Commutative Grammars and Permutation Grammars

Martin Tomko

Faculty of Information Technology, BUT

December 9, 2019

Table of contents

Motivation: Jumping Automata

Basic Terminology

Commutative Grammars: Definition

Permutation Grammars

Motivation: Jumping Automata

Jumping Finite Automata

- $M = (Q, \Sigma, R, s, F)$ – all with the same meaning as an ordinary finite automaton;
- The *jumping relation*:

$$xpaz \curvearrowright_M x'qz'$$

  where $pa \rightarrow q \in R$ and $xz = x'z'$;
- $L(M) = \{uv \mid u, v \in \Sigma^*, usv \curvearrowright_M^* f, f \in F\}$;
- The order of symbols in the input string essentially does not matter.[1]

---

[1] The situation is different in a *general jumping finite automaton*

Basic Terminology

Bags

- Informally: unordered strings;
- Formally, a *bag* over an alphabet V is a finite multiset of elements in V;
- The set of all bags over V is denoted by ${}^{*}V$;
- The empty bag is denoted by ε, ${}^{+}V = {}^{*}V \setminus \{\varepsilon\}$
  - ${}^{*}V$ can be defined as the free **commutative** monoid generated by V;
- Let $V = \{a_1, \ldots, a_k\}$. Any $w \in {}^{*}V$ can be written as

$$w = a_1^{i_1} \cdots a_k^{i_k}$$

where $i_j \in \mathbb{N}_0$ for $1 \leq j \leq k$

Parikh Mapping

- A function that maps a string to the number of occurences of each symbol;
- Let $V = \{a_1, \ldots, a_k\}$, where $k = |V|$:
- $\Psi_V : V^* \to \mathbb{N}_0^k$
- $\Psi_V(w) = (\#_{a_1}(w), \ldots, \#_{a_k}(w))$
  - The subscript V can be omitted when not necessary.
- Can also be defined for bags: $\Psi(a_1^{i_1} \cdots a_k^{i_k}) = (i_1, \ldots, i_k)$[2]
- Can be generalized to sets of strings / bags
- Can also be defined as $\Psi : V^* \to {}^*V$

[2] Note that this is a bijection.

Commutative Grammars: Definition

Commutative Grammars

- A *commutative grammar* is a 4-tuple $G^c = (N, T, S, P^c)$ where
  - N, T are disjoint finite alphabets, $V = N \cup T$,
  - $S \in N$ is a start symbol,
  - $P^c \subseteq {}^+N \times {}^*V$ is a finite set of production rules;
- $L(G^c) = \{w \in {}^*T \mid S \Rightarrow^*_{G^c} w\}$
- A commutative grammar G^c is
  - of *type 0* with no additional restrictions on P^c,
  - *context-sensitive* if $\alpha \to \beta \in P^c$ implies $|\alpha| \leq |\beta|$,
  - *context-free* if $P^c \subseteq N \times {}^*V$,
  - *regular* if $P^c \subseteq N \times {}^*T(N \cup \{\varepsilon\})$.

Comparing bags and strings: Ψ-equivalence

- Let G be a phrase-structure grammar and G^c a commutative grammar;
- G and G^c are Ψ-equivalent iff

$$\Psi(L(G)) = \Psi(L(G^c))$$

- Given $G = (N, T, S, P)$ and $G^c = (N, T, S, P^c)$, and for each $\alpha \to \beta \in P^c$ a rule $u \to v \in P$ such that $\Psi(\alpha) = \Psi(u)$ and $\Psi(\beta) = \Psi(v)$ does **not** imply that the grammars are Ψ-equivalent;
- Counterexample: Consider the rules $\{S \to BaC, BC \to b\}$
- The implication **does hold** for context-free grammars.

Related models: Petri nets

- *Petri nets* – a bag can represent the marking of a Petri net:
  - Each nonterminal represents a place
  - Each production rule represents a transition

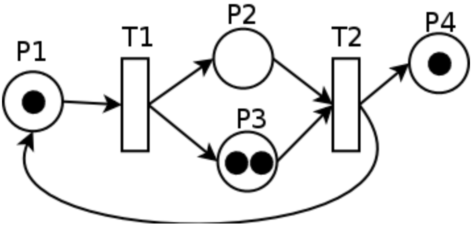

Image source: Wikimedia Commons

Related models: vector addition systems

- An n-dimensional vector addition system is a pair (r, W), where
  - $r \in \mathbb{N}_0^n$ is a vector of nonnegative integers,
  - $W \subseteq \mathbb{Z}^n$ is a finite set of integer vectors.
- The set $R(r, W)$ of reachable states:
  - Vectors of the form $r + \sum_{i=1}^{q} c_i$, $c_i \in W$, such that
  - $r + \sum_{i=1}^{k} c_i \in \mathbb{N}_0^n$ for all $1 \leq k \leq q$

Relation to Matrix Grammars

- For any commutative grammar G^c, there exists a Ψ-equivalent matrix grammar G, and conversely.

Permutation Grammars

Permutation Grammars: Definition

- A *permutation grammar* is a grammar $G = (N, \Sigma, P, S)$, where for each $r \in P$:
  a) r is a context-free rule $r : A \to \gamma$,
  b) r is a *permutation rule* $r : \alpha \to \beta$ where $\Psi(\alpha) = \Psi(\beta)$, $\alpha \neq \beta$;
- L(G) is called a *permutation language*;
- The class of all permutation languages is denoted by **Perm**;
- Clearly, **CF** $\subseteq$ **Perm** $\subseteq$ **CS**.

Basis Language

- Let $G = (N, \Sigma, P \cup R, S)$ be a permutation grammar, where
  - P only contains context-free rules,
  - R only contains permutation rules;
- Let $L = L(G)$, $G' = (N, \Sigma, P, S)$;
- Then $L^B = L(G')$ is a *basis language* of L wrt. G;
- The languages L and L^B are Ψ-equivalent.

Permutation languages: Example

- $L_1 = (w \in \{a, b, c\}^* \mid \#_a(w) = \#_b(w) = \#_c(w))$
- $L_1 = L(G_1)$, where $G_1 = (\{S, A, B, C, X\}, \{a, b, c\}, P_1, S)$, and P_1 contains:
  - $S \to \varepsilon \mid X$
  - $X \to ABCX \mid ABC$
  - $A \to a$
  - $B \to b$
  - $C \to c$
  - $AB \to BA$
  - $BA \to AB$
  - $AC \to CA$
  - $CA \to AC$
  - $BC \to CB$
  - $CB \to BC$
- $L_1 \in \textbf{Perm} \setminus \textbf{CF}$
- Note: $L_1^B = \{abc\}^*$

Permutation languages: Counterexample

- $L_2 = \{a^n b^n c^n \mid n \geq 1\}$
- No context-free infinite subset of L_2 exists – there is no possible basis language for L_2.
- $L_2 \in \mathbf{CS} \setminus \mathbf{Perm}$

Conclusion: $\mathbf{CF} \subset \mathbf{Perm} \subset \mathbf{CS}$

- The inclusions shown previously turn out to be proper:

$$\mathbf{CF} \subset \mathbf{Perm} \subset \mathbf{CS}$$

- Proof:
  - $\mathbf{CF} \subseteq \mathbf{Perm} \subseteq \mathbf{CS}$,
  - $L_1 \in \mathbf{Perm} \setminus \mathbf{CF}$,
  - $L_2 \in \mathbf{CS} \setminus \mathbf{Perm}$.

Generative Power: An infinite hierarchy

- A permutation rule $\alpha \to \beta$ is of length n if $|\alpha| = |\beta| = n$;
- A permutation grammar G is of order n if all its permutation rules are of length at most n;
- $\mathbf{Perm}_n$ denotes the class of languages generated by permutation grammars of order n;
- Clearly, $\mathbf{Perm}_2 \subseteq \mathbf{Perm}_3 \subseteq \mathbf{Perm}_4 \subseteq \cdots \subseteq \mathbf{Perm}$
- Furthermore, for all positive integers n,

$$\mathbf{Perm}_{4n-2} \subset \mathbf{Perm}_{4n-1}$$

- a