Linear Bidirectional Parsing for a Subclass of Linear Languages

Ştefan ANDREI *, Manfred KUDLEK†

Abstract

In this paper, our intention is to describe a useful subclass of linear grammars, called $LLin(m,n)$. We have denoted them in such a way because they are similar to the class of $LL(k)$ grammars ([1], [7]), and correspond to the linear grammars. Intuitively, "looking ahead" to the next m terminal symbols and "looking back" to the previous n terminal symbols is enough to determine uniquely the production which has to be applied. The membership problem for $LLin(m,n)$ grammars can be solved using a linear time complexity algorithm.

In the first section, we present some general properties of such grammars, such as unambiguity, a hierarchy of them, a comparison to $LL(k)$ grammars, recursiveness and closure properties. We have to notice that there exist non-deterministic languages which can be generated by this new class of grammars.

In the second section, we give a characterization theorem for such grammars (somehow similar to the $LL(k)$ grammars). Next, we describe a bidirectional parser for $LLin(m,n)$ grammars.

The third section treats $LLin(1,1)$ grammars. One of the main point is that the auxiliary function $first\_last$ can be computed using a polynomial time complexity algorithm. In this way, we can easily decide whether or not a linear grammar is $LLin(1,1)$.

Keywords: linear and $LL(k)$ grammars, bidirectional parsing

Mathematics Subject Classification: 68Q50, 68Q52, 68Q68.

*Faculty of Informatics, "Al.I.Cuza" University, Str. Berthelot, nr. 16, 6600, Iaşi, România. E-mail: stefan@infoiasi.ro. This work was supported by The World Bank/Joint Japan Graduate Scholarship Program.

†Fachbereich Informatik, Universität Hamburg, Vogt-Kölln-Straße 30, 22527, Hamburg, Germany. E-mail: kudlek@informatik.uni-hamburg.de

1 $LLin(m,n)$ Grammars. General Properties

In this paper, we define a new subclass of linear languages, for which the membership problem can be solved using an algorithm which has linear time complexity. For the general class of linear languages, it is known that w (an arbitrary word, $n = |w|$) can be parsed in time proportional to n^2 ([2]). We know that every sentential form of a linear grammar contains at most one nonterminal symbol. Using this property, our subclass of linear grammars is a generalization of $LL(k)$ grammars. The difference is that for the new subclass, the parsing is done from both sides of the word.

Before giving the definition of the new subclass of grammars, we give a list of definitions and notations which we will use in that paper.

Definitions:

- **context free grammar:** $G = (V_N, V_T, S, P)$, where:
  - V_N - the set of nonterminal symbols;
  - V_T - the set of terminal symbols;
  - $V = V_N \cup V_T$ - the set of symbols of G;
  - S - the start symbol;
  - $P \subseteq V_N \times V^*$ - the set of productions. A pair $(A, \beta) \in P$ is called $A-$production and it is denoted by $A \rightarrow \beta$. The productions $A \rightarrow \beta_1,\ A \rightarrow \beta_2, ...,\ A \rightarrow \beta_k$ will (sometimes) be denoted by $A \rightarrow \beta_1 \,|\, \beta_2 \,|\, ... \,|\, \beta_k$.
- **empty word:** λ (the word of length 0);
- **linear grammar:** is a context free grammar for which the set of productions satisfies $P \subseteq V_N \times (V_T^*(V_N V_T^* \cup \{\lambda\}))$;
- **derivation in G:** $\alpha \underset{G}{\Longrightarrow} \beta$ if $\exists\ A \in \alpha$ and $A \rightarrow r \in P$ such that $\alpha = \alpha_1\, A\, \alpha_2,\ \beta = \beta_1\, r\, \beta_2$; the transitive (reflexive) closure of the relation $\underset{G}{\Longrightarrow}$ is denoted by $\overset{+}{\underset{G}{\Longrightarrow}}$ ($\overset{*}{\underset{G}{\Longrightarrow}}$);
- X is an **accesible symbol in** G if there exists a derivation $S \overset{*}{\underset{G}{\Longrightarrow}} \alpha X \beta$, $\alpha, \beta \in V^*$;
- $A \in V_N$ is **productive** if there exists a derivation $A \overset{*}{\underset{G}{\Longrightarrow}} u$, with $u \in V_T^*$ (the other nonterminal symbols are called **useless**);
- G is a **reduced grammar** if all symbols are accesible and all nonterminal symbols are productive;
- **the set of sentential forms of the grammar** G**:** $S(G) = \{\alpha \in V^* \mid \exists\ S \overset{*}{\underset{G}{\Longrightarrow}} \alpha\}$.

- **the language of the grammar** G**:** $L(G) = \{w \in V_T^* \mid \exists\, S \overset{*}{\underset{G}{\Longrightarrow}} w\}$ (in fact, $L(G) = S(G) \cap V_T^*$).

Notations:

- nonterminal symbols: S (start symbol), A, B, ...
- terminal symbols: a, b, c, ...
- symbols (terminal or nonterminal): X, Y, ...
- terminal words: u, v, x, y, w, ...
- words (over terminal and nonterminal symbols): α, β, γ ...
- derivations: $\overset{r}{\underset{G}{\Longrightarrow}}$ means that the production r was applied in G; $\overset{\pi}{\underset{G}{\Longrightarrow}}$ refers to a sequence of productions (syntactic analysis);
- let $w = w_1\, w_2 \ldots w_k$ be a word over V. Then
  - $^{(m)}w = \begin{cases} w_{k-m+1}\, w_{k-m+2} \ldots w_k & \text{if } m \leq k \\ w & \text{otherwise} \end{cases}$
  - $w^{(n)} = \begin{cases} w_1\, w_2 \ldots w_n & \text{if } n \leq k \\ w & \text{otherwise} \end{cases}$
- $\mathbf{N}$ denotes the set of natural numbers, $\mathbf{N}^+$ denotes the set of strict positive natural numbers.

Definition 1.1 *Let* $G = (V_N, V_T, S, P)$ *be a linear grammar. We say that* G *is* $LLin(m,n)$, $m,\ n \in \mathbf{N}$, *if for any two derivations of the form*

$$S \overset{*}{\underset{G}{\Longrightarrow}} u\,A\,v \underset{G}{\Longrightarrow} u\,\beta_1\,v \overset{*}{\underset{G}{\Longrightarrow}} u\,x\,v$$

$$S \overset{*}{\underset{G}{\Longrightarrow}} u\,A\,v \underset{G}{\Longrightarrow} u\,\beta_2\,v \overset{*}{\underset{G}{\Longrightarrow}} u\,y\,v$$

with $u,\ v,\ x,\ y \in V_T^*$, *for which* $x^{(n)} = y^{(n)}$ *and* $^{(m)}x = {}^{(m)}y$, *then* $\beta_1 = \beta_2$.

Intuitively, this definition means for linear grammars that: *Given an arbitrary sentential form, "looking back" to the previous* n *terminal symbols and "looking ahead" to the next* m *terminal symbols, we can decide uniquely which production has to be applied* (Figure 1). According to this definition, overlapping of symbols is allowed.

Definition 1.2 *We say that the language* $L \subseteq V_T^*$ *is* $\mathcal{L}\mathcal{L}in(m,n)$ *if there exists a* $LLin(m,n)$ *grammar* G *for which* $L = L(G)$.

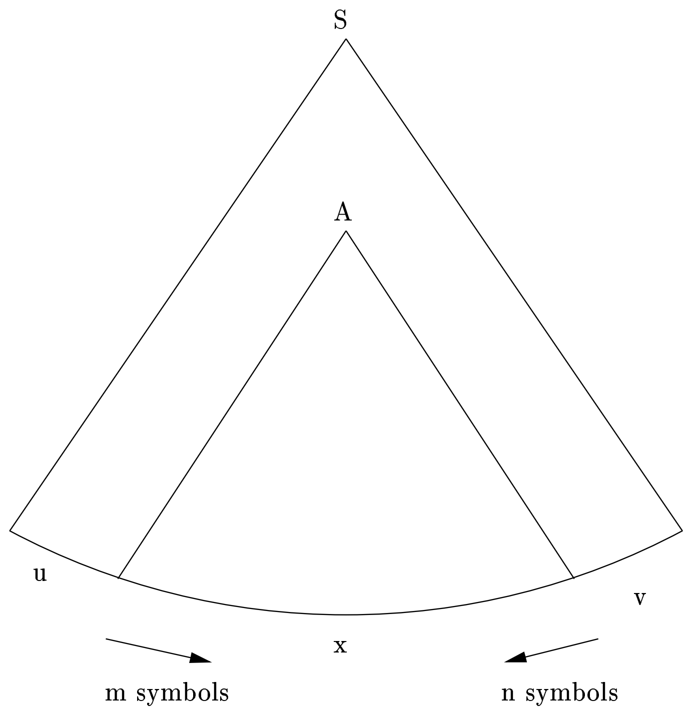

Figure 1

The next example contains some representative linear languages which can be expressed using $LLin(m,n)$ grammars.

Example 1.1

- $G_1 = (\{S\}, \{a,b,c\}, S, \{S \rightarrow a\,S\,a \mid b\,S\,b \mid c\})$ *is* $LLin(1,1)$ *and, of course,* $L(G_1) = \{w\,c\,\widetilde{w} \mid w \in \{a,b\}^*\}$;
- $G_2 = (\{S\}, \{a,b,c\}, S, \{S \rightarrow a\,S\,a \mid a\,S\,b \mid b\,S\,a \mid b\,S\,b \mid c\})$ *is* $LLin(1,1)$ *and* $L(G_2) = \{w_1\,c\,w_2 \mid w \in \{a,b\}^*,\ |w_1| = |w_2|\}$;
- $G_3 = (\{S\}, \{a,b,c\}, S, \{S \rightarrow a\,a\,S\,a\,a \mid a\,b\,S\,a\,b \mid a\,b\,S\,b\,a \mid b\,a\,S\,a\,b \mid b\,a\,S\,b\,a \mid b\,b\,S\,b\,b \mid c\})$ *is* $LLin(2,2)$ *and* $L(G_3) = \{w_1\,c\,w_2 \mid w \in \{a,b\}^*,\ |w_1| = |w_2| = even,\ N_{w_1}(a) = N_{w_2}(a)$ *and* $N_{w_1}(b) = N_{w_2}(b)\}$, *where* $N_{w_1}(a)$ *denotes the number of symbols 'a' from* w_1;
- $G_4 = (\{S,A\}, \{a,b\}, S, \{S \rightarrow a\,S \mid A\,b,\ A \rightarrow A\,b \mid \lambda\})$ *is* $LLin(1,1)$ *and* $L(G_4) = \{a^n\,b^m \mid n \geq 0,\ m \geq 1\}$;
- $G_5 = (\{S,A\}, \{a,b,c\}, S, \{S \rightarrow a\,S\,a \mid A,\ A \rightarrow b\,A\,b \mid c\})$ *is* $LLin(1,1)$ *and* $L(G_5) = \{a^n\,b^m\,c\,b^m\,a^n \mid n,\ m \geq 1\}$;

- $G_6 = (\{S, A, B\}, \{a, b\}, S, \{S \to A\,c \mid B,\ A \to a\,A\,b\,b \mid a\,b\,b,\ B \to a\,B\,b \mid a\,b\})$ *is* $LLin(2,1)$ *and* $L(G_6) = \{a^n\, b^{2n}\, c,\ a^n\, b^n \mid n \geq 1\}$;

Given a context-free grammar G, a derivation is called **left most** (denoted by $\underset{lm}{\Longrightarrow}$)) if in every sentential form (using a production from G) the first occurrence of a nonterminal symbol is replaced. A context-free grammar G is called **ambiguous** if there exists a word $w \in V_T^*$ for which there exist at least two distinct left most derivations $S \overset{*}{\underset{G}{\Longrightarrow}} w$.

A linear grammar may be ambiguous. For example, let us consider the linear grammar G given by the productions:

1. $S \to a\,S$
2. $S \to S\,a$
3. $S \to a$

For the word $w = a\,a\,a$, there exist two left (or right) most derivations:

$$S \underset{G}{\Longrightarrow} a\,S \underset{G}{\Longrightarrow} a\,S\,a \underset{G}{\Longrightarrow} a\,a\,a$$

$$S \underset{G}{\Longrightarrow} S\,a \underset{G}{\Longrightarrow} a\,S\,a \underset{G}{\Longrightarrow} a\,a\,a$$

So, we conclude that G is an ambiguous linear grammar.

Next, we shall show that the subclass $LLin(m, n)$ contains only unambiguous grammars.

Theorem 1.1 *Every* $LLin(m, n)$ *grammar is unambiguous.*

Proof Consider $G = (V_N, V_T, S, P)$ being $LLin(m, n)$ grammar and suppose that it is ambiguous. This means, that there exists a word $w \in L(G)$ such that we can construct two distinct derivations (in G):

$$S = \alpha_0 \underset{lm}{\Longrightarrow} \alpha_1 \underset{lm}{\Longrightarrow} \alpha_2 \underset{G}{\Longrightarrow} \ldots \underset{lm}{\Longrightarrow} \alpha_k = w$$

$$S = \beta_0 \underset{lm}{\Longrightarrow} \beta_1 \underset{lm}{\Longrightarrow} \beta_2 \underset{G}{\Longrightarrow} \ldots \underset{lm}{\Longrightarrow} \beta_{k'} = w$$

We shall show by induction on i that $\alpha_i = \beta_i,\ \forall\ i \geq 0$. The basis of induction $(i = 0)$ is clear.

Let us suppose that $\alpha_j = \beta_j,\ \forall\, 0 \leq j \leq i$. We have to prove that $\alpha_{i+1} = \beta_{i+1}$. Because G is a linear grammar (all derivations are also left and right most), we can rewrite the previous derivation such as:

$$S \overset{*}{\underset{G}{\Longrightarrow}} \alpha_i = u\,A\,v \underset{G}{\Longrightarrow} u\,\gamma_1\,v \overset{*}{\underset{G}{\Longrightarrow}} u\,x\,v = w$$

$$S \overset{*}{\underset{G}{\Longrightarrow}} \beta_i = u\,A\,v \underset{G}{\Longrightarrow} u\,\gamma_2\,v \overset{*}{\underset{G}{\Longrightarrow}} u\,y\,v = w$$

From $u\,x\,v = w$ and $u\,y\,v = w$, it follows that $x = y$. Therefore, ${}^{(m)}x ={}^{(m)} y$ and $x^{(n)} = y^{(n)}$. Thus, $\gamma_1 = \gamma_2$, so $\alpha_{i+1} = u\,\gamma_1\,v = u\,\gamma_2\,v = \beta_{i+1}$. Hence, $\alpha_i = \beta_i,\ \forall\ 0 \le i \le \min(k, k')$. But $\alpha_k = \beta_{k'} = w$, so $k = k'$. Therefore the assumption that G is ambiguous is false. ■

We shall denote the set of $LL(k)$ linear grammars by $LLin(k)$. We may easily observe that the class of $LLin(1,1)$ grammars is bigger than $LLin(1)$ grammars. For instance, G_2 (from Example 1.1) is $LLin(1,1)$, but not $LLin(1)$.

Lemma 1.1 *Every $LLin(k)$ grammar is a $LLin(k,k')$ grammar,* $\forall\ k' \ge 0$.

Proof Directly from the definitions. ■

Lemma 1.2 *If G is a $LLin(m,n)$ grammar, then G is a $LLin(m',n')$ grammar, where $m' \ge m,\ n' \ge n$.*

Proof Directly from the definitions. ■

Theorem 1.2 *For all $m, n \ge 0$, the class $LLin(m,n)$ is properly included in the class $LLin(m',n')$, $\forall\ m' \ge m,\ \forall n' \ge n$.*

Proof The fact that $LLin(m,n)$ is included in $LLin(m',n')$ is obvious, where $m' \ge m,\ n' \ge n$ (Lemma 1.2). It remains to show that the inclusion is proper.

Let us consider the following linear grammar:

$$G:\ S \to a^m\,b^n \,|\, a^{m'}\,b^{n'}\ \ (m' \ge m,\ n' \ge n)$$

It is obvious that G is $LLin(m',n')$, but not $LLin(m,n)$ (of course, we have $(m'-m)^2 + (n'-n)^2 \ne 0$). ■

Theorem 1.3 *There exist linear languages which are not $\mathcal{LL}in(m,n)$, for any $m,\ n \in \mathbf{N}$.*

Proof Let us consider the linear language $L = L_1 \cup L_2$, where

$$L_1 = \{a^k\,c\,b^k \mid k \ge 1\} \text{ and } L_2 = \{a^k\,d\,b^{2k} \mid k \ge 1\}.$$

For instance, L can be generated by the linear grammar G_3:

- $S \to A \,|\, B$
- $A \to a\,A\,b \,|\, c$
- $B \to a\,B\,b\,b \,|\, d$

Let us suppose, by contrary, that there exist $m, n \in \mathbf{N}$ and $G \in LLin(m,n)$ such as $L(G) = L$. Let us denote $i = \max(m,n)$ and the words $w_1 = a^i\,c\,b^i$, $w_2 = a^i\,d\,b^{2i}$ which belong to L_1, respectively L_2. Because $L = L(G)$, then there exist the derivations:

$$S \overset{*}{\underset{G}{\Longrightarrow}} a^i\,c\,b^i$$

$$S \overset{*}{\underset{G}{\Longrightarrow}} a^i\, d\, b^{2i}$$

It is obvious that ${}^{(m)}w_1 ={}^{(m)} w_2$ and $w_1^{(n)} = w_2^{(n)}$, so this means that the first production applied in the above derivations is the same. Let $a^k\, A\, b^j$ be the last sentential form for which:

$$S \overset{*}{\underset{G}{\Longrightarrow}} a^k\, A\, b^j \underset{G}{\Longrightarrow} a^k\, \beta_1\, b^j \overset{*}{\underset{G}{\Longrightarrow}} a^k\, a^{i-k}\, c\, b^{i-j}\, b^j = w_1$$

$$S \overset{*}{\underset{G}{\Longrightarrow}} a^k\, A\, b^j \underset{G}{\Longrightarrow} a^k\, \beta_2\, b^j \overset{*}{\underset{G}{\Longrightarrow}} a^k\, a^{i-k}\, d\, b^{2i-j}\, b^j = w_2$$

and ${}^{(m)}a^{i-k}\, c\, b^{i-j} ={}^{(m)} a^{i-k}\, d\, b^{2i-j}$, $a^{i-k}\, c\, b^{i-j\,(n)} = a^{i-k}\, d\, b^{2i-j\,(n)}$. Because G is $LLin(m,n)$ it follows that $\beta_1 = \beta_2$. But during the following derivation $a^k\, A\, b^j \overset{*}{\underset{G}{\Longrightarrow}} a^k\, a^{i-k}\, c\, b^{i-j}\, b^j$ only productions corresponding to L_1 will be applied (which are distinct from productions corresponding to L_2, $L_1 \cap L_2 = \emptyset$). So, we obtain a contradiction because $A \to \beta_1 = A \to \beta_2$. Therefore G is not a $LLin(m,n)$ grammar. ■

If $\alpha = \alpha_1\, \alpha_2 \ldots \alpha_k$ is a word over V, then $\widetilde{\alpha} = \alpha_k \ldots \alpha_2\, \alpha_1$ is called **the reverse (mirror) of** α. If $G = (V_N, V_T, S, P)$ is a context-free grammar, we shall denote by $\widetilde{G} = (V_N, V_T, S, \widetilde{P})$ its **reverse (mirror) grammar**, where $\widetilde{P} = \{A \to \widetilde{\beta} \mid A \to \beta \in P\}$.

Corrolary 1.1 *The following facts hold:*

- *G is $LLin(m,n)$ grammar iff $\widetilde{G}$ is $LLin(n,m)$ grammar, i.e. the class of $LLin(m,n)$ grammars is closed under mirror image, $\forall\ m \geq 0,\ n \geq 0$;*
- *G is $LLin(m,0)$ grammar iff G is $LLin(m)$ grammar;*
- *G is $LLin(0,n)$ grammar iff $\widetilde{G}$ is $LLin(n)$ grammar.*

Proof Directly from the definitions and the fact that $\widetilde{G}$ is also linear grammar, where G is a linear grammar. ■

Definition 1.3 *Let $G = (V_N, V_T, S, P)$ be a context-free grammar. We say that:*

- *$A \in V_N$ is* **left-recursive**, *if there exists a derivation $A \overset{+}{\underset{G}{\Longrightarrow}} A\, \alpha,\ \alpha \in V^+$;*
- *$A \in V_N$ is* **right-recursive**, *if there exists a derivation $A \overset{+}{\underset{G}{\Longrightarrow}} \beta\, A$, where $\beta \in V^+$;*
- *G is* **left (right) recursive grammar** *if there exists $A \in V_N$ a left (right) recursive symbol.*

It is known that a left-recursive grammar cannot be $LL(k)$ ([6], [5]), for any $k \geq 0$. However, there exist some procedures to transform left-recursion into right-recursion. In comparison with $LL(k)$ grammars, the $LLin(m,n)$ grammars can be left-recursive, even right-recursive (like G_4 from Example 1.1), but not for the same nonterminal symbol.

Theorem 1.4 *If the reduced linear grammar G contains a left and right recursive nonterminal symbol A, then G cannot be $LLin(m,n)$, $\forall\ m,n \in \mathbf{N}$.*

Proof Let A be a left and right recursive symbol. Because G is linear grammar, this means that there exist the derivations:

$$A \overset{+}{\underset{G}{\Longrightarrow}} A\,v',\ A \overset{+}{\underset{G}{\Longrightarrow}} u'\,A,\ u',\ v' \in V_T^+.$$

Without loss of generality, we suppose that the first distinct productions applied in the above derivations are:

$$A \to B\,v_1 \text{ and } A \to u_1\,C$$

Now, because G is a reduced linear grammar, it follows that there exists a derivation:

$$S \overset{*}{\underset{G}{\Longrightarrow}} u\,A\,v$$

Now, we suppose that there exist $m,n \in \mathbf{N}$ such as G is $LLin(m,n)$. Continuing the above derivation, we may write:

$$S \overset{*}{\underset{G}{\Longrightarrow}} u\,A\,v \underset{G}{\Longrightarrow} u\,x\,v_1\,v \overset{*}{\underset{G}{\Longrightarrow}} u\,A\,v'\,v \overset{+}{\underset{G}{\Longrightarrow}} u\,u'\,A\,v'\,v \overset{+}{\underset{G}{\Longrightarrow}} \ldots \overset{+}{\underset{G}{\Longrightarrow}} u\,(u')^m A(v')^n\,v$$

$$S \overset{*}{\underset{G}{\Longrightarrow}} u\,A\,v \underset{G}{\Longrightarrow} u\,u_1\,y\,v \overset{*}{\underset{G}{\Longrightarrow}} u\,u'\,A\,v \overset{+}{\underset{G}{\Longrightarrow}} u\,u'\,A\,v'\,v \overset{+}{\underset{G}{\Longrightarrow}} .. \overset{+}{\underset{G}{\Longrightarrow}} u(u')^{m+1}A(v')^{n+1}v$$

But ${}^{(m)}\left((u')^m\,A\,(v')^n\right) ={}^{(m)} \left((u')^{m+1}\,A\,(v')^{n+1}\right)$ and $\left((u')^m\,A\,(v')^n\right)^{(n)} =$ $= \left((u')^{m+1}\,A\,(v')^{n+1}\right)^{(n)}$. Using the fact that G is $LLin(m,n)$, it follows that $A \to B\,v_1$ coincides with $A \to u_1\,C$ (a contradiction !).

Therefore G cannot be $LLin(m,n)$, $\forall\ m,n \in \mathbf{N}$. ∎

The class of $LLin(m,n)$ grammars can generate some classical non-deterministic languages ([4]), such as $L = \{a^n\,b^{2n}\,c,\ a^n\,b^n \mid n \geq 1\}$. For instance, G_6 from Example 1.1 can generate this language.

Proposition 1.1 *(The Pumping Lemma for Linear Languages, [8])*

For every linear language $L \subseteq V_T^*$, there exists a natural number N, depending only on L, such that if $z \in L$ with $|z| > N$ then there exist $u,v,w,x,y \in V_T^*$ for which the following conditions are fulfilled:

(a) $z = u\,v\,w\,x\,y$;

(b) $|v\,x| > 0$;

(c) $|u\,v\,x\,y| \leq N$;

(d) $\forall\ i \geq 0:\ u\,v^i\,w\,x^i\,y \in L$.

Theorem 1.5 *(closure properties)* $\mathcal{LL}in(m, n)$ *is not closed under:*

(i) *union*

(ii) *intersection*

(iii) *catenation*

(iv) *homomorphism*

Proof

(i) Let $G_1 = (\{S\}, \{a, b, c\}, S, \{S \rightarrow a\,S\,b \mid c\})$ and $G_2 = (\{S\}, \{a, b, d\}, S, \{S \rightarrow a\,S\,b\,b \mid d\})$ be two $LLin(1, 0)$ (or $LLin(0, 1)$) grammars. Obviously, we have $L(G_1) = \{a^k\,c\,b^k \mid k \geq 1\}$ and $L(G_2) = \{a^k\,d\,b^{2k} \mid k \geq 1\}$. The language $L(G_1) \cup L(G_2)$ is not a linear language (proof of Theorem 1.3);

(ii) Consider $G_1 = (\{S, A\}, \{a, b, c\}, S, \{S \rightarrow S\,c \mid A,\ A \rightarrow a\,A\,b \mid a\,b\})$ and $G_2 = (\{S, A\}, \{a, b, c\}, S, \{S \rightarrow a\,S \mid A,\ A \rightarrow b\,A\,c \mid b\,c\})$ be two $LLin(0, 2)$ and $LLin(2, 0)$ grammars, respectively. So $L(G_1) = \{a^n\,b^n\,c^m \mid m, n \geq 1\}$ and $L(G_2) = \{a^m\,b^n\,c^n \mid m, n \geq 1\}$. Then the intersection of these languages $L(G_1) \cap L(G_2) = \{a^n\,b^n\,c^n \mid n \geq 1\}$ is not a context free language (nor linear, of course) ([2], [3], [6]);

(iii) Let $G = (\{S\}, \{a, b\}, S, \{S \rightarrow a\,S\,b \mid a\,b\})$ be a $LLin(2, 0)$ (or $LLin(0, 2)$) grammar. We obtain $L(G) = \{a^n\,b^n \mid n \geq 1\}$. We shall prove that the language $L(G) \cdot L(G) = \{a^n\,b^n\,a^m\,b^m \mid m,\, n \geq 1\}$ is not linear. Denoting $L = L(G) \cdot L(G)$, we suppose, by contrary, that L is a linear language. Applying the pumping lemma for linear languages, we can choose the word $z = a^N\,b^N\,a^N\,b^N$ which belongs to L. Then $u\,v \in \{a\}^*$ and $x\,y \in \{b\}^*$ because of (c) condition. This implies that there exist $i_1,\ i_2,\ i_3,\ i_4 \in \mathbf{N}$, $i_2 + i_3 \geq 1$ (because of the (b) condition) such that:

$$u = a^{i_1},\ v = a^{i_2},\ w = a^{N-i_1-i_2}\,b^N\,a^N\,b^{N-i_3-i_4},\ x = b^{i_3},\ y = b^{i_4}.$$

Using the condition (d) and choosing, for instance, $i = 0$, we obtain that $u\,w\,y \in L$, i.e. $a^{N-i_2}\,b^N\,a^N\,b^{N-i_3} \in L$. Since $i_2 + i_3 \geq 1$, we get neither $N - i_2 \neq N$, nor $N - i_3 \neq N$. Therefore $a^{N-i_2}\,b^N\,a^N\,b^{N-i_3}$ cannot belong to L.

(iv) Let $G = (\{S, A, B\}, \{a, b, c, d, e, f\}, S, \{S \rightarrow A \mid B,\ A \rightarrow a\,A\,b \mid c,\ B \rightarrow e\,B\,f\,f \mid d\})$ be a $LLin(1, 0)$ (or $LLin(0, 1)$) grammar. Obvious, the language generated by it is $L(G) = \{a^k\,c\,b^k,\ d^k\,e\,f^{2k} \mid k \geq 1\}$. Now, we consider the homomorphism defined by $h(a) = a,\ h(b) = b,\ h(c) = c,\ h(d) = a,\ h(e) = d,\ h(f) = b$. This implies that $h(L(G)) = L$, where L is the language used in the proof of Theorem 1.3. Because L does not belongs to the class of $\mathcal{LL}in(m, n)$ languages, our class of languages is not closed under homomorphism.

■

2 A Bidirectional Parser for $LLin(m,n)$ Grammars

In this section, we shall define some useful sets of pairs of words. We shall present a characterisation theorem for $LLin(m,n)$ grammars and a bidirectional parser for them.

Definition 2.1 *Let $G = (V_N, V_T, S, P)$ be a linear grammar, $\alpha \in V^*$, $\#$ a new terminal symbol and $m, n \in \mathbf{N}^+$. We define $first_m\_last_n(\alpha)$ as the union of the following sets of pairs of words corresponding to α, m, n such as:*

- (u,v) *if* $\exists\, \alpha \underset{G}{\overset{*}{\Longrightarrow}} u\,x\,v,\ u,x,v \in V_T^*,\ |u| = m,\ |v| = n,\ |u\,x\,v| \geq \max\{m,n\}$;
- $(\#\,x\,v, v)$ *if* $\exists\, \alpha \underset{G}{\overset{*}{\Longrightarrow}} x\,v,\ x,v \in V_T^*,\ |x\,v| = k < m,\ k \geq n,\ |v| = n$;
- $(u, u\,x\,\#)$ *if* $\exists\, \alpha \underset{G}{\overset{*}{\Longrightarrow}} u\,x,\ u,x \in V_T^*,\ |u\,x| = k < n,\ k \geq m,\ |u| = m$;
- $(\#\,x, x\,\#)$ *if* $\exists\, \alpha \underset{G}{\overset{*}{\Longrightarrow}} x,\ x \in V_T^*,\ |x| = k,\ k < m,\ k < n$.

Theorem 2.1 *(characterization of $LLin(m,n)$ grammars)*
Let $G = (V_N, V_T, S, P)$ be a reduced linear grammar. Then G is $LLin(m,n)$ grammar iff the following condition holds:

(1) $first_m\_last_n(\beta_1) \cap first_m\_last_n(\beta_2) = \emptyset,\ \forall A \to \beta_1,\ A \to \beta_2 \in P,\ \beta_1 \neq \beta_2.$

Proof
($\Longrightarrow$) Let us suppose that G does not satisfy the condition (1). This means that there exist two distinct productions $A \to \beta_1$, $A \to \beta_2$ such that the following relation holds:

$$first_m\_last_n(\beta_1) \cap first_m\_last_n(\beta_2) \neq \emptyset.$$

According to the Definition 2.1, there exist four situations (remind that $\#$ is a new terminal symbol):

1) $(u', v') \in first_m\_last_n(\beta_1) \cap first_m\_last_n(\beta_2)$. Then there exist the derivations ($|u'| = m,\ |v'| = n$):

$$\beta_1 \underset{G}{\overset{*}{\Longrightarrow}} u'\,x\,v',\ x \in V_T^*,$$

$$\beta_2 \underset{G}{\overset{*}{\Longrightarrow}} u'\,y\,v',\ y \in V_T^*.$$

Because G is a reduced grammar, it follows that A is an accesible nonterminal, so we obtain the derivations:

$$S \underset{G}{\overset{*}{\Longrightarrow}} u\,A\,v \underset{G}{\Longrightarrow} u\,\beta_1\,v \underset{G}{\overset{*}{\Longrightarrow}} u\,u'\,x\,v'\,v$$

$$S \overset{*}{\underset{G}{\Longrightarrow}} u\,A\,v \underset{G}{\Longrightarrow} u\,\beta_2\,v \overset{*}{\underset{G}{\Longrightarrow}} u\,u'\,y\,v'\,v$$

According to the Definition 1.1, it follows that $\beta_1 = \beta_2$. Contradiction !

2) $(\#\,x\,v', v') \in first_m\_last_n(\beta_1) \cap first_m\_last_n(\beta_2)$. Then according to Definition 2.1, there exist the derivations ($|x\,v'| = k < m,\ |v'| = n \leq k$):

$$\beta_1 \overset{*}{\underset{G}{\Longrightarrow}} x\,v',\ x \in V_T^*,$$

$$\beta_2 \overset{*}{\underset{G}{\Longrightarrow}} x\,v'.$$

So, again we obtain the derivations:

$$S \overset{*}{\underset{G}{\Longrightarrow}} u\,A\,v \underset{G}{\Longrightarrow} u\,\beta_1\,v \overset{*}{\underset{G}{\Longrightarrow}} u\,x\,v'\,v$$

$$S \overset{*}{\underset{G}{\Longrightarrow}} u\,A\,v \underset{G}{\Longrightarrow} u\,\beta_2\,v \overset{*}{\underset{G}{\Longrightarrow}} u\,x\,v'\,v$$

According to the Definition 1.1, it follows that $\beta_1 = \beta_2$. Contradiction !

The rest of cases can be solved in an similar way.

($\Longleftarrow$) Let us suppose that G is not a $LLin(m,n)$ grammar. Then there exist two distinct derivations:

$$S \overset{*}{\underset{G}{\Longrightarrow}} u\,A\,v \underset{G}{\Longrightarrow} u\,\beta_1\,v \overset{*}{\underset{G}{\Longrightarrow}} u\,x\,v$$

$$S \overset{*}{\underset{G}{\Longrightarrow}} u\,A\,v \underset{G}{\Longrightarrow} u\,\beta_2\,v \overset{*}{\underset{G}{\Longrightarrow}} u\,y\,v$$

such that $x^{(n)} = y^{(n)}$ and ${}^{(m)}x = {}^{(m)}y$. Then there exist $u',\ v' \in V_T^*$ such as $|u'| = m,\ |v'| = n$ and $x = u'\,z_1\,v',\ y = u'\,z_2\,v'$. This implies that the pair $(u', v') \in first_m\_last_n(\beta_1) \cap first_m\_last_n(\beta_2)$. But $A \to \beta_1$ and $A \to \beta_2$ are distinct productions (i.e. $\beta_1 \neq \beta_2$) in G, so we obtain a contradiction to the fact that G satisfies the condition (1). ■

Theorem 2.1 prove that the following problem is decidable:
"*Given a linear grammar* $G = (V_N, V_T, S, P)$ *and two integers* m *and* n, *one can decide if the grammar is* $LLin(m,n)$."

Next, we shall define a device similar somehow with a deterministic pushdown "transducer". This will be called the **bidirectional parser** (syntactic analyser) attached to the $LLin(m,n)$ grammar G. It scans an "input string", one or/and two strings at a time, from left to right or right to left. It can push or pop strings in the double ended queue (deque) from both sides. In the output tape, it provides the syntactic analysis. It returns with the value "ACC" or "ERR" depending on whether the input string is accepted or not.

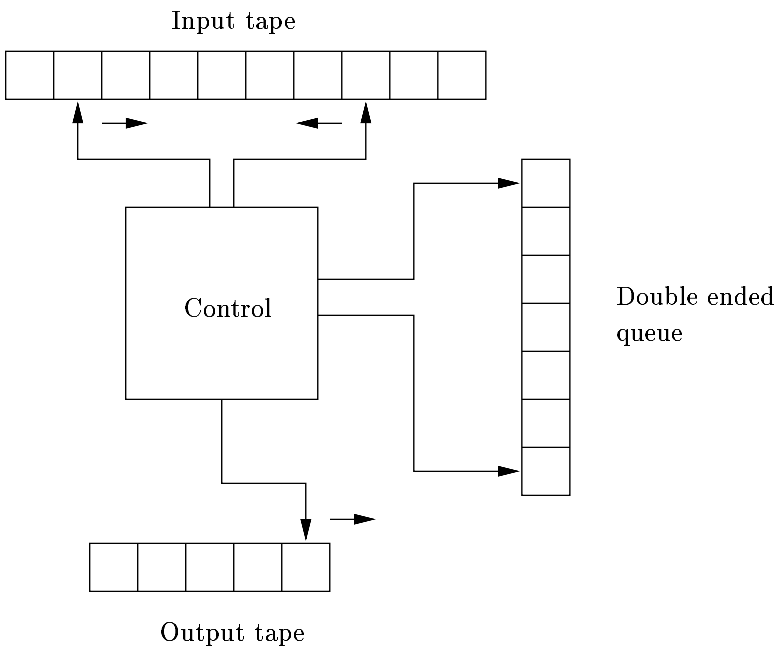

Figure 2. LLin(m,n) style bidirectional parser

Formally, we give the following definition:

Definition 2.2 *Let* $G = (V_N, V_T, S, P)$ *be a* $LLin(m,n)$ *grammar. We denote by* $\mathcal{C} \subseteq \#V_T^*\# \times V^* \times \{1,2,...,|P|\}^*$ *the* **set of possible configurations**, *where* $\#$ *is a special character (a new terminal symbol). The* **bidirectional parser** *(denoted by* $BP_{m,n}(G)$*) is the pair* $(\mathcal{C}_0, \vdash)$*, where the set* $\mathcal{C}_0 = \{(w, S, \lambda) \mid w \in V_T^*\} \subseteq \mathcal{C}$ *is called the set of* **initial configurations**, *and* $\vdash \subseteq \mathcal{C} \times \mathcal{C}$ *is the* **transition binary relation** *(sometimes denoted* $\underset{BP_{m,n}(G)}{\vdash}$*) between configurations given by:*

1^0. **Expand transition:**

$(\#u\#, A, \lambda) \vdash (\#u\#, \beta, \pi r)$ *if* $r = A \to \beta$ *for which the pair* $({}^{(m)}u\#, \#u^{(n)}) \in first_m\_last_n(\beta)$

2^0. **Reduce transitions:**

a) $(\#v_1\, u\#, v_1\, A, \pi) \vdash (\#u\#, A, \pi),\ v_1 \in V_T^+$

b) $(\#u\, v_2\#, A\, v_2, \pi) \vdash (\#u\#, A, \pi),\ v_2 \in V_T^+$

c) $(\#v_1\, u\, v_2\#, v_1\, A\, v_2, \pi) \vdash (\#u\#, A, \pi),\ v_1, v_2 \in V_T^+$

3^0. **Acceptance transition:**

$(\#\#, \lambda, \pi) \vdash ACC$

4^0. **Rejection transition:**

$$(\#u\#, \alpha, \pi) \vdash ERR \text{ if no transitions of type } 1^0,\ 2^0,\ 3^0 \text{ can be applied.}$$

We denote by $\overset{+}{\vdash}$ ($\overset{*}{\vdash}$) the transitive (reflexive) closure of the above binary relation $\vdash$. Sometimes, for a given grammar G, we may denote these closures by $\underset{BP_{m,n}(G)}{\overset{+}{\vdash}}$ ($\underset{BP_{m,n}(G)}{\overset{*}{\vdash}}$ respectively).

It is obvious that the bidirectional parser $BP_{m,n}(G)$ is deterministic, i.e. for an arbitrary configuration, at most one configuration may be reached. The only place at which this not so obvious, is at the expand transition 1^0. But the condition $({}^{(m)}(u\#), (\#u)^{(n)}) \in first_m\_last_n(\beta)$ ensures the uniqueness of the production $A \to \beta$ because G is a $LLin(m,n)$ grammar.

Lemma 2.1 *Let G be a $LLin(m,n)$ grammar. Then, the following implications are fulfilled:*

(i) $(\#v_1\, u\, v_2\#, S, \lambda) \underset{BP_{m,n}(G)}{\overset{*}{\vdash}} (\#u\#, X, \pi')$, *implies* $S \overset{\pi'}{\underset{G}{\Longrightarrow}} v_1\, X\, v_2$;

(ii) $(\#w\#, S, \lambda) \underset{BP_{m,n}(G)}{\overset{*}{\vdash}} (\#\#, \lambda, \pi)$, *implies* $S \overset{\pi}{\underset{G}{\Longrightarrow}} w$.

Proof

(i) We proceed by induction on the length of π'.

Basis: $|\pi'| = 0$. Thus $v_1 = v_2 = \lambda,\ A = S$, thus obviously $S \overset{\lambda}{\underset{G}{\Longrightarrow}} S$.

Inductive Step: $|\pi'| > 0$. Let us consider $\pi' = \pi_1'\ r$, where $r = B \to \beta$ is the last applied production. Denoting $v_1 = v_{11}\, v_{12}$ and $v_2 = v_{21}\, v_{22}$ we obtain:

$$(\#v_1\, u\, v_2\#, S, \lambda) = (\#v_{11}\, v_{12}\, u\, v_{21}\, v_{22}\#, S, \lambda) \overset{*}{\vdash} (\#v_{12}\, u\, v_{21}\#, B, \pi_1').$$

From the inductive hypothesis, it follows that $S \overset{\pi_1'}{\underset{G}{\Longrightarrow}} v_{11}\, B\, v_{22}$. Then applying 1^0, from Definition 2.2, we obtain the configuration $(\#v_{12}\, u\, v_{21}\#, \beta, \pi_1\, r)$, where $({}^{(m)}v_{12}\, u\, v_{21}, v_{12}\, u\, v_{21}^{(n)}) \in first_m\_last_n(\beta)$. The next transitions

$$(\#v_{12}\, u\, v_{21}\#, \beta, \pi_1'\, r) \underset{BP_{m,n}(G)}{\overset{*}{\vdash}} (\#u\#, X, \pi')$$

imply that $BP_{m,n}(G)$ made only reduce transitions. So, $\beta = v_{12}\, X\, v_{21}$ (from 2^0 a),b),c), Definition 2.2). Now, we may write the derivation:

$$S \overset{\pi_1'}{\underset{G}{\Longrightarrow}} v_{11}\, B\, v_{22} \overset{r}{\underset{G}{\Longrightarrow}} v_{11}\, \beta\, v_{22} = v_{11}\, v_{12}\, X\, v_{21}\, v_{22} = v_1\, X\, v_2$$

(ii) We take in (i) $u = \lambda,\ X = \lambda,\ v_1\, v_2 = w,\ \pi' = \pi$. ■

Lemma 2.2 *Let G be a $LLin(m,n)$ grammar. Then, the following implications are fulfilled:*

(i) $S \overset{\pi'}{\underset{G}{\Longrightarrow}} v_1\,X\,v_2$ *implies* $(\#v_1\,u\,v_2\#, S, \lambda) \underset{BP_{m,n}(G)}{\overset{*}{\vdash}} (\#u\#, X, \pi')$*;*

(ii) $S \overset{\pi}{\underset{G}{\Longrightarrow}} w$ *implies* $(\#w\#, S, \lambda) \underset{BP_{m,n}(G)}{\overset{*}{\vdash}} (\#\#, \lambda, \pi)$*;*

Proof

(i) We proceed by induction on length of π'.

Basis: $|\pi'| = 0$. Thus $v_1 = v_2 = \lambda,\ A = S$, so the following transitions hold:

$$(\#u\#, S, \lambda) \underset{BP_{m,n}(G)}{\overset{*}{\vdash}} (\#u\#, S, \lambda).$$

Inductive Step: $|\pi'| > 0$. Let us consider $\pi' = \pi_1'\,r$, where $r = B \rightarrow \beta$ is the last applied production which form the sentential form $v_1\,X\,v_2$. So, the derivation may be written as:

$$S \overset{\pi_1'}{\underset{G}{\Longrightarrow}} v_{11}\,B\,v_{22} \overset{r}{\underset{G}{\Longrightarrow}} v_{11}\,v_{12}\,X\,v_{21}\,v_{22} = v_1\,X\,v_2$$

For π_1' we apply the inductive hypothesis, so we obtain

$$(\#v_{11}\,v_{12}\,u\,v_{21}\,v_{22}\#, S, \lambda) \underset{BP_{m,n}(G)}{\overset{*}{\vdash}} (\#v_{12}\,u\,v_{21}\#, B, \pi_1').$$

Now, we may continue with expand transition, and obtain the configuration $(\#v_{12}\,u\,v_{21}\#, v_{12}\,X\,v_{21}, \pi_1'\,r)$. Right now, we apply the reduce transitions a),b),c) and obtain the configuration $(\#u\#, X, \pi_1'\,r) = (\#u\#, X, \pi')$.

(ii) We take in (i) $u = \lambda,\ X = \lambda,\ v_1\,v_2 = w,\ \pi' = \pi$. ∎

Theorem 2.2 *(correctness and complexity of $BP_{m,n}(G)$)*
Let G be a $LLin(m,n)$ grammar. Then

$$(\#w\#, S, \lambda) \underset{BP_{m,n}(G)}{\overset{*}{\vdash}} (\#\#, \lambda, \pi) \underset{BP_{m,n}(G)}{\vdash} ACC \text{ iff } S \overset{\pi}{\underset{G}{\Longrightarrow}} w.$$

Obviously, $(\#w\#, S, \lambda) \underset{BP_{m,n}(G)}{\overset{*}{\vdash}} ERR$ *iff* $w \notin L(G)$. *The number of transitions of* $BP_{m,n}(G)$ *has* $\mathcal{O}(|w|)$ *time complexity, where* w *is the input word.*

Proof Both equivalences result directly from Lemmas 2.1 (ii) and 2.2 (ii), respectively. The time complexity results from the fact that $BP_{m,n}(G)$ is defined over a finite structure (grammar G) and $BP_{m,n}(G)$ (and syntactic analysis) is deterministic (no backtrack step is needed). ∎

In the next section, we shall refer to another practical bidirectional parser, i.e. for $LLin(1,1)$ grammars (the sets $first_m\_last_n$ can be computed in polynomial time related to the dimension of the input grammar).

3 Bidirectional Parsing for $LLin(1,1)$ Grammars

The $LLin(0,0)$ grammars have the property that there exists no nonterminal symbols which may be the left side of a production. Obvious, for a reduced $LLin(0,0)$ grammar, its language is finite, so there is no practical interest.

Also, we don't consider $LLin(1,0)$ or $LLin(0,1)$ grammars because they coincide with $LLin(1)$ grammars or reverse (mirror) $LLin(1)$ grammars (Corollar 1.1).

We remind to Definition 2.1 for $m = n = 1$.

Definition 3.1 *Let $G = (V_N, V_T, S, P)$ be a linear grammar, $\alpha \in V^*$. Then*

$$first\_last(\alpha) = \{(a,b) \mid \exists \alpha \overset{*}{\underset{G}{\Longrightarrow}} a\,v\,b,\ v \in V_T^*,\ a,\ b \in V_T\} \cup \{(\lambda,\lambda) \mid \alpha \overset{*}{\underset{G}{\Longrightarrow}} \lambda\}$$

Obvious, Theorem 2.1 becomes:

Theorem 3.1 *G is $LLin(1,1)$ grammar iff $first\_last(\beta_1) \cap first\_last(\beta_2) = \emptyset$, $\forall\ A \to \beta_1 \in P$, $\forall\ A \to \beta_2 \in P$, $\beta_1 \neq \beta_2$.*

The bidirectional parser $BP_{1,1}(G)$ (denoted simply by $BP(G)$) can also be reformulated (we present only the transition relation, # being a new nonterminal symbol):

1^0 **Expand transition:**

$(\#u\#, A, \pi) \vdash (\#u\#, \beta, \pi\, r)$ if $r = A \to \beta$ and the pair $({}^{(1)}u\#, \#u^{(1)}) \in first\_last(\beta)$

2^0. **Reduce transitions:**

a) $(\#v_1\, u\#, v_1\, A, \pi) \vdash (\#u\#, A, \pi),\ v_1 \in V_T^+$
b) $(\#u\, v_2\#, A\, v_2, \pi) \vdash (\#u\#, A, \pi),\ v_2 \in V_T^+$
c) $(\#v_1\, u\, v_2\#, v_1\, A\, v_2, \pi) \vdash (\#u\#, A, \pi),\ v_1,\, v_2 \in V_T^+$

3^0. **Acceptance transition:**

$(\#\#, \lambda, \pi) \vdash ACC$

4^0. **Rejection transition:**

$(\#u\#, \alpha, \pi) \vdash ERR$ if no transitions of type $1^0,\ 2^0,\ 3^0$ can be applied.

$BP(G)$ may be used in practical compiler applications, because for instance the computation of the sets $first\_last(\alpha)$ (α being right side part of a production) can be done in polynomial time complexity related to the dimension of input linear grammar G.

Example 3.1 *Let us review the grammar G_4 from Example 1.1.*

1. $S \to a\,S$

2. $S \to A\,b$

3. $A \to A\,b$

4. $A \to \lambda$

We can easily compute the sets:

- $first\_last(a\,S) = \{(a, b)\}$;
- $first\_last(A\,b) = \{(b, b)\}$;
- $first\_last(\lambda) = \{(\#, \#)\}$;

According to the Theorem 3.1, it follows that G_4 *is* $LLin(1,1)$ *grammar. Let us now consider the word* $w = a\,a\,b\,b\,b$. *The following transitions of* $BP(G_4)$ *can be:*

$$(\#a\,a\,b\,b\,b\#, S, \lambda) \vdash (\#a\,a\,b\,b\,b\#, a\,S, [1]) \vdash (\#a\,b\,b\,b\#, S, [1]) \vdash$$
$$\vdash (\#a\,b\,b\,b\#, a\,S, [1,1]) \vdash (\#b\,b\,b\#, S, [1,1]) \vdash (\#b\,b\,b\#, A\,b, [1,1,2]) \vdash$$
$$\vdash (\#b\,b\#, A, [1,1,2]) \vdash (\#b\,b\#, A\,b, [1,1,2,3]) \vdash (\#b\#, A, [1,1,2,3]) \vdash$$
$$(\#b\#, A\,b, [1,1,2,3,3]) \vdash (\#\#, A, [1,1,2,3,3]) \vdash (\#\#, \lambda, [1,1,2,3,3,4]) \vdash ACC$$

So, w *is "accepted" by* $BP(G_4)$, *and then according to Theorem 2.2, it follows that* $w \in L(G_4)$.

Next, we define two supplementary functions and two supplementary binary relations which will be used for determining the sets $first\_last(\alpha)$, where α is a right side part of a production of G. These are `first`, `last` : $V_N \to \mathcal{P}(V_T) \cup \{\lambda\}$ such that:

- $a \in$ `first`(A) iff there exists the derivation $A \overset{*}{\underset{G}{\Longrightarrow}} a\,\alpha$;
- $a \in$ `last`(A) iff there exists the derivation $A \overset{*}{\underset{G}{\Longrightarrow}} \alpha\,a$;
- $\lambda \in$ `first`(A) (or `last`(A)) iff there exists the derivation $A \overset{*}{\underset{G}{\Longrightarrow}} \lambda$.

and `begin`, `end` $\subseteq V \times V_N$ given by:

- X `begin` A iff there exists the production $A \to \beta\,X\,v$ and $\beta \overset{*}{\underset{G}{\Longrightarrow}} \lambda$, where $\beta \in V_N \cup \{\lambda\}$;
- X `end` A iff there exists the production $A \to u\,X\,\beta$ and $\beta \overset{*}{\underset{G}{\Longrightarrow}} \lambda$, where $\beta \in V_N \cup \{\lambda\}$.

The following lemma gives a procedure for obtaining the relations `begin` and `end`.

Lemma 3.1

1) If $Y\ \mathtt{begin}^n\ X$ *then there exists* $m,\ m \geq n$ *such that* $X \overset{m}{\underset{G}{\Longrightarrow}} Y\,\alpha$*;*

2) If $X \overset{n}{\underset{G}{\Longrightarrow}} Y\,\alpha$ *then there exists* $m,\ m \leq n$ *such that* $Y\ \mathtt{begin}^m\ X$*;*

3) $a\ \mathtt{begin}^*\ A$ *iff there exists a derivation* $A \overset{*}{\underset{G}{\Longrightarrow}} a\,\alpha$*;*

4) If $Y\ \mathtt{end}^n\ X$ *then there exists* $m,\ m \geq n$ *such that* $X \overset{m}{\underset{G}{\Longrightarrow}} \alpha\,Y$*;*

5) If $X \overset{n}{\underset{G}{\Longrightarrow}} \alpha\,Y$ *then there exists* $m,\ m \leq n$ *such that* $Y\ \mathtt{end}^m\ X$*;*

6) $a\ \mathtt{end}^*\ A$ *iff there exists a derivation* $A \overset{*}{\underset{G}{\Longrightarrow}} \alpha\,a$*;*

Proof Obviously, by induction on m or n. ■

Using Lemma 3.1, it is obvious that:

- $a \in \mathtt{first}(A)$ iff $a\ \mathtt{begin}^*\ A$;
- $a \in \mathtt{last}(A)$ iff $a\ \mathtt{end}^*\ A$.

The computation of $first\_last$ is presented as a returned value of the following self-explanatory recursive function.

Input: The linear grammar $G = (V_N, V_T, S, P)$
Output: $first\_last(\alpha),\ \alpha \in V^*$.

function $first\_last(\alpha)$;
begin
  if $(\alpha = \lambda)$ **then** $first\_last(\alpha) := \{(\#, \#)\}$;
  if $(\alpha = a,\ a \in V_T)$ **then** $first\_last(\alpha) := \{(a, a)\}$;
  if $(\alpha = a\,\beta\,b,\ a, b \in V_T)$ **then** $first\_last(\alpha) := \{(a, b)\}$;
  if $(\alpha = A\,u\,b,\ A \in V_N,\ u \in V_T^*,\ b \in V_T)$ **then** **begin**
    $first\_last(\alpha) := \{(a, b) \mid a \in \mathtt{first}(A) - \{\lambda\}\}$;
    if $(\lambda \in \mathtt{first}(A))$ **then** add to $first\_last(\alpha)$ the pair $({}^{(1)}u\,b, b)$;
  end ;
  if $(\alpha = a\,u\,A,\ a \in V_T,\ u \in V_T^*,\ A \in V_N)$ **then** **begin**
    $first\_last(\alpha) := \{(a, b) \mid a \in \mathtt{last}(A) - \{\lambda\}\}$;
    if $(\lambda \in \mathtt{last}(A))$ **then** add to $first\_last(\alpha)$ the pair $(a, a\,u^{(1)})$;
  end ;
  if $(\alpha = A,\ A \in V_N)$ **then** **begin**
    $set\_chain(A) := \{B \mid A \overset{*}{\underset{G}{\Longrightarrow}} B,\ B \in V_N\}$;
    $set\_fst\_snd := \emptyset$;
    for (any $A \rightarrow \beta \in P,\ B \in set\_chain(A)$) **do**
      if $(\beta \notin V_N)$ **then** $set\_fst\_snd := set\_fst\_snd \cup first\_last(\beta)$;
    $first\_last(\alpha) := set\_fst\_snd$
  end

end .

The function $first\_last$ needs polynomial time complexity (related to the dimension of G) because it describes (in a recursive manner) the transitive closure of the derivation relation from linear grammars.

As we can see in the following example, $first\_last(\alpha)$ is properly included in $first(\alpha) \times last(\alpha)$.

Example 3.2 *Let* $G = (\{S, A\}, \{a, b, c\}, S, \{S \rightarrow A,\ \ A \rightarrow a\,A\,b \,|\, b\,A\,a \,|\, c\})$ *be a linear grammar. Using, for instance, the function* $first\_last$*, we obtain* $first\_last(A) = \{(a, b), (b, a), (c, c)\}$*. On the other hand,* $\texttt{first}(A) = \{a, b, c\}$ *and* $\texttt{last}(A) = \{a, b, c\}$*. It results that* G *is* $LLin(1, 0)$ *(or* $LLin(0, 1)$*) grammar.*

4 Conclusions

According to the results related to $LLin(m, n)$ grammars, the following picture is valid:

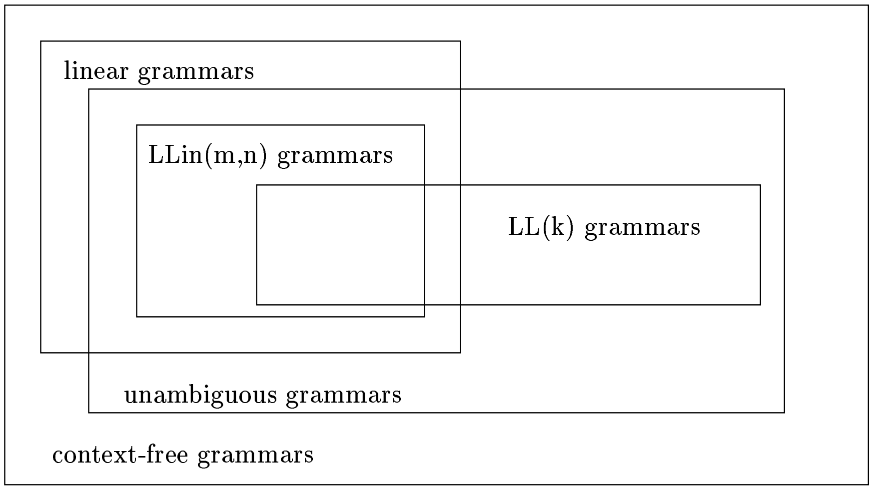

Figure 3

Without loss the generality, we allow three modifications of the bidirectional parser for testing the power of the device defined in Definition 2.2:

(i) we allow reading (and replacing) at the ends of the deque of two consecutive symbols (instead of only one);

(ii) we allow to interchange the contents of that two ends of the deque;

(iii) we shall remove the third component, i.e. the syntactic analysis (it does not interfere in the deterministic transitions).

With such modifications, we shall present an example of a bidirectional parser which can analyse the context sensitive language $L = \{a^n\, b^n\, c^n \mid n \geq 1\}$. In fact, we shall simulate the monotone grammar given by the following productions:

1. $A \to a\, A\, B\, c$
2. $A \to a\, b\, c$
3. $c\, B \to B\, c$
4. $b\, B \to b\, b$

As a initial configuration, we take $(\#w\#, A)$, where $w \in \{a, b, c\}^*$ is the input word. Assuming that the notations w and γ stand for words (of any length) over $\{a, b, c\}$, and $\{a, b, c, A, B\}$ respectively, the transitions will be the following:

1. $(\#a\, a\, w\, c\#, A\, \gamma) \vdash (\#a\, a\, w\, c\#, a\, A\, B\, c\, \gamma)$
2. $(\#a\, w\, c\#, a\, \gamma\, c) \vdash (\#w\#, \gamma)$
3. $(\#a\, w\#, a\, \gamma\, B) \vdash (\#w\#, \gamma\, B)$
4. $(\#a\, b\, w\#, A\, \gamma) \vdash (\#a\, b\, w\#, a\, b\, c\, \gamma)$
5. $(\#b\, w\, c\#, b\, \gamma\, B) \vdash (\#w\, c\#, \gamma\, B)$
6. $(\#b\, w\, c\#, c\, \gamma\, c\, B) \vdash (\#b\, w\, c\#, c\, \gamma\, B\, c)$
7. $(\#b\, w\, c\#, c\, B\, \gamma\, c) \vdash (\#b\, w\, c\#, B\, c\, \gamma\, c)$
8. $(\#b\, w\#, B\, \gamma) \vdash (\#b\, w\#, b\, \gamma)$
9. $(\#b\, w\, c\#, b\, \gamma\, c) \vdash (\#w\#, \gamma)$
10. $(\#b\, b\, w\, c\, c\#, c\, c\, \gamma\, B\, B) \vdash (\#b\, b\, w\, c\, c\#, B\, c\, \gamma\, B\, c)$
11. $(\#b\, c\#, c\, B) \vdash (\#b\, c\#, B\, c)$
12. $(\#\#, \lambda) \vdash ACC$
13. $(\cdot, \cdot) \vdash ERR$ - in the other cases.

Obviously, the above bidirectional parser is deterministic because at each step at most one transition may be applied. For instance, we may say that the parser is of type $(3, 3)$ because at the transition 11, we need to read three symbols from the left, and right, respectively.

We conclude that the subclass of $\mathcal{LL}in(m, n)$ is more powerful than some deterministic context-free languages, keeping the linear time complexity of the algorithm associated to the membership problem. In general, it does not mantain the main closure properties. In addition, we can formulate some open-problems, for instance the closure under complementation, intersection with regular languages and inverse homomorphism.

References:

1. Aho, A.V., Ullman, J.D.: *The Theory of Parsing, Translation, and Compiling.* Volume I: *Parsing*, Prentice Hall, 1972

2. Harrison, M. A.: *Introduction to Formal Language Theory.* Addison - Wesley Publishing Company, 1978

3. Hopcroft, J.E., Ullman, J.D.: *Introduction to Automata Theory, Languages and Computation.* Addison - Wesley Publishing Company, 1979

4. Knuth, D.E.: On the translation of languages from left to right. *Information Control.* 8: pp. 607-639 (1965)

5. Knuth, D.E.: Top-down analysis. *Acta Informatica.* 1: pp: 79-110 (1971)

6. Jucan, T, Andrei, Şt.: *Limbaje formale şi teoria automatelor. Culegere de probleme.* Editura Universităţii "Al. I. Cuza", Iaşi, 1997

7. Lewis, P.M., Stearns, R.: Syntax-directed transduction. *Journal of the ACM.* 15: pp. 464-488 (1968)

8. Salomaa, A.: *Formal Languages.* Academic Press. New York, 1973