal. Clearly, $m \leqslant \mathrm{M}$ and because $E_i = X_i \cdot u$ for each i it follows that $|E_i| \leqslant \mathrm{M} + 1$ (and $E_1 + \cdots + E_k$ is admissible by Fact 2). Consider F in $(\mathrm{M}+1)$-head form, $\beta_1G_1 + \cdots + \beta_nG_n$. Therefore, because $|u| \leqslant \mathrm{M}$ and $|w(X_i)| \leqslant \mathrm{M}$ the following are true via Proposition 5 (and that if $|\beta_i| < \mathrm{M}+1$ then $G_i = \varepsilon$):

$$(F \cdot u) = (\beta_1 \cdot u)G_1 + \cdots + (\beta_n \cdot u)G_n,$$

$$(F \cdot w(X_i)) = (\beta_1 \cdot w(X_i))G_1 + \cdots + (\beta_n \cdot w(X_i))G_n.$$

Let F'_i be $(\beta_i \cdot u)$ for $i : 1 \leqslant i \leqslant n$, and therefore $|F'_i| \leqslant 2\mathrm{M} + 1$. Because $F' = F \cdot u$ the second goal has the form $E_1H_1 + \cdots + E_kH_n = F'_1G_1 + \cdots + F'_nG_n$. BAL(L) sanctions

[9] If the goals do contain recursive nonterminals then the argument is similar when the goals are large, as discussed in Section 6.

$$\dfrac{\dfrac{}{BAL(R)\quad F = X_1H_1 + \cdots + X_kH_k \quad \vdots \quad F' = E_1H_1 + \cdots + E_kH_k}\ \mathrm{C}}{F' = E_1(F \cdot w(X_1)) + \cdots + E_k(F \cdot w(X_k))}$$

$$\dfrac{\dfrac{}{BAL(L)\quad X_1H_1 + \cdots + X_kH_k = F \quad \vdots \quad E_1H_1 + \cdots + E_kH_k = F'}\ \mathrm{C}}{E_1(F \cdot w(X_1)) + \cdots + E_k(F \cdot w(X_k)) = F'}$$

where C is the condition

1. There are precisely $m = \max\{|w(X_i)| : E_i \neq \emptyset \text{ for } 1 \leqslant i \leqslant k\}$ applications of UNF between the top goal and the bottom goal, and no application of any other rule, and if u is the associated word with this sequence of UNFs then $E_i = (X_i \cdot u)$ for each $i : 1 \leqslant i \leqslant k$.

Fig. 3. The rules BAL(L) and BAL(R).

reduction to the subgoal

$$E_1(F \cdot w(X_1)) + \cdots + E_k(F \cdot w(X_k)) = F'_1G_1 + \cdots + F'_nG_n.$$

The left-hand configuration (which is admissible by Proposition 4) has the following matrix form:

$$\begin{array}{l} E_1(\beta_1 \cdot w(X_1))G_1 + \cdots + E_1(\beta_n \cdot w(X_1))G_n + \\ \qquad\qquad \vdots \qquad\qquad\qquad\qquad \vdots \\ E_k(\beta_1 \cdot w(X_k))G_1 + \cdots + E_k(\beta_n \cdot w(X_k))G_n \end{array}$$

Let $E'_i = E_1(\beta_i \cdot w(X_1)) + \cdots + E_k(\beta_i \cdot w(X_k))$. It follows that $|E'_i| \leqslant 3\mathrm{M} + 2$ because for each j and i, $|\beta_j \cdot w(X_i)| \leqslant 2\mathrm{M} + 1$ and $|E_i| \leqslant \mathrm{M} + 1$. Therefore, the subgoal has the form $E'_1G_1 + \cdots + E'_nG_n = F'_1G_1 + \cdots + F'_nG_n$ whose maximum imbalance is $3\mathrm{M} + 2$. This bound on imbalance is independent of the sizes of the tails G_i.

Example 4. The instance of BAL(L) below uses the starting goal of Example 3:

$$\dfrac{\dfrac{BA + C = B(BA + C) + C}{BBA + BC + C = BB(BA + C) + BC + C}}{BB(BA + C) + BC + C = BB(BA + C) + BC + C}\ \mathrm{BAL(L)}$$

Here $w(B)=a$ and $w(C)=b$ and so $m=1$. The second goal is the result of UNF when the label is c. Assume $F=B(BA+C)+C$. The left configuration in the subgoal is $BB(F\cdot a)+(BC+C)(F\cdot b)$.

Next, we establish soundness and completeness of BAL(L) and BAL(R). Completeness is straightforward, if the goals are true then so is the subgoal.

Proposition 7. *If* $X_1H_1+\cdots+X_kH_k\sim F$ *and* $E_1H_1+\cdots+E_kH_k\sim F'$ *then* $E_1(F\cdot w(X_1))+\cdots+E_k(F\cdot w(X_k))\sim F'$.

Proof. Let E be $X_1H_1+\cdots+X_kH_k$ and assume $E\sim F$. From Fact 6.1 because $E\stackrel{w(X_i)}{\rightarrow}H_i$ it follows that $H_i\sim F\cdot w(X_i)$. Assume $E_1H_1+\cdots+E_kH_k\sim F'$. Using Facts 6.4 and 6.7 it follows that $E_1(F\cdot w(X_1))+\cdots+E_k(F\cdot w(X_k))\sim F'$. □

Soundness of the rules is more intricate. First, we explain "global" soundness of the proof system. The overall idea is that if there is a successful tableau whose root is false then there is a path through the tableau within which each subgoal is false. The idea is refined using approximants. If the root is false then there is an offending path (of false goals) through the tableau within which the approximant indices decrease whenever rule UNF has been applied, and hence this would mean that a successful final goal is false (which, as we shall show, is impossible). Soundness of a conditional rule is that if the premises are on an offending path then the subgoal preserves the falsity index of the final premise. In the case of BAL(R) assume that the offending path passes through the premise goals. There is a least n such that for the initial premise $F\sim_n X_1H_1+\cdots+X_kH_k$ and $F\not\sim_{n+1}X_1H_1+\cdots+X_kH_k$. As there are m applications of UNF between the initial and final premise it follows that $F'\sim_{n-m}E_1H_1+\cdots+E_kH_k$. However, because this is the offending path $F'\not\sim_{(n+1)-m}E_1H_1+\cdots+E_kH_k$. Soundness is that we may conclude $F'\not\sim_{(n+1)-m}E_1(F\cdot w(X_1))+\cdots+E_k(F\cdot w(X_k))$.

Proposition 8. *If* $X_1H_1+\cdots+X_kH_k\sim_n F$ *and* $E_1H_1+\cdots+E_kH_k\not\sim_{(n+1)-m}F'$ *where* $m=\max\{|w(X_i)|:E_i\neq\emptyset\}$ *then* $E_1(F\cdot w(X_1))+\cdots+E_k(F\cdot w(X_k))\not\sim_{(n+1)-m}F'$.

Proof. Assume $X_1H_1+\cdots+X_kH_k\sim_n F$ and $E_1H_1+\cdots+E_kH_k\not\sim_{(n+1)-m}F'$. If $E_i=\emptyset$ then clearly $E_i(F\cdot w(X_i))\sim_{(n+1)-m}E_iH_i$. If $E_i\neq\emptyset$ then by assumption $|E_i|>0$ and $|w(X_i)|\leqslant m$. However $X_1H_1+\cdots+X_kH_k\stackrel{w(X_i)}{\rightarrow}H_i$ and therefore $H_i\sim_{n-m}(F\cdot w(X_i))$. By Fact 6.9 $E_i(F\cdot w(X_i))\sim_{(n+1)-m}E_iH_i$. Therefore by Fact 6.5 $E_1(F_1\cdot w(X_1))+\cdots+E_k(F\cdot w(X_k))\sim_{(n+1)-m}E_1H_1+\cdots+E_kH_k$, and now the result follows using Fact 6.6. □

4.3. CUT

Bounding imbalance between configurations is not enough for showing decidability. The sizes of subgoals may keep growing.

Example 5. To illustrate this consider the following derivation[10] where A, A', B and B' are from Example 2:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots \quad \vdots}{AAA'A' + BBB' = A'A'A'A' + B'B'B'B'}\ \text{UNF}}{AAAA'A' + BBBB' = A'A'A'A'A' + B'B'B'B'B'}\ \text{BAL(L)}}{AAA'A'A' + BBB'B' = A'A'A'A'A' + B'B'B'B'B'}\ \text{UNF}}{AAAA'A'A' + BBBB'B' = A'A'A'A'A'A' + B'B'B'B'B'B'}\ \text{BAL(L)}}{AAA'A'A'A' + BBB'B'B' = A'A'A'A'A'A'A' + B'B'B'B'B'B'B'}\ \text{UNF}}{AAA + BB = A'A'A' + B'B'}\ \text{UNF}}{AAAA + BBB = A'A'A'A' + B'B'B'B'}\ \text{BAL(L)}$$

The application of UNF here only tracks the result of "after c". The derivation will go on for ever with increasing size of subgoals.

The next and crucial step in the argument is a mechanism for controlling size. It is at this point that we appeal to recursive nonterminals. The balanced goal, $E_1G_1 + \cdots + E_nG_n = F_1G_1 + \cdots + F_nG_n$ can be reduced to a subgoal $E_1V_1 + \cdots + E_nV_n = F_1V_1 + \cdots + F_nV_n$ where $(V_1, \ldots, V_n)$ is a family of recursive nonterminals. The mechanism for goal reduction constructs the recursive family $(V_1, \ldots, V_n)$ from a subsidiary family of goals, $E_1^iG_1 + \cdots + E_n^iG_n = F_1^iG_1 + \cdots + F_n^iG_n$ where $i \geqslant 1$, which have the same tails as the goal.

Before introducing the exact rule, CUT, we develop some results.

Lemma 1. *If $k \geqslant 1$ and $E_1^iG_1 + \cdots + E_n^iG_n \sim F_1^iG_1 + \cdots + F_n^iG_n$ for each $i : 1 \leqslant i \leqslant k$ then there is a family of recursive nonterminals $(V_1, \ldots, V_n)$ such that*
1. *$E_1^iV_1 + \cdots + E_n^iV_n \sim F_1^iV_1 + \cdots + F_n^iV_n$ for each $i : 1 \leqslant i \leqslant k$.*
2. *If $V_i \stackrel{\text{def}}{=} H_1V_1 + \cdots + H_nV_n$ then $G_i \sim H_1G_1 + \cdots + H_nG_n$.*
3. *If $V_i \stackrel{\text{def}}{=} V_j$ then $G_i \sim G_j$.*

Proof. The proof proceeds by iteratively refining families of recursive nonterminals for each $E_1^iG_1 + \cdots + E_n^iG_n \sim F_1^iG_1 + \cdots + F_n^iG_n$ in order starting with $i = 1$. Let E be $E_1^1G_1 + \cdots + E_n^1G_n$ and let F be $F_1^1G_1 + \cdots + F_n^1G_n$. For the base case $V_i^0 \stackrel{\text{def}}{=} V_i^0$, $1 \leqslant i \leqslant n$. Clearly 2 and 3 hold for each V_i^0. Assume that the jth family $(V_1^j, \ldots, V_n^j)$, $j \geqslant 0$, is given and that 2 and 3 hold for each V_i^j. Let E' be $E_1^1V_1^j + \cdots + E_n^1V_n^j$ and let F' be $F_1^1V_1^j + \cdots + F_n^1V_n^j$, which are both admissible by Proposition 4. If $E' \sim F'$ then we have dealt with the first equation. Now let E be $E_1^2G_1 + \cdots + E_n^2G_n$ and F be $F_1^2G_1 + \cdots + F_n^2G_n$ and let E' be $E_1^2V_1^j + \cdots + E_n^2V_n^j$ and let F' be $F_1^2V_1^j + \cdots + F_n^2V_n^j$. If $E' \sim F'$ then we have dealt with the second equation too. We keep repeating this until either all the equations are exhausted (and then $(V_1^j, \ldots, V_n^j)$ is the required family of recursive nonterminals) or E is $E_1^lG_1 + \cdots + E_n^lG_n$ and F is $F_1^lG_1 + \cdots + F_n^lG_n$ and E' is

[10] Although the example is very contrived, it does provide a very simple illustration.

$E_1^l V_1^j + \cdots + E_n^l V_n^j$ and F' is $F_1^l V_1^j + \cdots + F_n^l V_n^j$ and $E' \nsim_k F'$ for a least k. Let u be the smallest distinguishing word for E' and F'. There are two possibilities by Proposition 3 and Fact 3. First that one and only one of $(E' \cdot u)$ and $(F' \cdot u)$ is $\emptyset$. Second is that just one of this pair is a particular terminating nonterminal V_i^j. We show below that the first possibility is impossible because $E \sim F$. In the case of the second possibility we refine the family of recursive nonterminals to $(V_1^{j+1}, \ldots, V_n^{j+1})$ where each V_i^{j+1} obeys conditions 2 and 3. By Proposition 6, $E_1^i V_1^{j+1} + \cdots + E_n^i V_n^{j+1} \sim F_1^i V_1^{j+1} + \cdots + F_n^i V_n^{j+1}$ for all $i < l$. Hence we continue the construction for E is $E_1^l G_1 + \cdots + E_n^l G_n$ and F be $F_1^l G_1 + \cdots + F_n^l G_n$ and E' is $E_1^l V_1^{j+1} + \cdots + E_n^l V_n^{j+1}$ and F' is $F_1^l V_1^{j+1} + \cdots + F_n^l V_n^{j+1}$.

We now examine the case when $E' \nsim_k F'$ and $u = a_1 \ldots a_k$ is the smallest distinguishing word. Consider the following four sequences when Z is E', F', E and F, respectively,

$$(Z \cdot a_1), \ldots, (Z \cdot a_1 \ldots a_i), \ldots, (Z \cdot a_1 \ldots a_k)$$

Consider the initial part of the sequence in the case Z is E' up to the first prefix, if there is one, $u_1 = a_1 \ldots a_m$ such that $Z \cdot u_1 = E''$ where $E'' = H_1 V_1^j + \cdots + H_n V_n^j$ and $(E' \cdot a_1 \ldots a_{m-1}) \xrightarrow{a_m} V_i^j$. From 2 we know that $G_i \sim H_1 G_1 + \cdots + H_n G_n$ because $V_i^j \stackrel{\text{def}}{=} E''$. The initial part of the sequence when Z is E up to $E \cdot a_1 \ldots a_{m-1}$ is similar to the initial part for Z is E' in that they have the same "heads". Consequently $E \cdot u_1 = G_i$. Therefore the sequence for Z is E is updated from position m to k. Let $E \cdot a_1 \ldots a_s$, for $s \geqslant m$, be $(H_1 G_1 + \cdots + H_n G_n) \cdot a_{m+1} \ldots a_s$. This updating restores the same heads in the two sequences Z is E and Z is E' until the next occurrence of a $G_{i'}$ in the updated sequence for Z is E. We repeatedly update the new sequence for Z is E whenever there is a later position $E' \cdot a_1 \ldots a_t = H_1' V_1^j + \cdots + H_n' V_n^j$ and $V_{i'}^j \stackrel{\text{def}}{=} H_1' V_1^j + \cdots + H_n' V_n^j$ and $E \cdot a_1 \ldots a_t$ in the (updated) sequence is $G_{i'}$ for $t < k$. The same updating construction is applied to the sequences when Z is F' and Z is F. Note that repeated updating of the sequences for E and F does not affect the property that their corresponding positions are equivalent.

The final positions of the sequences for E' and F' are the elements $E' \cdot u$ and $F' \cdot u$. If one of them is $\emptyset$ then one of the final positions of the updated sequences for E and F is also $\emptyset$, which would contradict that $E \sim F$. Therefore one of them is a terminating recursive nonterminal V_i^j. Without loss of generality assume that $E' \cdot u = V_i^j$. Consider the final element of the updated sequence for E. It is either G_i or G_t when $V_t^j \stackrel{\text{def}}{=} V_i^j$ (and $i \leqslant t$). In the second case by 2, $G_t \sim G_i$. Consider now the final element $F' \cdot u$.

The first case is that $F' \cdot u$ is $H_1' V_1^j + \cdots + H_n' V_n^j$ where each $H_i' \neq \varepsilon$ and in the updated sequence $F \cdot u$ is $H_1' G_1 + \cdots + H_n' G_n$. Because $E \sim F$ it follows that $G_i \sim H_1' G_1 + \cdots + H_n' G_n$. The family $(V_1^j, \ldots, V_n^j)$ is refined to $(V_1^{j+1}, \ldots, V_n^{j+1})$ as follows. First $V_i^{j+1} \stackrel{\text{def}}{=} H_1' V_1^{j+1} + \cdots + H_n' V_n^{j+1}$. Next for any index t such that $V_t^j \stackrel{\text{def}}{=} V_i^j$ let $V_t^{j+1} \stackrel{\text{def}}{=} H_1' V_1^{j+1} + \cdots + H_n' V_n^{j+1}$. For the other entries we merely update the index j to $j+1$ on the V_i^js on both sides of $\stackrel{\text{def}}{=}$. By construction properties 2 and 3 both hold for the new family $(V_1^{j+1}, \ldots, V_n^{j+1})$.

The second case is that $F' \cdot u = V^j_{i'}$. Therefore, the final element $F \cdot u$ in the updated sequence is either $G_{i'}$ or $G_{t'}$ such that $G_{t'} \sim G_{i'}$ where $V^j_{t'} \stackrel{\text{def}}{=} V^j_{i'}$ and $i' \leqslant t'$. Because $E' \cdot u = V^j_i$ we know that $i \neq i'$ since u distinguishes E' and F'. However $G_i \sim G_{i'}$ because $E \sim F$. Consider $\min\{i, i'\}$. Without loss of generality assume it is i'. The refined family of recursive nonterminals $(V^{j+1}_1, \ldots, V^{j+1}_n)$ is defined as follows. Firstly, $V^{j+1}_i \stackrel{\text{def}}{=} V^{j+1}_{i'}$. Secondly for any index t such that $V^j_t \stackrel{\text{def}}{=} V^j_i$ let $V^{j+1}_t \stackrel{\text{def}}{=} V^{j+1}_{i'}$. For the rest of the entries we just update the index j to $j+1$ as in the first case. By construction, properties 2 and 3 hold for the new family of recursive nonterminals.

The stages of the construction produce a sequence of families of recursive nonterminals $(V^0_1, \ldots, V^0_n), \ldots, (V^j_1, \ldots, V^j_n), \ldots$ where each family refines the previous family. The final step in the proof is that the iteration must terminate by stage $n-1$. At each stage j exactly one terminating nonterminal V^j_i is directly refined. Other elements V^j_t when $V^j_t \stackrel{\text{def}}{=} V^j_i$ and $t > i$ may also be refined. No element V^k_i with index i is directly refined more than once. Therefore by stage $n-1$ the iteration must terminate with the family $(V^{n-1}_1, \ldots, V^{n-1}_n)$. Now it is a simple argument that if property 1 does not hold by stage $n-1$ then afterall $E \nsim F$ for the then current E and F. □

The recursive family $(V_1, \ldots, V_n)$ as constructed in the proof of Lemma 1 is said to be "canonical" for the family $E^i_1 G_1 + \cdots + E^i_n G_n \sim F^i_1 G_1 + \cdots + F^i_n G_n$. The construction of canonical recursive nonterminals is independent of the tails G_i.

Fact 7. *If $(V_1, \ldots, V_n)$ is canonical for $E^i_1 G_1 + \cdots + E^i_n G_n \sim F^i_1 G_1 + \cdots + F^i_n G_n$ then it is also canonical for the family $E^i_1 J_1 + \cdots + E^i_n J_n \sim F^i_1 J_1 + \cdots + F^i_n J_n$, where $i : 1 \leqslant i \leqslant k$.*

Example 6. To illustrate Lemma 1 and Fact 7 consider the following schematic example, $(*)$ $AG_1 + BG_2 + CG_3 \sim A'G_1 + B'G_2 + C'G_3$. Assume the following transitions:

$$A \stackrel{a}{\to} \varepsilon, \quad B \stackrel{b}{\to} \varepsilon, \quad C \stackrel{c}{\to} \varepsilon,$$

$$A' \stackrel{a}{\to} \varepsilon, \quad B' \stackrel{b}{\to} \varepsilon, \quad C' \stackrel{c}{\to} \varepsilon.$$

However also assume $A \stackrel{d^n}{\to} \varepsilon$ for arbitrary $n > 0$, and $B' \stackrel{d}{\to} B'$ and $C' \stackrel{d}{\to} C'$. Therefore $G_1 \sim B'G_2 + C'G_3$ and so assume $G_1 \stackrel{d}{\to} G_1$ and $G_1 \stackrel{b}{\to} G_2$ and $G_1 \stackrel{c}{\to} G_3$. The admissible configurations G_2 and G_3 can be arbitrary. Consider the construction of a canonical family of recursive nonterminals for $(*)$. Initially $V^0_i \stackrel{\text{def}}{=} V^0_i$ for $i : 1 \leqslant i \leqslant 3$. However $AV^0_1 + BV^0_2 + CV^0_3 \nsim A'V^0_1 + B'V^0_2 + C'V^0_3$. The smallest distinguishing word is d^n, because $(AV^0_1 + BV^0_2 + CV^0_3) \cdot d^n = V^0_1$ and $(A'V^0_1 + B'V^0_2 + C'V^0_3) \cdot d^n = B'V^0_2 + C'V^0_3$. So the recursive nonterminals are refined, $V^1_1 \stackrel{\text{def}}{=} \emptyset V^1_1 + B'V^1_2 + C'V^1_3$ and $V^1_i \stackrel{\text{def}}{=} V^1_i$ for $i : 2 \leqslant i \leqslant 3$. The family (V^1_1, V^1_2, V^1_3) is canonical for $(*)$.

The assembly of a canonical family proceeds in stages. Each recursive family $(V^{j+1}_1, \ldots, V^{j+1}_n)$ refines $(V^j_1, \ldots, V^j_n)$ and the construction must terminate by stage $j = n-1$. The building of the V^{j+1}_i's from the V^j_i's appeals to the smallest distinguishing word u_{j+1} for $E' \nsim F'$ (when E' is $E^i_1 V^j_1 + \cdots + E^i_n V^j_n$ and F' is $F^i_1 V^j_1 + \cdots + F^i_n V^j_n$). We have

no insight as to the upper bound on $|u_{j+1}|$. For instance, as Example 7 illustrates, it is not determined by the maximum norm of the heads E_j^i and F_j^i. Indeed this turns out to be the reason why we cannot offer a complexity bound for the decision procedure.

Lemma 1 is not usable directly in the tableau proof system because it presupposes that the configurations are equivalent. We need to consider how to introduce canonical recursive nonterminals for goals which need not be true. The idea is to approximate canonicity by defining when a recursive family $(V_1,\ldots,V_n)$ is an "m-unifier", $m\geqslant 0$, for a family of pairs of heads $(E_1^i+\cdots+E_n^i, F_1^i+\cdots+F_n^i)$ of goals $E_1^iG_1+\cdots+E_n^iG_n = F_1^iG_1+\cdots+F_n^iG_n$ with common tails, when i: $1\leqslant i\leqslant k$. For the construction we assume that $E_1^iG_1+\cdots+E_n^iG_n \sim_m F_1^iG_1+\cdots+F_n^iG_n$, for each i, which guarantees that there is an m-unifier for these heads. The construction is iterative (and closely follows the proof of Lemma 1). We define stages $(V_1^j,\ldots,V_n^j)$ with depth d_j. Initially, we have $(V_1^0,\ldots,V_n^0)$ with depth $d_0=0$. Assume we have stage $j\geqslant 0$, $(V_1^j,\ldots,V_n^j)$ with depth $d_j\leqslant m$. There are two cases to consider.

1. $E_1^iV_1^j+\cdots+E_n^iV_n^j \sim_{m-d_j} F_1^iV_1^j+\cdots+F_n^iV_n^j$ for each i: $1\leqslant i\leqslant k$. In which case $(V_1^j,\ldots,V_n^j)$ is the required m-unifier.
2. $E_1^lV_1^j+\cdots+E_n^lV_n^j \nsim_{m-d_j} F_1^lV_1^j+\cdots+F_n^lV_n^j$, where l is the smallest index in $\{1,\ldots,k\}$. Assume that this goal is $E'=F'$. The family of V_i^js is updated to $(V_1^{j+1},\ldots,V_n^{j+1})$ as in Lemma 1 but with depth d_{j+1}. Assume that u, where $|u|\leqslant m-d_j$, is the smallest word which distinguishes between E' and F'. Because $E_1^iG_1+\cdots+E_n^iG_n \sim_m F_1^iG_1+\cdots+F_n^iG_n$ for each i, it is not possible for one of $E'\cdot u$ and $F'\cdot u$ to be $\emptyset$. Without loss of generality assume that $E'\cdot u = V_i^j$ and $V_i^j \stackrel{\text{def}}{=} V_i^j$. There are two subcases:
   (a) $F'\cdot u = H_1'V_1^j+\cdots+H_n'V_n^j$ where each $H_i'\neq\varepsilon$. Therefore we set $V_i^{j+1}\stackrel{\text{def}}{=} H_1'V_1^{j+1}+\cdots+H_n'V_n^{j+1}$. And for any index t such that $V_t^j\stackrel{\text{def}}{=}V_i^j$ we also let $V_t^{j+1}\stackrel{\text{def}}{=}H_1'V_1^{j+1}+\cdots+H_n'V_n^{j+1}$. The other entries of V^{j+1} just have their indices j updated to $j+1$. The result is the family $(V_1^{j+1},\ldots,V_n^{j+1})$ whose depth is $d_{j+1}=d_j+|u|$ (which by definition is no more than m).
   (b) $F'\cdot u = V_{i'}^j$ and $i\neq i'$. Without loss of generality assume that i' is $\min\{i,i'\}$. $V_i^{j+1}\stackrel{\text{def}}{=}V_{i'}^{j+1}$ and for any index t such that $V_t^j\stackrel{\text{def}}{=}V_i^j$, $V_t^{j+1}\stackrel{\text{def}}{=}V_{i'}^{j+1}$. The remaining entries of V^{j+1} just have their indices j updated to $j+1$. The result is the family $(V_1^{j+1},\ldots,V_n^{j+1})$ whose depth is $d_{j+1}=d_j+|u|$.

In the case of Example 6 the initial family of recursive nonterminals has depth 0, and the final family is an n-unifier for the pair of heads $(A+B+C, A'+B'+C')$.

For $m\geqslant 0$ it is decidable whether $E_1^iG_1+\cdots+E_n^iG_n \sim_m F_1^iG_1+\cdots+F_n^iG_n$ because there are only finitely many words u of size at most m to examine. For the same reason the construction of an m-unifier for a family of heads $(E_1^i+\cdots+E_n^i, F_1^i+\cdots+F_n^i)$ is effective. The next result provides a clear relationship between canonicity and m-unification.

Fact 8. *If $(V_1,\ldots,V_n)$ is canonical for the family $E_1^iG_1+\cdots+E_n^iG_n \sim F_1^iG_1+\cdots+F_n^iG_n$ then there exists $m'\geqslant 0$ such that for all $m\geqslant m'$, $(V_1,\ldots,V_n)$ is an m-unifier for the heads $(E_1^i+\cdots+E_n^i,\ F_1^i+\cdots+F_n^i)$.*

$$\textit{CUT}\quad \frac{\begin{array}{c} E_1^1G_1+\cdots+E_n^1G_n = F_1^1G_1+\cdots+F_n^1G_n \\ \vdots \\ E_1^kG_1+\cdots+E_n^kG_n = F_1^kG_1+\cdots+F_n^kG_n \\ \vdots \\ E_1G_1+\cdots+E_nG_n = F_1G_1+\cdots+F_nG_n \end{array}\ \mathrm{C}}{E_1V_1+\cdots+E_nV_n = F_1V_1+\cdots+F_nV_n}$$

where C is the condition

1. $E_1^iG_1+\cdots+E_n^iG_n \sim_m F_1^iG_1+\cdots+F_n^iG_n$ and $(V_1,\ldots,V_n)$ is an m-unifier for the family $(E_1^i+\cdots+E_n^i, F_1^i+\cdots+F_n^i)$ for $1\leqslant i\leqslant k\leqslant n$.
2. Between the goal $E_1^kG_1+\cdots+E_n^kG_n = F_1^kG_1+\cdots+F_n^kG_n$ and the bottom goal there are at least m applications of UNF (as well as possible applications of BAL(L) and BAL(R)).

Fig. 4. The rule CUT.

The final tableau proof rule CUT, presented in Fig. 4, is a conditional rule. This rule cuts common tails of a goal and replaces them with recursive nonterminals. Unlike the BAL rules the number of premises of an application of CUT varies, but is at most $n+1$ where n is the number of common "tails" G_i.

Example 7. We show how CUT applies in the case of Example 5:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{AAA+BB=A'A'A'+B'B'}{AAAA+BBB=A'A'A'A'+B'B'B'}\ \mathrm{UNF}}{(*)\ AAA'A'+BBB'=A'A'A'A'+B'B'B'}\ \mathrm{BAL(L)}}{(**)\ AAAA'A'+BBBB'=A'A'A'A'A'+B'B'B'B'}\ \mathrm{UNF}}{AAAV_1+BBBV_2=A'A'A'V_1+B'B'B'V_2}\ \mathrm{CUT}$$

Here $V_i \stackrel{\mathrm{def}}{=} V_i$ for $i\in\{1,2\}$, and so (V_1,V_2) is a 0-unifier for the head $(AA+BB, A'A'+B'B')$. The initial goal for CUT is $(*)$ and the final goal is $(**)$. There are at least 0 applications of UNF between these two goals. In this case $k=1$.

Completeness of CUT is more subtle than for BAL and UNF, because the rule does not guarantee preservation of equivalence for all possible applications. However this is not an impediment because, as we show below, the rule is sound. Assume that all the premises are true goals. If $(V_1,\ldots,V_n)$ is canonical for the initial k-premises then by Fact 8 there is an m such that $(V_1,\ldots,V_n)$ is an m-unifier for the "heads"

$(E_1^i+\cdots+E_n^i, F_1^i+\cdots+F_n^i)$. If it is also true that $E_1V_1+\cdots+E_nV_n \sim F_1V_1+\cdots+F_nV_n$ then completeness is assured. Otherwise if $E_1V_1+\cdots+E_nV_n \nsim F_1V_1+\cdots+F_nV_n$ then $(V_1,\ldots,V_n)$ can be refined to $(V_1'\ldots,V_n')$ so it is canonical for all the premises (and the refined family is then an m'-unifier for the family of heads). By Lemma 1 there can be at most $n-1$ refinements, and so eventually CUT is applicable with at most $n+1$ premises if true goals with common tails persist, as we show in the next section.

For soundness of CUT the idea is the same as for the BAL rules. If the premises of an application of CUT are on an offending path of false goals then the subgoal preserves the falsity index of the final premise.

Proposition 9. *If $d \geqslant m$ and $E_1^iG_1+\cdots+E_n^iG_n \sim_d F_1^iG_1+\cdots+F_n^iG_n$ for $1 \leqslant i \leqslant k$ and $(V_1,\ldots,V_n)$ is an m-unifier for the heads $(E_1^i+\cdots+E_n^i, F_1^i+\cdots+F_n^i)$ and $m \leqslant m' \leqslant d$ and $E_1G_1+\cdots+E_nG_n \nsim_{(d+1)-m'} F_1G_1+\cdots+F_nG_n$ then $E_1V_1+\cdots+E_nV_n \nsim_{(d+1)-m'} F_1V_1 + \cdots + F_nV_n$.*

Proof. Assume $d \geqslant m$ and $(V_1,\ldots,V_n)$ is an m-unifier for the family $(E_1^i+\cdots+E_n^i, F_1^i+\cdots+F_n^i)$ and $E_1^iG_1+\cdots+E_n^iG_n \sim_d F_1^iG_1+\cdots+F_n^iG_n$ for each i: $1 \leqslant i \leqslant k$. By construction of the m-unifier it follows that if $V_i \stackrel{\text{def}}{=} H_1V_1+\cdots+H_nV_n$ then $G_i \sim_{d-m} H_1G_1+\cdots+H_nG_n$ and if $V_i \stackrel{\text{def}}{=} V_j$ then $G_i \sim_{d-m} G_j$. Next, assume that $m' \leqslant d$ and $m' \geqslant m$ and $E \nsim_{(d+1)-m'} F$ where E is $E_1G_1+\cdots+E_nG_n$ and F is $F_1G_1+\cdots+F_nG_n$, but $E' \sim_{(d+1)-m'} F'$ where E' is $E_1V_1+\cdots+E_nV_n$ and F' is $F_1V_1+\cdots+F_nV_n$. Consider the smallest word $u=a_1\ldots a_l$ which distinguishes between E and F. Note that $E\cdot a_1\ldots a_i \nsim_{(d+1)-(m'+i)} F\cdot a_1\ldots a_i$ and $E'\cdot a_1\ldots a_i \sim_{(d+1)-(m'+i)} F'\cdot a_1\ldots a_i$. Consider the following four sequences when Z is E, F, E' and F': $(Z\cdot a_1),\ldots,(Z\cdot a_1\ldots a_{l'})$ where either $l'=l$ or $l'<l$ and the final elements for the sequences when Z is E' and F' is a terminating nonterminal V_i. The idea is as in the proof of Lemma 1 to update the sequences for Z is E and Z is F so that they have the same heads as those for Z is E' and Z is F'. Consider the initial prefix $u_1=a_1\ldots a_i$, $i>0$, of Z is E', if there is one, such that $Z\cdot u_1=E''$ and $E''=H_1V_1+\cdots+H_nV_n$ and $(E'\cdot a_1\ldots a_{i-1}) \xrightarrow{a_i} V_j$. Hence $E\cdot u_1=G_j$. Because $V_j \stackrel{\text{def}}{=} E''$, $G_j \sim_{d-m'} H_1G_1+\cdots+H_nG_n$ and $G_j \nsim_{(d+1)-(m'+i)} F\cdot u_1$. Therefore using Facts 6.9, 6.3 and 6.5 $H_1G_1+\cdots+H_nG_n \nsim_{(d+1)-(m'+i)} F\cdot u_1$. The sequence when Z is E is updated from position i to l': $E\cdot a_1\ldots a_s$, $s>i$, becomes $(H_1G_1+\cdots+H_nG_n)\cdot a_{i+1}\ldots a_s$. This updating restores the same heads in the two sequences for Z is E and Z is E' until the next occurrence of a $G_{i'}$ in the first sequence in which case we then again update it. The same updating construction is applied to the sequence Z is F using Z is F'. The repeated updating of the sequences for E and F does not affect the property that their corresponding positions j are inequivalent at $(d+1)-(m'+j)$. Consider now the final elements in the updated sequences for E and F. The first case is that one and only one of the elements is $\emptyset$, but then one and only one of the corresponding elements in the sequences for E' and F' is also $\emptyset$ which is a contradiction. The second case is that one of the elements, say in the sequence for E, is a terminating recursive nonterminal U_j, which means that some G_i is U_j. But then the corresponding element

Successful final goals :

$$\begin{array}{l} E = F \\ \quad\vdots \qquad \text{UNF} \\ \quad\vdots \qquad \text{at least once} \\ 1.\ E = E, \quad 2.\ E = F. \end{array}$$

Unsuccessful final goals :

1. $\emptyset = E$ or $E = \emptyset$ and $E \neq \emptyset$,
2. $V_i = V_j$ when $i \neq j$ and $V_i \stackrel{\text{def}}{=} V_i$ and $V_j \stackrel{\text{def}}{=} V_j$.

Fig. 5. Final goals.

in the sequence for E' is also a terminating nonterminal V_i (because $G_i \sim_{d-m} H$ iff H is G_i). Therefore, the corresponding element in the sequence for F' is also V_i and so the final element in the sequence for F is U_j as well. □

5. Correctness of tableaux

In the previous section we introduced the rules of the tableau proof system. There are just four rules, UNF, BAL(L), BAL(R) and CUT. In each case we described the rule and its soundness and completeness. There is also the important notion of when a current goal counts as final. Final goals are classified as either successful or unsuccessful. A *tableau proof* for a starting Goal is a finite proof tree, whose root is Goal and all of whose leaves are successful final goals, and all of whose inner subgoals are the result of an application of one of the rules.

Final goals are presented in Fig. 5. Unsuccessful goals are clearly false. A final goal is successful if it is either an identity or a repeat. An offending path of false goals with decreasing falsity indices cannot include either kind of successful goal. Clearly it is not possible for $E \not\sim_m E$. For the other case, suppose the offending path passes through $E = F$ twice. At the first instance there is an m, $E \sim_m F$ and $E \not\sim_{m+1} F$, but as there is at least one application of UNF between the two occurrences this would imply that $E \not\sim_m F$, which is a contradiction.

The first main result is that a successful tableau for $E = F$ indeed constitutes a proof that $E \sim F$.

Theorem 1. *If there is a successful tableau for $E = F$ then $E \sim F$.*

Proof. Suppose there is a successful tableau for $E = F$ but $E \not\sim F$. Then there is a least n such that $E \not\sim_n F$. We now construct an offending path of false goals through the tableau within which the approximant indices decrease whenever UNF is applied.

But this is impossible, for we must reach a successful final goal because the tableau is finite. □

Example 8. Below is a successful tableau which establishes $pYX \sim pYYX$ of Fig. 1, where the determinised strict grammar is as in Example 1. A is $[pXp]$, B is $[pYp]$ and C is $[pYr]$:

$$\dfrac{\dfrac{(*)\ BA+C=B(BA+C)+C}{(\mathbf{1})\quad \varepsilon=\varepsilon \quad BBA+BC+C=BB(BA+C)+BC+C}\ \text{UNF}}{BBA+BC+C=BBA+BC+C}\ \text{BAL(R)}$$

where (**1**) is the subtableau

$$\dfrac{\dfrac{A=BA+C}{A=A \quad \varepsilon=\varepsilon \quad \dfrac{(**)\ A=B(BA+C)+C}{A=BA+C \quad \varepsilon=\varepsilon \quad \dfrac{A=BB(BA+C)+BC+C}{A=BBA+BC+C}\ \text{BAL(R)}}\ \text{UNF}}\ \text{UNF}}{}$$

This proof does not use CUT. The premise $(*)$ is the initial premise for the application of BAL(L), and $(**)$ is the initial premise for BAL(R). The leaf goals are either identities or repeats.

Example 9. Part of the successful tableau for $AAA+BB \sim A'A'A'+B'B'$ whose nonterminals belong to Example 2 is presented below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{AAA+BB=A'A'A'+B'B'}{\ldots \quad AAAA+BBB=A'A'A'A'+B'B'B'}\ \text{UNF}}{(*)\ AAA'A'+BBB'=A'A'A'A'+B'B'B'}\ \text{BAL(L)}}{\ldots \quad AAAA'A'+BBBB'=A'A'A'A'A'+B'B'B'B'}\ \text{UNF}}{AAAV_1+BBBV_2=A'A'A'V_1+B'B'B'V_2}\ \text{CUT}}{\ldots \quad AAAAV_1+BBBBV_2=A'A'A'A'V_1+B'B'B'B'V_2}\ \text{UNF}}{(**)\ AAA'A'V_1+BBB'B'V_2=A'A'A'A'V_1+B'B'B'B'V_2}\ \text{BAL(L)}}{\ldots \quad AAAA'A'V_1+BBBB'B'V_2=A'A'A'A'A'V_1+B'B'B'B'B'V_2}\ \text{UNF}}{AAAV_1+BBBV_2=A'A'A'V_1+B'B'B'V_2}\ \text{CUT}$$

Here $V_1 \stackrel{\text{def}}{=} V_1$ and $V_2 \stackrel{\text{def}}{=} V_2$, and these recursive nonterminals are introduced twice. (V_1, V_2) is a 0-unifier for the heads $(AA+BB, A'A'+B'B')$ in both goals $(*)$ and $(**)$. The rest of the tableau is finite for similar reasons.

The proof of the converse of Theorem 1, that if $E \sim F$ then there is a successful tableau for $E = F$, is more intricate. Given a true goal one applies the rules, preserving truth, according to the strategy described below. It is therefore not possible to reach an unsuccessful final goal. Thus, the main issue is how to guarantee that the tableau construction is finite. We show that on any infinite path of goals developed using the strategy there must be infinitely many successful final goals.

We start with a simple observation:

(1) For any $m \geqslant 0$, there are only finitely many different goals $E = F$ (whose recursive nonterminals belong to $(V_1, \ldots, V_n)$) with $|E| \leqslant m$ and $|F| \leqslant m$.

If F contains recursive nonterminals from the family $(V_1, \ldots, V_n)$ then $\mathrm{rec}(F)$ is the size of the largest definition in the family, $\max\{|H|\colon V_i \stackrel{\mathrm{def}}{=} H\}$. If F does not contain recursive nonterminals then $\mathrm{rec}(F)$ is 0. The next observation tells us how much a configuration can increase in size through an application of UNF.

(2) For any a, $|E \cdot a| \leqslant \max\{\mathrm{rec}(E), |E| + 1\}$.

The size of an application of BAL is the size of the configuration F in the initial goal of the rule (see Fig. 3), and the application is said to use the configuration F. The subgoal contains the configuration $E_1(F \cdot w(X_1)) + \cdots + E_k(F \cdot w(X_k))$. E_i is a "head" of the application of BAL and $(F \cdot w(X_i))$ is a "tail". The size of a head is bounded, $|E_i| \leqslant \mathrm{M} + 1$ (using (2)). Moreover by Proposition 5 if $E_i(F \cdot w(X_i)) \stackrel{u}{\rightarrow} (F \cdot w(X_i))$ then $(E_j(F \cdot w(X_j)) \cdot u) = \emptyset$ for $j \neq i$. Another observation about BAL (which uses (2)) is as follows:

(3) If $E' = F'$ is the result of an application of BAL of size m then $|E'|$, $|F'| \leqslant k + 2\mathrm{M} + 1$, where $k = \max\{m, \mathrm{rec}(E')\}$.

Let $\mathrm{S} = \mathrm{M}^2 + 4\mathrm{M} + 1$. A configuration F is "small" if $|F| \leqslant \mathrm{rec}(F) + \mathrm{S}$. The strategy is to apply the BAL rules wherever possible when the sizes of their applications are small, and otherwise to apply UNF. The rule CUT is not applied. Any infinite path of goals containing infinitely many small applications of BAL, and no application of CUT, must therefore contain infinitely many final goals ("repeats") by properties (3) and (1). Later we also show that any infinite path of goals with only finitely many applications of BAL must contain infinitely many final goals.

Next, suppose there is an application of BAL which uses a large F of size m. The strategy is now to build a "block". Assume that it is an application of BAL(L).

$$\begin{array}{lll} & & F \\ & & \vdots \quad \mathrm{BAL(L)} \\ (*) \qquad E_1(F \cdot w(X_1)) + \cdots + E_k(F \cdot w(X_k)) & = & F' \end{array}$$

F is the "root configuration" of the block and $(*)$ is its "root goal" (which will also be a potential root, the initial premise, of an application of CUT). If the block starts with BAL(L) then the strategy is to repeatedly apply BAL(L) wherever possible, and UNF otherwise.[11] However BAL(R) is permitted, once the "tail" of an application of

[11] If BAL(R) initiates the block then the strategy is to repeatedly apply BAL(R) and UNF.

$$
\begin{array}{rcll}
 & & F'' & \\
 & & \vdots & \text{BAL(L)} \\
E'_1(F'' \cdot w(X'_1)) + \cdots + E'_{k'}(F'' \cdot w(X'_{k'})) & = & H & \\
\vdots & & \vdots & \text{UNFs} \\
(F'' \cdot w(X'_i)) = G_1 & = & H_1 & \\
\vdots & & \vdots & \\
G_k & = & H_k &
\end{array}
$$

Fig. 6. A potential switch from BAL(L) to BAL(R).

BAL(L) is exposed, see Fig. 6. Assume an application of BAL(L) using F''. Between its result and the goal $G_1 = H_1$ there are no further applications of BAL(L), and G_1 is a tail of the BAL application. BAL(R) is now permitted provided it uses configuration G_i, $i \geqslant 1$. BAL(R) is not permitted using a configuration from a goal above $G_1 = H_1$. BAL(R) is not enforced, for one can still apply BAL(L). The strategy is always to apply a BAL rule whenever it is permitted. If BAL(R) is applied then the strategy is to repeatedly apply BAL(R), and to use UNF otherwise. BAL(L) is only permitted once a tail of an application of BAL(R) is the right-hand configuration of a goal. Thus, a block consists of alternating sub-blocks of BAL(L)s and UNFs and BAL(R)s and UNFs. If a later application of BAL is smaller than m then either a new block with a smaller root configuration is initiated or the size of the application is small and the earlier strategy applies.

Assume a root configuration F of size m with block root $E_1 = F_1$. Let π be a path of goals $E_1 = F_1, \ldots, E_l = F_l, \ldots$ belonging to the block developed from $E_1 = F_1$ using the strategy, where all applications of BAL have size at least m. We show the following crucial property:

(4) For every G which is used in an application of BAL in π there is a word u such that G is $(F \cdot u)$ and $|(F \cdot v)| > m - (\mathrm{M}^2 + 3\mathrm{M})$ for all prefixes v of u.

Property (4) holds for the initial root configuration F because F is $F \cdot \varepsilon$ and $|F| = m$. Assume the block is initiated with a BAL(L). Consider a later application of BAL(L) using F' (where there are no intervening applications of BAL(R)) as depicted in the left derivation of Fig. 7. F' arises from F via applications of UNF (and possibly BAL(L)). Consequently, there is a word u associated with the applications of UNF, and $F' = (F \cdot u)$, and by assumption $|F'| \geqslant m$. For every prefix v of u, $(F \cdot v)$ is a configuration on the path between F and F'. Assume that for one of these configurations F'', $|F''| \leqslant m - (\mathrm{M}^2 + 3\mathrm{M})$. There are two cases to examine.

The first case is that F'' occurs between a configuration used for a BAL and its application (between, for example, F and F_1 in Fig. 7). The second case is that F'' occurs at or after an application of BAL(L), between F_1 and F' in Fig. 7. Consider the first case. There are at most $\mathrm{M} - 1$ applications of UNF between F'' and the application of BAL (because F'' cannot be the configuration used in this application). Assume that

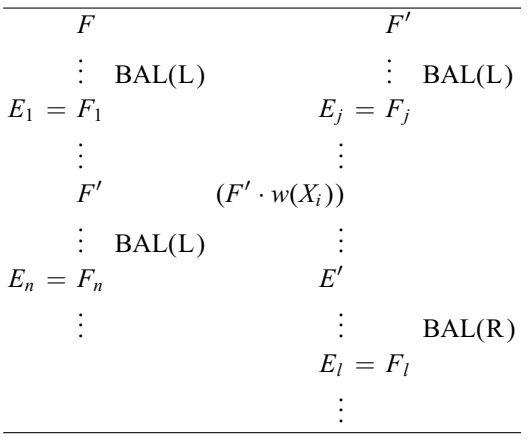

Fig. 7. Showing property (4).

$E_1 = F_1$ is the result of this BAL(L) which uses F and that F' is the next configuration used in an application of BAL(L). Because $|F''| \leqslant m-(\mathrm{M}^2+3\mathrm{M})$ and there are at most $\mathrm{M}-1$ applications of UNF between it and F_1, by (2) it follows that $|F_1| \leqslant m-(\mathrm{M}^2+2\mathrm{M}+1)$. Because $|F'| \geqslant m$, there must be at least $(\mathrm{M}^2+2\mathrm{M}+1)$ applications of UNF between F_1 and F' which are size increasing: for F_1 must increase its size and become F'. The second case is also covered by these observations: if F'' occurs between F_1 and F' then there must be at least $(\mathrm{M}^2+3\mathrm{M})$ applications of UNF between F_1 and F' which are size increasing. However within at most $\mathrm{M}^2+\mathrm{M}$ applications of UNF from F_1 a tail $(F \cdot w(X_i))$ of the application of BAL(L) must occur as the left-hand configuration of a goal. E_1 has the form $E'_1(F \cdot w(X_1))+\cdots+E'_k(F \cdot w(X_k))$ where each $|E'_i| \leqslant \mathrm{M}+1$. Because BAL(L) does not apply between F_1 and F', each E'_i's size must be declining. If E_1 in 1-head form is $Y_1H_1+\cdots+Y_lH_l$ then within M applications of UNF the left-hand configuration must be H_j for some j, and within another M applications of UNF H_j must lose its "head" nonterminals, and so on. Consequently, within $\mathrm{M}^2+\mathrm{M}$ applications of UNF between F_1 and F' a goal $(F \cdot w(X_i)) = F_k$ has to occur. F_1 may have increased in size in becoming F_k but only by $\mathrm{M}^2+\mathrm{M}$, and so $|F_k| \leqslant m-(\mathrm{M}+1)$. So there are still at least $\mathrm{M}+1$ applications of UNF between F_k and F' which are size increasing. However BAL(R) is now permitted, and clearly it must apply between F_k and F' because there must be a sequence of at most M UNFs where the initial right-hand configuration does not decrease in size. But this is a contradiction.

Next, we show that property (4) continues to hold when there is a switch from BAL(L) using F' to an application of BAL(R) using E', pictured on the right in Fig. 7. BAL(R) is only permitted when a tail $(F' \cdot w(X_i))$ of the application of BAL(L) occurs as a left-hand configuration. By assumption there is a word u such that F' is $(F \cdot u)$. Therefore the tail $(F' \cdot w(X_i))$ is $(F \cdot uw(X_i))$. There are no applications of BAL between this tail and E', and therefore there is a word v such that E' is $(F \cdot uw(X_i)v)$. Moreover both F' and E' have size at least m. Assume that for some prefix v' of

$w(X_i)v$, $|F' \cdot v'| \leqslant m - (\mathrm{M}^2 + 3\mathrm{M})$. There are two cases to consider. First is that v' is a prefix of $w(X_i)$, and secondly that it is a prefix of the form $w(X_i)\,v''$. Because $|w(X_i)| \leqslant \mathrm{M}$ for the first case this means that $|(F' \cdot w(X_i))| \leqslant m - (\mathrm{M}^2 + 2\mathrm{M})$. Therefore, there has to be at least $(\mathrm{M}^2 + 2\mathrm{M})$ applications of UNF between $(F' \cdot w(X_i))$ and E' which increase size, and for the second case there has to be at least $\mathrm{M}^2 + 3\mathrm{M}$ applications. However BAL(L) is still permitted between $(F' \cdot w(X_i))$ and E'. Clearly, BAL(L) must therefore apply to a configuration belonging to a goal strictly above E' because there must be a sequence of at most M UNFs where the initial left-hand configuration does not decrease in size.

The argument for (4) is now repeated for all further applications of BAL within π.

Using (4) we now establish a final property which shows that CUT eventually applies in a block. Assume a root configuration F of large size m with block root $E_1 = F_1$, and assume π is a path of goals belonging to this block developed using the strategy. By definition $|F| > \mathrm{rec}(F) + \mathrm{S}$ where S is $\mathrm{M}^2 + 4\mathrm{M} + 1$. Consider F in S-head normal form (as defined in Section 3). If F does not contain recursive nonterminals then $F = \beta_1 G_1 + \cdots + \beta_n G_n$ where $|\beta_i| = \mathrm{S}$ or $|\beta_i| < \mathrm{S}$ and $G_i = \varepsilon$. If F contains recursive nonterminals then it has a similar form $\beta_1 G_1 + \cdots + \beta_n G_n$ where $|\beta_i| = \mathrm{S}$ and $|G_i| \neq \varepsilon$ or $|\beta_i| < \mathrm{S}$ and G_i is a recursive nonterminal V. However using Fact 5 we can assume that if $|\beta_i| < \mathrm{S}$ then G_i is a terminating recursive nonterminal V_j, with $V_j \stackrel{\mathrm{def}}{=} V_j$. This form can be achieved by replacing recursive nonterminals with their definitions without increasing the size of F. If βV is a component of F and $|\beta| < \mathrm{S}$ and $V \stackrel{\mathrm{def}}{=} H$ then $|\beta H| < |F|$. Whether or not F contains recursive nonterminals, there is an upper bound on the "width" n (as it can be at most the number of sequences of nonterminals of the grammar whose length is at most S).

The last key property is as follows.

(5) The result of every application of BAL within π has the following tail form: $E_1 G_1 + \cdots + E_n G_n = F_1 G_1 + \cdots + F_n G_n$ where the G_i's are the tails of the root configuration F (in S-head normal form).

Condition (5) essentially follows from (4) and admissibility. The head of F, $\beta_1 + \cdots + \beta_n$, is admissible and each β_i has the form $X_1^i \ldots X_t^i$. Let β_i^j be the jth suffix $X_j^i \ldots X_t^i$ of β_i. Using Proposition 5 if $\beta_i \stackrel{w}{\rightarrow} \beta_i^j$ then either $\beta_k \stackrel{w}{\rightarrow} \beta_k^j$ (and $X_1^i \ldots X_{j-1}^i$ is the same sequence as $X_1^k \ldots X_{j-1}^k$) or $(\beta_k \cdot w) = \emptyset$. Let G be used in an application of BAL in π. By property (4) G is $(F \cdot u)$ for some u and for all prefixes v of u $|(F \cdot v)| > m - (\mathrm{M}^2 + 3\mathrm{M})$. Therefore $G = E_1 \beta_1^{b_1} G_1 + \cdots + E_n \beta_n^{b_n} G_n$ where if $|\beta_i| \geqslant (\mathrm{M}^2 + 3\mathrm{M})$ then $b_i = \mathrm{M}^2 + 3\mathrm{M}$ and if $|\beta_i| < \mathrm{M}^2 + 3\mathrm{M}$ then $\beta_i^{b_i}$ is ε. The result of BAL using G, assume it is BAL(L), has the form $E_1'(G \cdot w(X_1)) + \cdots + E_k'(G \cdot w(X_k)) = (G \cdot w)$ where $|w| \leqslant \mathrm{M}$. However,

$$(G \cdot w(X_i)) = (E_1 \beta_1^{b1} \cdot w(X_i)) G_1 + \cdots + (E_n \beta_n^{bn} \cdot w(X_i)) G_n,$$

$$(G \cdot w) = (E_1 \beta_1^{b1} \cdot w) G_1 + \cdots + (E_n \beta_n^{bn} \cdot w) G_n$$

which establishes (5).

Property (5) shows that all configurations in all results of an application of BAL within π have common tails. A stronger property is that all goals in a block have this form. However this need not be true. It is possible that a configuration may become small before BAL is applied as follows, where H is small:

$$\begin{aligned} E_1G_1 + \cdots + E_nG_n &= F_1G_1 + \cdots + F_nG_n \\ &\vdots \\ H &= F_1^1G_1 + \cdots + F_n^1G_n \\ &\vdots \\ E_1'G_1 + \cdots + E_n'G_n &= F_1'G_1 + \cdots + F_n'G_n \end{aligned}$$

The top and bottom goals are the result of consecutive applications of BAL (which must be BAL(L) in the latter case). In between there are only applications of UNF.

If there are sufficient applications of BAL then (5) ensures that CUT must apply in a block. First, we show that BAL is repeatedly applied. Suppose not. Consider the result of the last application of BAL. As noted in the proof of (4), within $\mathrm{M}^2 + \mathrm{M}$ applications of UNF both BAL rules are permitted. Consider any goal in 1-head form, $X_1J_1+\cdots+X_kJ_k = Y_1K_1+\cdots+Y_lK_l$, where both BAL rules are permitted. Within at most M applications of UNF the left-hand configuration must have the form J_i and within M applications of UNF the right-hand configuration must have the form K_j, for otherwise BAL is applied. This argument is repeated. If the configurations do not contain recursive nonterminals π is finite with a last goal which is final. If the configurations do contain recursive nonterminals then they must repeatedly cycle around definitions of recursive nonterminals. In which case goals must eventually consist of small configurations only, and therefore final goals (repeats) must occur, by property (1).

Next, assume a block with repeated applications of BAL. Let $F = \beta_1G_1 + \cdots + \beta_nG_n$ be the root configuration (in S-head form) and let $E_1^1G_1+\cdots+E_n^1G_n = F_1^1G_1+\cdots+F_n^1G_n$ be the root goal of the block. The heads E_i^1, F_i^1 have bounded size, $(\mathrm{S} + 2\mathrm{M} + 1)$, which is independent of the sizes of the tails G_i. Let $(V_1^1,\ldots,V_n^1)$ be the canonical family of recursive nonterminals for this true root goal. Therefore by Fact 8 let m_1 be the least index such that $(V_1^1,\ldots,V_n^1)$ is an m_1-unifier for the heads $(E_1^1 + \cdots + E_n^1, F_1^1 + \cdots + F_n^1)$. Consider the result $E_1^2G_1 + \cdots + E_n^2G_n = F_1^2G_1 + \cdots + F_n^2G_n$ of the first application of BAL after m_1 applications of UNF from the root goal. There are two possibilities. First $E_1^2V_1^1 + \cdots + E_n^2V_n^1 \sim F_1^2V_1^1 + \cdots + F_n^2V_n^1$, in which case CUT applies with $k = 1$. Otherwise $E_1^2V_1^1 + \cdots + E_n^2V_n^1 \nsim F_1^2V_1^1 + \cdots + F_n^2V_n^1$. The recursive nonterminals are refined to $(V_1^2,\ldots,V_n^2)$ so that they are canonical for the two true goals $E_1^iG_1+\cdots+E_n^iG_n = F_1^iG_1+\cdots+F_n^iG_n$, for $i\in\{1,2\}$. Therefore there is a least m_2 such that $(V_1^2,\ldots,V_n^2)$ is an m_2-unifier for the heads $(E_1^i+\cdots+E_n^i, F_1^i+\cdots+F_n^i)$. Consider the result $E_1^3G_1 + \cdots + E_n^3G_n = F_1^3G_1 + \cdots + F_n^3G_n$ of the first application of BAL after m_2 applications of UNF from the goal $E_1^2G_1 + \cdots + E_n^2G_n = F_1^2G_1 + \cdots + F_n^2G_n$. There are the same two possibilities. However, there can be at most n refinements of the initial family of recursive nonterminals, which guarantees that CUT will apply with at most $k + 1$ premises where $k \leqslant n$.

Once a CUT applies the strategy is to build a new block as above. The argument is completed by showing that in any infinite path containing infinitely many applications of CUT there must be infinitely many final goals. Assume π is a path with infinitely many applications of CUT. There must be infinitely many root configurations $F^i = \beta_1 G_1^i + \cdots + \beta_n G_n^i$ with the same heads and therefore infinitely many root goals of an application of CUT with the same heads, $E_1^1 G_1^i + \cdots + E_n^1 G_n^i = F_1^1 G_1^i + \cdots + F_n^1 G_n^i$ because the heads are bounded by $\mathrm{S}+2\mathrm{M}+1$. $(V_1^1, \ldots, V_n^1)$ is canonical for all these true goals, and therefore there is a least m_1 such that it is also an m_1-unifier for the heads $(E_1^1 + \cdots + E_n^1, F_1^1 + \cdots + F_n^1)$. Consider the results $E_1^{2i} G_1^i + \cdots + E_n^{2i} G_n^i = F_1^{2i} G_1^i + \cdots + F_n^{2i} G_n^i$ of the first application of BAL after m_1 applications of UNF. Infinitely many of these goals must have the same heads because their size is bounded. Via the proof of property (4) any application of BAL within m_1 applications of UNF from a root goal uses a configuration $(\beta_1 \cdot u)G_1^i + \cdots + (\beta_n \cdot u)G_n^i$ where $|u| \leqslant m_1 + \mathrm{M}$. Consider goals which are the result of exactly m_1 applications of UNF from the roots F^i. If BAL does not apply within $\mathrm{M}^2 + \mathrm{M}$ further applications of UNF then, as seen earlier, both BAL rules are permitted, and hence as we saw above configurations on both sides of a goal decline in size. So for infinitely many i, either CUT applies with result $E_1^2 V_1^1 + \cdots + E_n^2 V_n^1 = F_1^2 V_1^1 + \cdots + F_n^2 V_n^1$ or $(V_1^1, \ldots, V_n^1)$ is refined. The argument is now repeated. As there can be at most n refinements, there must be infinitely many repeat goals.

Theorem 2. *If $E \sim F$ then there is a successful tableau for $E = F$.*

Proof. Assume that $E \sim F$. Now we keep applying the rules preserving truth using the strategy described above. If goals are small then one keeps applying BAL(L), BAL(R) and UNF. Otherwise one tries to build a block and apply CUT. By preserving truth it is not possible to reach an unsuccessful final goal. Also it is not possible to become stuck, as UNF is always applicable unless a goal is final. Hence the only issue is that the tableau construction goes on forever. Assume that there is an infinite path through the tableau. If CUT is only applied finitely often on this path then consider the subsequence after its final application. All attempts to build a block are thwarted, and therefore infinitely often there are small goals and so infinitely often there are final goals. Consequently, CUT must be applied infinitely often. However, by the analysis above there must then be infinitely many final goals. □

6. Conclusion

We have provided a proof of decidability of equivalence between DPDAs, which is essentially a simplification of [10]. However because the procedure consists of two semi-decision procedures we are unable to provide a complexity bound. More work is needed to see if we can find a useful bound on the depth m such that a canonical family of recursive nonterminals is an m-unifier (as in Fact 8). Example 6 illustrates the problem for seeking such a bound.

An intriguing open question is whether there is a more general class of context-free grammars than the strict deterministic for which language equivalence is decidable.

The proof technique developed in this paper can also be applied to decision problems for bisimulation equivalence. Language equivalence and bisimulation equivalence coincide in the deterministic case (provided there is no redundancy). A pushdown automaton is disjoint if it obeys the following condition:

if $pX \xrightarrow{\varepsilon} q\alpha$ and $pX \xrightarrow{a} r\lambda$ then $a = \varepsilon$.

DPDA are by definition disjoint. Sénizergues extends his result to decidability of bisimulation equivalence between configurations of disjoint nondeterministic pushdown automata which have deterministic stack popping ε-transitions [11]. The method here should also extend to this case. However there are still open problems. First, is bisimulation equivalence decidable for disjoint pushdown automata with nondeterministic stack popping ε-transitions (equivalent to "Type-1" processes in [12])? More generally, is bisimulation equivalence decidable for disjoint pushdown automata where ε-transitions may also be stack increasing (equivalent to "Type-2" processes in [12])?

Acknowledgements

I am deeply indebted to Olaf Burkart and Didier Caucal for numerous discussions about DPDA, and to Géraud Sénizergues for explanations of his result, and explaining the relationship between the proof here and his proof. I would also like to thank Petr Jančar for comments, and the referee for such detailed reports.

References

[1] S. Christensen, H. Hüttel, C. Stirling, Bisimulation equivalence is decidable for all context-free processes, Inform. Comput. 121 (1995) 143–148.
[2] S. Ginsberg, S. Greibach, Deterministic context-free languages, Inform. Control 9 (1966) 620–648.
[3] M. Harrison, Introduction to Formal Language Theory, Addison-Wesley, Reading, MA, 1978.
[4] M. Harrison, I. Havel, Strict deterministic grammars, J. Comput. System Sci. 7 (1973) 237–277.
[5] M. Harrison, I. Havel, A. Yehudai, On equivalence of grammars through transformation trees, Theoret. Comput. Sci. 9 (1979) 173–205.
[6] J. Hopcroft, J. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Reading, MA, 1979.
[7] H. Hüttel, C. Stirling, Actions speak louder than words: proving bisimilarity for context free processes, Proc. 6th Ann. Symp. on Logic in Computer Science, IEEE Computer Science Press, 1991, pp. 376–386.
[8] M. Oyamaguchi, N. Honda, Y. Inagaki, The equivalence problem for real-time strict deterministic languages, Inform. and Control 45 (1980) 90–115.
[9] G. Sénizergues, The Equivalence Problem for Deterministic Pushdown Automata is Decidable, Lecture Notes in Computer Science, Vol. 1256, Springer, Berlin, 1997, pp. 671–681.
[10] G. Sénizergues, L(A) = L(B)? decidability results from complete formal systems, Theoret. Comput. Sci. 251 (2001) 1–166.
[11] G. Sénizergues, Decidability of bisimulation equivalence for equational graphs of finite out-degree, Proc. IEEE 39th FOCS, 1998, pp. 120–129.
[12] C. Stirling, Decidability of bisimulation equivalence for normed pushdown processes, Theoret. Comput. Sci. 195 (1998) 113–131.