Contents

Chapter 4

Unambiguous grammars

4.1 Syntactic ambiguity

4.1.1 Ambiguity in natural languages

Ambiguity essential in natural languages.

The sentence “I saw a man on the hill with a telescope” admits up to five different readings.

In English, many the same word may act as different parts of speech, and thus many sentences can be read in different ways. Three interpretations of “time flies”: as a sentence that the time does fly, as a fragment describing a hypothetical genus of insects, and as a command to measure time in some relation to insects. Chomsky: “time flies like an arrow”, “fruit flies like a banana”. Adding one more: “swallow flies like a frog”.

Every grammar for a natural language must describe all these syntactic structures, and therefore has to be ambiguous.

4.1.2 Ambiguity in programming languages

Programming languages are designed to be unambiguous.

A good grammar should be unambiguous.

The following grammar for expressions is unsatisfactory.

$$E \to E + E \mid E * E \mid 1$$

But can be fixed by writing a more precise grammar.

Another unsatisfactory example, which found its way into the first version of Algol 60, before it was noted by Cantor [1].

$$S \to \text{if } E \text{ then } S \mid \text{if } E \text{ then } S \text{ else } S$$

(two parses of `if x then if y then s else t`) Unambiguity in the definition. Requires a special mention to disambiguate, and then a more precise grammar.

In C: `x=a,b; x=f(a); x=f(a, b);`

4.1.3 Unambiguous grammars

Definition 4.1. *An ordinary grammar G is called* unambiguous *if every string $w \in L(G)$ has a unique parse tree.*

Let a concatenation $L_1 \cdot \ldots \cdot L_k$ be called *unambiguous* if every string $w \in L_1 \cdot \ldots \cdot L_k$ admits a unique partition $w = u_1 \ldots u_k$ with $u_i \in L_i$.

Definition 4.1 in more detail. Note two types of ambiguity.

I. **Unambiguous choice of a rule:** if different rules for every single category symbol A generate disjoint languages.

II. **Unambiguous concatenation:** if for every rule $A \to X_1 \dots X_\ell$, the concatenation $L_G(X_1) \cdot \ldots \cdot L_G(X_\ell)$ is unambiguous, that is,

Definition 4.2. *An ordinary grammar is called unambiguous if it satisfies the above two conditions.*

For an ordinary grammar without useless symbols, Definition 4.1 and 4.2 are equivalent.

Example 4.1. *The language $\{\, a^k b^\ell c^m \mid k = \ell \text{ or } \ell = m\}$: is defined by the following grammar.*

$$\begin{array}{rcl} S & \to & AB \mid DC \\ A & \to & aA \mid \varepsilon \\ B & \to & bBc \mid \varepsilon \\ C & \to & cC \mid \varepsilon \\ D & \to & aDb \mid \varepsilon \end{array}$$

According to Definition 4.2, this grammar is ambiguous, because every string of the form $a^n b^n c^n$ can be obtained both as AB and as DC, and thus there is an ambiguity of choice between the rules $S \to AB$ and $S \to DC$.

Later it shall be proved that every ordinary grammar for this language is ambiguous.

Example 4.2. *Consider the following grammar for the language $\overline{\{\, ww \mid w \in \{a, b\}^*\}}$.*

$$\begin{aligned} &S \to AB \mid BA \mid O \\ &A \to XAX \mid a \\ &B \to XBX \mid b \\ &X \to a \mid b \\ &O \to XXO \mid X \end{aligned}$$

This grammar demonstrates both types of ambiguity. First, the choice between the rules $S \to AB$ and $S \to BA$ is ambiguous on such strings as $w = abba$. Secondly, the concatenation AB is ambiguous, as a string $aabb$ can be represented both as $a \cdot abb$ and as $aab \cdot b$. The concatenation BA is simularly ambiguous.

4.1.4 Unambiguous conjunctive and Boolean grammars

For Boolean grammars, Definition 4.1 is not applicable, because a parse tree contains only a partial information about the derivation of a string. Definition 4.2 works as follows.

Definition 4.3. *Let $G = (\Sigma, N, R, S)$ be a Boolean grammar. Then*

I. the choice of a rule in G is unambiguous, if different rules for every single nonterminal A generate disjoint languages, that is, for every string w there exists at most one rule $A \to \alpha_1 \,\&\, \dots \,\&\, \alpha_m \,\&\, \neg\beta_1 \,\&\, \dots \,\&\, \neg\beta_n$ with $w \in L_G(\alpha_1) \cap \dots \cap L_G(\alpha_m)$ and $w \notin L_G(\beta_1) \cup \dots \cup L_G(\beta_n)$.

II. concatenation in G is said to be unambiguous, if for every conjunct $\pm\alpha = \pm X_1 \dots X_\ell$, the concatenation $L_G(X_1) \cdot \ldots \cdot L_G(X_\ell)$ is unambiguous.

If both conditions are satisfied, the grammar is called unambiguous.

Example 4.1′. *The language* $\{ a^k b^\ell c^m \mid k = \ell \text{ or } \ell = m\}$ *from Example 4.1 is generated by the following unambiguous Boolean grammar.*

$$\begin{array}{rcl} S & \to & AB \mid DC \,\&\, \neg AB \\ A & \to & aA \mid \varepsilon \\ B & \to & bBc \mid \varepsilon \\ C & \to & cC \mid \varepsilon \\ D & \to & aDb \mid \varepsilon \end{array}$$

This transformation can be extended to a general method.

4.2 Limitations of unambiguous grammars

4.2.1 Combinatorial methods

Using pumping arguments to construct multiple parse trees for a single string, by pumping two different shorter strings. Ogden's lemma is particularly helpful (that was Ogden's original motivation).

Proposition 4.1 (Parikh [9]; Chomsky, Schützenberger [2]). *Every ordinary grammar that describes the language* $\{ a^i b^n c^n \mid i, n \geqslant 0\} \cup \{ a^m b^m c^j \mid m, j \geqslant 0\}$ *from Example 4.1 is ambiguous.*

Sketch of a proof. The proof follows the method of Ogden. Consider any grammar $G = (\Sigma, N, R, S)$ with $L(G) = L$. Let p be the constant given by Ogden's lemma. First, pump $w = a^p b^p c^{p+p!}$ with distinguished positions a^p to obtain $\widehat{w} = a^{p+p!} b^{p+p!} c^{p+p!}$. Secondly, pump $w' = a^{p+p!} b^p c^p$ with distinguished positions c^p to obtain the same string $\widehat{w}$.

The two resulting parse trees are different, because the first tree (the one obtained by pumping w) includes a subtree that contains at least $p!$ symbols b and at least $p!$ symbols a, but no symbols c. On the other hand, the second tree has a subtree with at least $p!$ symbols b and at least $p!$ symbols c, but no symbols a. No single tree with only $p + p!$ b-leaves could contain two such subtrees. □

Theorem 4.1. *Unambiguous ordinary languages are closed under intersection with regular languages.*

Proof. The same construction as for the whole class of ordinary grammars. It preserves unambiguity. □

Example 4.3. *The following language is not described by any unambiguous ordinary grammar.*

$$\{ a^{k_1} b \ldots a^{k_\ell} b \mid \ell \geqslant 1,\ k_1, \ldots, k_\ell \geqslant 0,\ \exists i : k_i = i\}$$

Sketch of a proof. □

Example 4.4 (Crestin). *The following language is not described by any unambiguous ordinary grammar.*

$$\{ w_1 w_2 \mid w_1, w_2 \in \{a, b\}^*,\ w_1 = w_1^R,\ w_2 = w_2^R\}$$

4.2.2 Analytic methods

Definition 4.4. *Let $L \subseteq \Sigma^*$ be a language. For each length $n \geqslant 0$, denote by c_n^L the number of strings of length n in L. Then the* generating function *of L is a complex function $f_L \colon \mathbb{C} \to \mathbb{C}$ defined by the following power series.*

$$f_L(z) = \sum_{n=0}^{\infty} c_n^L z^n.$$

Example 4.5. *The generating function of the language $L = \{\, a^n b^n \mid n \geqslant 0\}$ is*

$$f_L(z) = 1 + z^2 + z^4 + z^6 + \ldots = \frac{1}{1 - z^2},$$

defined on the open unit disk.

How do the operations on languages affect their generating functions.

Note that $c_n^{K \cup L} \leqslant c_n^K + c_n^L$, and if the union is unambiguous ($K \cap L = \varnothing$), then $a_n^{K \cup L} = c_n^K + c_n^L$. Similarly, $c_n^{KL} \leqslant \sum_{i=0}^n c_i^K \cdot c_{n-i}^L$, and if the concatenation is unambiguous, then $c_n^{KL} = \sum_{i=0}^n c_i^K \cdot c_{n-i}^L$. For unambiguous union and concatenation, $f_{K \cup L}(z) = f_K(z) \cup f_L(z)$ and $f_{K \cdot L}(z) = f_K(z) \cdot f_L(z)$.

Example 4.6. *The generating function of the language $L' = \{\, b^n c^{2n} \mid n \geqslant 0\}$ is*

$$f_{L'}(z) = 1 + z^3 + z^6 + z^9 + \ldots = \frac{1}{1 - z^3},$$

and therefore the generating function of the language $L \cup L' = \{\, a^m b^{m+n} c^{2n} \mid m, n \geqslant 0\}$ is

$$f_{L \cup L'}(z) = f_L(z) \cdot f_{L'}(z) = \frac{1}{(1 - z^2)(1 - z^3)}.$$

Theorem 4.2 (Chomsky, Schützenberger [2])**.** *If L is generated by an unambiguous ordinary grammar, then its generating function is algebraic.*

Proof. Transforming language equations defining a language into functional equations defining its generating function,

For each language variable $A \in N$, define the corresponding functional variable $f_A(z)$. For each language equation in the system,

$$A = \bigcup_{A \to X_1 \ldots X_\ell \in R} X_1 \cdot \ldots \cdot X_\ell,$$

define the corresponding functional equation, where each union becomes a sum, concatenations are translated to products, and every symbol of the alphabet is represented by a function $f(z) = z$.

$$f_A(z) = \sum_{A \to X_1 \ldots X_\ell \in R} \prod_{i=1}^{\ell} \left\{ \begin{array}{ll} f_{X_i}(z), & \text{if } X_i \in N \\ z, & \text{if } X_i \in \Sigma \end{array} \right\}$$

Each rule $A \to \varepsilon$ is counted as a constant $f(z) = 1$.

Example: $A = BC \cup aDb \cup \{\varepsilon\}$ to $f_A(z) = f_B(z) f_C(z) + z^2 f_D(z) + 1$.

Claim: if $(L_1, \ldots, L_n)$ is a solution of the system of language equations, then $(f_{L_1}, \ldots, f_{L_n})$ is a solution of the system of functional equations, which consists of algebraic functions by definition.

In particular, $f_{L(G)}(z)$ is an algebraic function. $\square$

Flajolet [3] demonstrated that many examples of languages have non-algebraic generating functions.

Example 4.7 (Flajolet [3])**.** *Consider the alphabet $\Sigma = \{a, b\}$ and the languages*

$$K = \{\, a^n b^{2n} \mid n \geqslant 1\}^* a^*,$$
$$L = a\{\, b^n a^{2n} \mid n \geqslant 1\}^* b^*.$$

Each of them is generated by an unambiguous grammar. However, the generating function of their union, $f_{K\cup L}(z)$, is a transcendental function, and therefore $K \cup L$ is not an unambiguous language.

Proof. The languages K and L have unambiguous grammars, and hence their generating functions f_K and f_L are algebraic. Consider that $f_{K\cup L}(z) = f_K(z) + f_L(z) - f_{K\cap L}(z)$. It is then sufficient to prove that the function $f_{K\cap L}(z)$ is transcendental.

The intersection $K \cap L$ equals

$$K \cap L = \{a, ab^2, ab^2a^4, ab^2a^4b^8, \ldots\},$$

and its generating function accordingly is

$$f_{K\cap L}(z) = \sum_{n=1}^{\infty} z^{2^n-1}.$$

To see that it is transcendental, consider any number $k \geqslant 2$. Then the value of $f_{K\cap L}(z)$ at a rational point $\frac{1}{k}$ is

$$f_{K\cap L}(\tfrac{1}{k}) = \sum_{n=1}^{\infty} \frac{1}{k^{2^n-1}}.$$

As proved by Liouville [7], this number is transcendental. Therefore, so is $f_{K\cup L}(\frac{1}{k})$, and then the function $f_{K\cup L}$ must be transcendental. ☐

Example 4.8 (Goldstine)**.** *Consider the language $\{\, a^{k_1}b \ldots a^{k_\ell}b \mid \ell \geqslant 1,\ k_1, \ldots, k_\ell \geqslant 0,\ \exists i : k_i \neq i\}$, which is defined by the following grammar. ***TBW*** Every ordinary grammar for this language is ambiguous.*

Example 4.9 (Petersen [10])**.** *Let $\Sigma = \{a, b\}$ and define the language of so-called* <u>primitive strings</u>*, that is, strings not representable as a power of any shorter string: $L = \overline{\{\, w^n \mid w \in \{a, b\}^*,\ n \geqslant 2\}}$. It is not known whether this language is ordinary, but there is a proof that there is no unambiguous ordinary grammar for this language.*

Exercises

4.2.1. Prove that every ordinary grammar generating the language $\{\, w_1w_2 \mid w_1, w_2 \in \{a, b\}^*,\ w_1 = w_1^R,\ w_2 = w_2^R\}$ is ambiguous.

4.2.2. Prove that there is no unambiguous ordinary grammar for the language $\{\, a^k b^\ell c^m d^n \mid k = m \vee \ell = n\}$ (this was the original example given by Parikh [9]).

4.3 Closure properties

Theorem 4.3. *The unambiguous languages are not closed under union and intersection.*

Proof. Consider the languages $L_1 = \{\, a^i b^n c^n \mid i, n \geqslant 0\}$ and $L_2 = \{\, a^m b^m c^j \mid m, j \geqslant 0\}$. Each of them is generated by an unambiguous grammar, which is a part of Example 4.1. However, their union $L_1 \cup L_2$ has no unambiguous ordinary grammar by Proposition 4.1, whereas their intersection $L_1 \cap L_2 = \{\, a^n b^n c^n \mid n \geqslant 0\}$ has no ordinary grammar at all. □

Theorem 4.4. *The unambiguous ordinary (conjunctive, Boolean) languages are closed under quotient with a single symbol.*

Proof. Let $G = (\Sigma, N, R, S)$ be a Boolean grammar in the binary normal form, let $a \in \Sigma$. Construct a new grammar $G' = (\Sigma, N \cup N', R \cup R', S')$, where $N' = \{\, A' \mid A \in N\}$ and the new rules are:

$$\begin{array}{ll} A' \to B_1C_1' \& \ldots \& B_mC_m' \& \neg D_1E_1' \& \ldots \& \neg D_nE_n' \& \neg\varepsilon & (A \to B_1C_1 \& \ldots \& B_mC_m \& \neg D_1E_1 \& \ldots \& \neg D_nE_n \& \neg\varepsilon \in \\ A' \to \varepsilon & (A \to a \in R) \end{array}$$

Then $L_{G'}(A) = L_G(A)$ and $L_{G'}(A') = L_G(A')a^{-1}$ for all $A \in N$, and, in particular, $L(G') = L(G)a^{-1}$. □

Theorem 4.5. *The unambiguous languages are not closed under concatenation with a two-element set.*

Proof. Assume that they are. Consider the language $L = \{\, a^m b^m c^n | m, n \geqslant 0\} \cup \{\, a^m b^n c^n d | m, n \geqslant 0\}$, which has an unambiguous grammar. Then the language

$$\big((L \cdot \{\varepsilon, d\}) \cap a^*b^*c^*d\big) \cdot d^{-1} = \{\, a^m b^m c^n \mid m, n \geqslant 0\} \cup \{\, a^m b^n c^n \mid m, n \geqslant 0\}$$

should have an unambiguous grammar as well, which contradicts Proposition 4.1. □

4.3.1 Non-closure under complementation

Theorem 4.6 (Hibbard, Ullian)**.** *The family of unambiguous languages is not closed under complementation.*

Bibliography

[1] D. J. Cantor, "On the ambiguity problem of Backus systems", *Journal of the ACM*, 9:4 (1962), 477–479.

[2] N. Chomsky, M. P. Schützenberger, "The algebraic theory of context-free languages", in: Braffort, Hirschberg (Eds.), *Computer Programming and Formal Systems*, North-Holland Publishing Company, Amsterdam, 1963, 118–161.

[3] Ph. Flajolet, "Analytic models and ambiguity of context-free languages", *Theoretical Computer Science*, 49 (1987), 283–309.

[4] S. Ginsburg, E. H. Spanier, "Bounded ALGOL-like languages", *Transactions of the AMS*, 113:2 (1964), 333–368.

[5] S. Ginsburg, E. H. Spanier, "Semigroups, Presburger formulas and languages", *Pacific Journal of Mathematics*, 16:2 (1966), 285–296.

[6] T. N. Hibbard, J. Ullian, "The independence of inherent ambiguity from complementedness among context-free languages", *Journal of the ACM*, 13:4 (1966), 588–593.

[7] J. Liouville, "Sur des classes très-étendues de quantités dont la valeur nŠest ni algébrique, ni même réductible à des irrationnelles algébriques", *Journal de mathématiques pures et appliquées, 1re série*, 16 (1851), 133–142.

[8] A. Okhotin, "Unambiguous Boolean grammars", *Information and Computation*, 206 (2008), 1234–1247.

[9] R. J. Parikh, "On context-free languages", *Journal of the ACM* 13:4 (1966), 570–581.

[10] H. Petersen, "On the language of primitive words", *Theoretical Computer Science*, 161:1–2 (1996), 141–156.

Index