Philippe Flajolet & Analytic Combinatorics: **Inherent Ambiguity of Context-Free Languages**

Frédérique Bassino and Cyril Nicaud

LIGM, Université Paris-Est & CNRS

December 16, 2011

I first met Philippe in 1996 (teaching an AofA course).

I first met Philippe in 1996 (teaching an AofA course).

I started my PhD in automata theory and read INRIA research reports for a year.

I first met Philippe in 1996 (teaching an AofA course).

I started my PhD in automata theory and read INRIA research reports for a year.

I attended the same course in 1998 !

I first met Philippe in 1996 (teaching an AofA course).

I started my PhD in automata theory and read INRIA research reports for a year.

I attended the same course in 1998 !

$$\frac{1}{2\pi i}\int \frac{f(z)}{z^{n+1}}dz$$

Philippe Flajolet & Analytic Combinatorics: **Inherent Ambiguity of Context-Free Languages**

Frédérique Bassino and Cyril Nicaud

LIGM, Université Paris-Est & CNRS

December 16, 2011

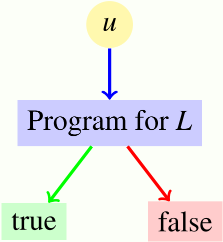
I. Context-free languages

- A word on a (finite) alphabet $A = \{a, b, \ldots\}$ is a (finite) sequence of letters : $u = aaba$, $v = bcbaa$, $w = aaaaab = a^5 b$.

- A word on a (finite) alphabet $A = \{a, b, \ldots\}$ is a (finite) sequence of letters : $u = aaba$, $v = bcbaa$, $w = aaaaab = a^5b$.
- The empty word ε is the word with no letter.

- A word on a (finite) alphabet $A = \{a, b, \ldots\}$ is a (finite) sequence of letters : $u = aaba$, $v = bcbaa$, $w = aaaaab = a^5 b$.
- The empty word ε is the word with no letter.
- A language is a set of words. It can be finite or infinite.

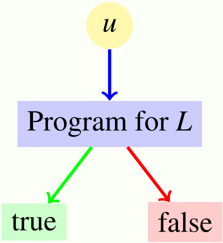
- A word on a (finite) alphabet $A = \{a, b, \ldots\}$ is a (finite) sequence of letters : $u = aaba$, $v = bcbaa$, $w = aaaaab = a^5b$.
- The empty word ε is the word with no letter.
- A language is a set of words. It can be finite or infinite.



Interested in languages L such that a machine can decide if $u \in L$:

- Turing machine
- Context-free languages
- Regular languages
- …

- A word on a (finite) alphabet $A = \{a, b, \ldots\}$ is a (finite) sequence of letters : $u = aaba$, $v = bcbaa$, $w = aaaaab = a^5b$.
- The empty word ε is the word with no letter.
- A language is a set of words. It can be finite or infinite.



Interested in languages L such that a machine can decide if $u \in L$:

- Turing machine
- **Context-free languages**
- Regular languages
- …

- A context-free grammar is a formal description of a context-free language. It is made of :
  - A finite set $V = \{S, X, Y, \ldots\}$ of variables.
  - A finite set $A = \{a, b, c, \ldots\}$ of terminals.
  - A starting axiom $S \in V$.
  - Rules of the form $X \to w$, where $X \in V$ and w is a sequence of symbols of $V \cup A$.

- A context-free grammar is a formal description of a context-free language. It is made of :
  - A finite set $V = \{S, X, Y, \ldots\}$ of variables.
  - A finite set $A = \{a, b, c, \ldots\}$ of terminals.
  - A starting axiom $S \in V$.
  - Rules of the form $X \to w$, where $X \in V$ and w is a sequence of symbols of $V \cup A$.

- The idea is to produce sequences of terminals only, by starting with S and by repeatedly applying the rules to the variables.

- A context-free grammar is a formal description of a context-free language. It is made of :
  - A finite set $V = \{S, X, Y, \ldots\}$ of variables.
  - A finite set $A = \{a, b, c, \ldots\}$ of terminals.
  - A starting axiom $S \in V$.
  - Rules of the form $X \to w$, where $X \in V$ and w is a sequence of symbols of $V \cup A$.

- The idea is to produce sequences of terminals only, by starting with S and by repeatedly applying the rules to the variables.

- Notation : $X \to aX \mid XY \mid YbbY$ instead of

$$\begin{cases} X & \to aX \\ X & \to XY \\ X & \to YbbY \end{cases}$$

Example 1

$$S \quad \rightarrow \quad aSbS$$

- $V = \{S\}$
- $A = \{a, b\}$
- $S \rightarrow aSbS \mid \varepsilon$

Example 1

- $V = \{S\}$
- $A = \{a, b\}$
- $S \to aSbS \mid \varepsilon$

$$\begin{array}{rcl} S & \to & aSbS \\ aSbS & \to & aSb \end{array}$$

Example 1

- $V = \{S\}$
- $A = \{a, b\}$
- $S \to aSbS \mid \varepsilon$

$$\begin{array}{ccc} S & \to & aSbS \\ aSbS & \to & aSb \\ aSb & \to & aaSbSb \end{array}$$

Example 1

- $V = \{S\}$
- $A = \{a, b\}$
- $S \to aSbS \mid \varepsilon$

S	$\to$	$aSbS$
$aSbS$	$\to$	aSb
aSb	$\to$	$aaSbSb$
$aaSbSb$	$\to$	$aabSb$
$aabSb$	$\to$	$aabaSbSb$
$aabaSbSb$	$\to$	$aababSb$
$aababSb$	$\to$	$aababb$

Example 1

- $V = \{S\}$
- $A = \{a, b\}$
- $S \to aSbS \mid \varepsilon$

$$
\begin{array}{ccc}
S & \to & aSbS \\
aSbS & \to & aSb \\
aSb & \to & aaSbSb \\
aaSbSb & \to & aabSb \\
aabSb & \to & aabaSbSb \\
aabaSbSb & \to & aababSb \\
aababSb & \to & aababb
\end{array}
$$

- *aababb* is in the language generated by the grammar.

- A context-free language is a language generated by a context-free grammar.
- Examples of context-free languages with $A = \{a, b, c\}$:

$$L_1 = \{a^n b^m c^k \mid n, m, k \geq 0\}$$
$$L_2 = \{a^n b^n c^m \mid n, m \geq 0\}$$

- Example of a language that is not context-free :

$$L_3 = \{a^n b^n c^n \mid n \geq 0\}$$

- A context-free language is a language generated by a context-free grammar.
- Examples of context-free languages with $A = \{a, b, c\}$:

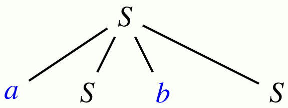
$$L_1 = \{a^n b^m c^k \mid n, m, k \geq 0\}$$
$$L_2 = \{a^n b^n c^m \mid n, m \geq 0\}$$

- Example of a language that is not context-free :

$$L_3 = \{a^n b^n c^n \mid n \geq 0\}$$

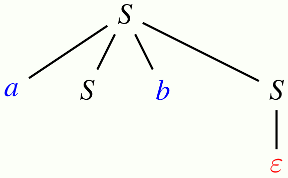
- The set of context-free languages is closed under union, concatenation and Kleene star.
- It is not closed under complementation and intersection.

Example 1



- $V = \{S\}$
- $A = \{a, b\}$
- $S \to aSbS \mid \varepsilon$

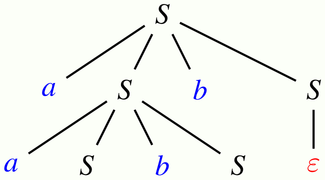
Example 1



- $V = \{S\}$
- $A = \{a, b\}$
- $S \to aSbS \mid \varepsilon$

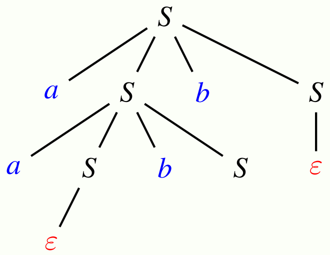
Example 1

- $V = \{S\}$
- $A = \{a, b\}$
- $S \to aSbS \mid \varepsilon$



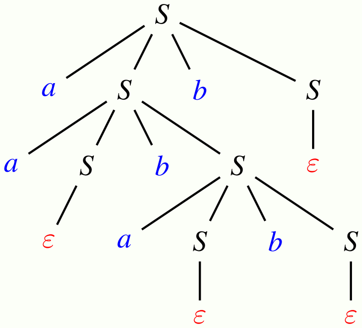
Example 1

- $V = \{S\}$
- $A = \{a, b\}$
- $S \to aSbS \mid \varepsilon$



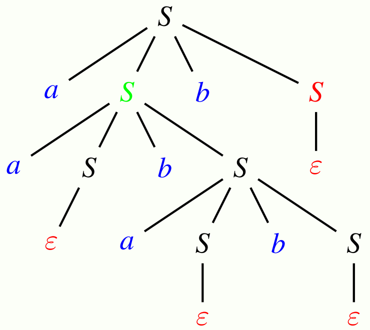
Example 1

- $V = \{S\}$
- $A = \{a, b\}$
- $S \to aSbS \mid \varepsilon$



Example 1

- $V = \{S\}$
- $A = \{a, b\}$
- $S \to aSbS \mid \varepsilon$



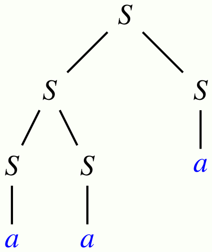
- The derivation tree of *aababb*.
- It is the **unique** derivation tree for *aababb*.

Example 2

- $V = \{S\}$
- $A = \{a\}$
- $S \to SS \mid a$

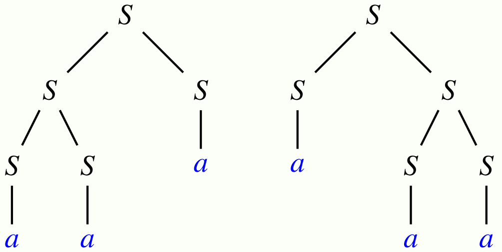
Example 2

- $V = \{S\}$
- $A = \{a\}$
- $S \to SS \mid a$



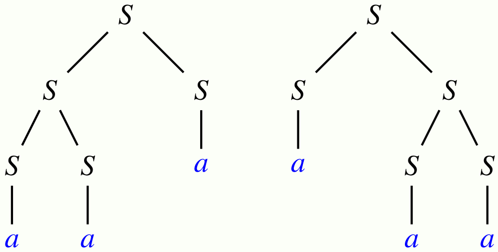
Example 2

- $V = \{S\}$
- $A = \{a\}$
- $S \to SS \mid a$



Example 2

- $V = \{S\}$
- $A = \{a\}$
- $S \to SS \mid a$



- The word *aaa* has two derivation trees.
- Every binary tree with $2n + 1$ nodes produces a^{n+1}.

- A grammar is ambiguous if there exists a word with at least two derivation trees in its generated language.
- A context-free language $\mathcal{L}$ is ambiguous (inherently ambiguous) if **every** grammar that generates $\mathcal{L}$ is ambiguous.

- A grammar is ambiguous if there exists a word with at least two derivation trees in its generated language.
- A context-free language $\mathcal{L}$ is ambiguous (inherently ambiguous) if **every** grammar that generates $\mathcal{L}$ is ambiguous.

- $\{a^n \mid n \geq 1\}$ is generated by $S \to SS \mid a$, which is an ambiguous grammar ...
- but $\{a^n \mid n \geq 1\}$ is also generated by the non-ambiguous $S \to Sa \mid a$, and is therefore a non-ambiguous language.

- A grammar is ambiguous if there exists a word with at least two derivation trees in its generated language.
- A context-free language $\mathcal{L}$ is ambiguous (inherently ambiguous) if **every** grammar that generates $\mathcal{L}$ is ambiguous.

- $\{a^n \mid n \geq 1\}$ is generated by $S \to SS \mid a$, which is an ambiguous grammar …
- but $\{a^n \mid n \geq 1\}$ is also generated by the non-ambiguous $S \to Sa \mid a$, and is therefore a non-ambiguous language.

- **Main focus :** sufficient conditions that ensure the ambiguity of a context-free language.

- Do ambiguous context-free languages exist ?

- Do ambiguous context-free languages exist ?
- Yes !

$$\{a^n b^m c^k \mid n = m \text{ or } m = k\}$$

- The original proof is combinatorial, using classical techniques of language theory (pumping lemmas, ...)

- Do ambiguous context-free languages exist ?
- Yes !

$$\{a^n b^m c^k \mid n = m \text{ or } m = k\}$$

- The original proof is combinatorial, using classical techniques of language theory (pumping lemmas, ...)

- Is the problem difficult ?

- Do ambiguous context-free languages exist ?
- Yes !

$$\{a^n b^m c^k \mid n = m \text{ or } m = k\}$$

- The original proof is combinatorial, using classical techniques of language theory (pumping lemmas, ...)

- Is the problem difficult ?
- Yes !
- Some languages seem to resist (discrete) combinatorial approaches
- The problem is undecidable : there is no algorithm to check whether a given context-free language is ambiguous.

II. From languages to functions

- The counting generating function of a language $\mathcal{L}$, is the formal power series (seen as a function) :

$$L(z) = \sum_{n \geq 0} \ell_n z^n,$$

where ℓ_n is the number of words of length n in $\mathcal{L}$.

- The function is analytic in a neighborhood of the origin : since $\ell_n \leq |A|^n$, we have

$$\frac{1}{|A|} \leq \rho \leq 1$$

- A function is algebraic (over $\mathbb{Q}$) when there exists a polynomial P with coefficients in $\mathbb{Q}$ such that $P(z, L(z)) = 0$. It is transcendental otherwise.

Theorem (Chomsky-Schützenberger)

The counting generating function of a non-ambiguous context-free language is algebraic over $\mathbb{Q}$.

Theorem (Chomsky-Schützenberger)

The counting generating function of a non-ambiguous context-free language is algebraic over $\mathbb{Q}$.

Proof :

$$\begin{cases} S \to XY \\ T \to aT \mid TbT \mid YcY \\ Y \to YaY \mid cY \mid abTaYYa \mid X \\ X \to a \mid b \mid c \end{cases}$$

Theorem (Chomsky-Schützenberger)

The counting generating function of a non-ambiguous context-free language is algebraic over $\mathbb{Q}$.

Proof :

$$\begin{cases} S \to XY \\ T \to aT \mid TbT \mid YcY \\ Y \to YaY \mid cY \mid abTaYYa \mid X \\ X \to a \mid b \mid c \end{cases} \Rightarrow \begin{cases} s(z) = x(z)y(z) \\ t(z) = zt(z) + zt(z)^2 + zy(z)^2 \\ y(z) = zy(z)^2 + zy(z) + z^4 t(z)y(z)^2 + x(z) \\ x(z) = 3z \end{cases}$$

Algebraic elimination gives

$$s(z)^8 - 27(z^3 - z^2)\, s(z)^5 + \ldots + 59049z^{10} = 0$$

Theorem (Chomsky-Schützenberger)

The counting generating function of a non-ambiguous context-free language is algebraic over $\mathbb{Q}$.

Theorem (Chomsky-Schützenberger)

The counting generating function of a non-ambiguous context-free language is algebraic over $\mathbb{Q}$.

Corollary

If the counting generating function is transcendental over $\mathbb{Q}$, then the language is ambiguous.

III. Transcendence

Transcendental numbers

- A number α is algebraic when there exists a polynomial P of $\mathbb{Q}[X]$ such that $P(\alpha) = 0$.
- $\sqrt{2}$ is algebraic, since it is a root of $X^2 - 2$.
- A number is transcendental when it is not algebraic.

- e is transcendental [Hermite 1873]
- π is transcendental [von Lindemann 1882]
- a^b is always transcendental for algebraic $a \notin \{0, 1\}$ and irrational algebraic b [Gelfond 1934] [Schneider 1935] (Hilbert's seventh problem).

- not known : $e + \pi$, e^e, $e\pi$, γ, ...

Transcendental functions

- It is usually easier to establish the transcendence of a function.
- Algebraic functions have some typical properties.
- Philippe gave several criteria to establish transcendence, using this properties.
- We shall see two of them in this talk.

Theorem

An algebraic function $L(z)$ over $\mathbb{Q}$ as finitely many singularities, which are algebraic numbers.

Criterion 1

A function having infinitely many singularities is transcendental.

Theorem (Puiseux+Transfert)

If $L(z)$ is an algebraic function over $\mathbb{Q}$ then

$$\ell_n \sim \frac{\beta^n n^s}{\Gamma(s+1)} \sum_{i=0}^{m} C_i \omega_i^n,$$

where $s \in \mathbb{Q} \setminus \{-1, -2, \ldots\}$, $\beta > 0$ is algebraic, the C_i and ω_i are algebraic, with $|\omega_i| = 1$.

Criterion 2

If the asymptotic of ℓ_n is of the form

$$\ell_n \sim \alpha\, \beta^n\, n^s,$$

with $s \notin \mathbb{Q} \setminus \{-1, -2, \ldots\}$, then the language is ambiguous.

IV. Ambiguous languages

Goldstine language

- Initial motivation for Philippe's paper.
- $G = \{a^{n_1}ba^{n_2}b \dots a^{n_p}b \mid p \geq 1, \exists i, n_i \neq i\}$
- $abaabaaab \notin G$ but $abaabaabb \in G$

Goldstine language

- Initial motivation for Philippe's paper.
- $G = \{a^{n_1} b a^{n_2} b \dots a^{n_p} b \mid p \geq 1, \exists i, n_i \neq i\}$
- $abaabaaab \notin G$ but $abaabaabb \in G$
- $A^* \setminus G = I \cup J$, with

$$I = \{ua \mid u \in A^*\}$$
$$J = \{\varepsilon\} \cup \{a^1 b a^2 b \dots a^p b \mid p \geq 1\}$$

- We obtain, using $|a^1 b a^2 b \dots a^p b| = \frac{n(n+1)}{2} - 1$, that

$$g(z) = \frac{1-z}{1-2z} - \sum_{n \geq 1} z^{n(n+1)/2-1}$$

Lacunary functions

- A lacunary function is an analytic function that cannot be analytically continued anywhere outside its circle of convergence.
- $f(z) = \sum_{n \geq 0} f_{\lambda_n} z^{\lambda_n}$, with $f_{\lambda_n} \neq 0$
- Sufficient conditions :
  - $\frac{\lambda_{n+1} - \lambda_n}{\lambda_n} \to \infty$ [Hadamard 1892]
  - $\frac{\lambda_{n+1} - \lambda_n}{\sqrt{\lambda_n}} \to \infty$ [Borel 1896]
  - $\lambda_{n+1} - \lambda_n \to \infty$ [Fabry 1896]
  - $\lambda_n / n \to \infty$ [Faber 1904]
- A lacunary function is transcendental (Criterion 1)

Goldstine language

- $G = \{a^{n_1} b a^{n_2} b \ldots a^{n_p} b \mid p \geq 1, \exists i, n_i \neq i\}$
- We obtained that

$$g(z) = \frac{1-z}{1-2z} - \sum_{n \geq 1} z^{n(n+1)/2-1}$$

- $\sum_{n \geq 1} z^{n(n+1)/2-1}$ is a lacunary function, hence $g(z)$ is transcendental.

Theorem (Flajolet)

The Goldstine language is ambiguous.

Another example

- Let Ω_3 be the context free language defined by

$$\Omega_3 = \{u \in \{a, b, c\}^* \mid |u|_a \neq |u|_b \text{ or } |u|_a \neq |u|_c\}$$

Another example

- Let Ω_3 be the context free language defined by

$$\Omega_3 = \{u \in \{a, b, c\}^* \mid |u|_a \neq |u|_b \text{ or } |u|_a \neq |u|_c\}$$

- Its complementary is

$$I = A^* \setminus \Omega_3 = \{u \in \{a, b, c\}^* \mid |u|_a = |u|_b = |u|_c\}$$

Another example

- Let Ω_3 be the context free language defined by

$$\Omega_3 = \{u \in \{a,b,c\}^* \mid |u|_a \neq |u|_b \text{ or } |u|_a \neq |u|_c\}$$

- Its complementary is

$$I = A^* \setminus \Omega_3 = \{u \in \{a,b,c\}^* \mid |u|_a = |u|_b = |u|_c\}$$

- Its counting generating function $O(z)$ satisfies

$$O_3(z) + \sum_{n \geq 0} \binom{3n}{n,n,n} z^{3n} = \frac{1}{1-3z}$$

- But using Stirling formula

$$\binom{3n}{n, n, n} \sim \frac{\sqrt{3}}{2\pi} \cdot 27^n \cdot n^{-1}$$

Criterion 2

If the asymptotic of ℓ_n is of the form

$$\ell_n \sim \alpha \, \beta^n \, n^s,$$

with $s \notin \mathbb{Q} \setminus \{-1, -2, \ldots\}$, then the language is ambiguous.

Theorem

The language Ω_3 is ambiguous.

Conclusion

- Need the counting generating function in some way
- Need to fulfill a criterion

Conclusion

- Need the counting generating function in some way
- Need to fulfill a criterion

- Solving computer science problems using analysis
- Solving discrete problems using continuous mathematics

Conclusion

- Need the counting generating function in some way
- Need to fulfill a criterion

- Solving computer science problems using analysis
- Solving discrete problems using continuous mathematics

- Beautiful ideas
- Exciting mathematics
- Simple proofs (relying on complicated earlier results)

Conclusion

- Need the counting generating function in some way
- Need to fulfill a criterion

- Solving computer science problems using analysis
- Solving discrete problems using continuous mathematics

- Beautiful ideas
- Exciting mathematics
- Simple proofs (relying on complicated earlier results)

- Analytic combinatorics for something else than asymptotic results.

I'm trying to get a unambiguous grammar that generates this context free language.

I’m trying to get a unambiguous grammar that generates this context free language.

You can’t, it’s inherently ambiguous !

I’m trying to get a unambiguous grammar that generates this context free language.

You can’t, it’s inherently ambiguous !

Why ?

I'm trying to get a unambiguous grammar that generates this context free language.

You can't, it's inherently ambiguous !

Why ?

Because π is a transcendental number.

I'm trying to get a unambiguous grammar that generates this context free language.

You can't, it's inherently ambiguous !

Why ?

Because π is a transcendental number.

That's why we are doing research !

I'm trying to get a unambiguous grammar that generates this context free language.

You can't, it's inherently ambiguous !

Why ?

Because π is a transcendental number.

That's why we are doing research !