

CONTEXT-FREE ERROR ANALYSIS BY EVALUATION OF ALGEBRAIC POWER SERIES

Ray Teitelbaum*
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa. 15213

ABSTRACT

Optimal error analysis with respect to a context-free language may be viewed as the evaluation of an algebraic power series. By generalization of the nodal span context-free recognition algorithm, any algebraic power series is computable in $O(n^3)$ steps. The closure of algebraic power series under sequential transduction yields a generous class of reasonable error measures for which optimal analysis is $O(n^3)$. Included is minimizing the number of symbol insertions, deletions and/or replacements needed for correction, a special case which has been studied separately.

INTRODUCTION

An important requirement of a programming language compiler is that it detect, locate and, if possible, correct syntax errors in badly formed programs. It is not uncommon for a compiler to diagnose the presence of many errors when, in fact, the programmer has made but a single mistake. Such cascades of spurious analysis may be avoided if a compiler is constrained to determine an "optimal" interpretation of a given erroneous input string.

Of course, the notion of optimality must be defined with respect to some quantitative model of the error making process. No particular model is being advocated here. Rather, it is suggested that many reasonable measures of error on context-free languages are algebraic power series [1]; for example, the minimal number of symbol changes needed to transform the input string back into the error-free language. Since the coefficients of any algebraic power series may be computed by a generalized nodal span recognition algorithm, optimal analysis for these notions of error is possible in $O(n^3)$ steps, where n is the length of the faulty input string.

ALGEBRAIC POWER SERIES [1]

Let R be a commutative semi-ring with identity. Denote the two associative, commutative operations of R by \odot and \oplus ; their identity elements by 1 and 0 , respectively. The distributive property, $i \odot (j \oplus k) = (i \odot j) \oplus (i \odot k)$ for all i, j, k in R , is crucial.

*This work supported by an IBM Graduate Fellowship.

Let $G = \langle V, T, P, S \rangle$ be a context-free grammar, where V and T are finite non-terminal and terminal alphabets, respectively; P is a finite set of productions in $V \rightarrow (VUT)^*$; S is a distinguished start symbol in V . If each production $X \rightarrow \alpha$ in P is associated with a weight i in R , denoted $X \rightarrow_i \alpha$, then G is called an R-weighted grammar.

Let Γ be a derivation tree for $X \Rightarrow^* t$, where X in V and t in T^* . Then the weight of Γ , denoted $w(\Gamma)$, is defined to be the product (\odot) of the weights of all production occurrences in Γ . If $w(\Gamma) = i$, then we shall also write $X \Rightarrow_i t$. For each X in V , the R-weighted grammar G induces a function $f_X: T^* \rightarrow R$, defined by

$$f_X(t) = \bigodot_{\Gamma} w(\Gamma)$$

where Γ ranges over all distinct derivation trees of t from X in G . If there is no derivation of t from X , then $f_X(t) = 0$. The function induced by the grammar, denoted f_G , is f_S .

If every rule $X \rightarrow \alpha$ in P has either $|\alpha| \geq 2$ or α in T , then G is called proper. When G is proper, no t in T^* has an infinite number of derivations, so f_G is well-defined. A function $f: T^* \rightarrow R$ is called an algebraic power series if there is some proper R-weighted grammar G such that $f = f_G$. If G is a right linear grammar, then f_G is called a rational power series. If $f = f_G$ except at the empty string, then we say that f_G is algebraic (rational) off ϵ .

Lemma 1. For every proper R-weighted grammar G , there exists a Chomsky Normal Form R-weighted grammar G' , such that $f_G = f_{G'}$.

Construction. The grammar G is transformed into G' as in the context-free case. For each τ in T , if τ occurs in any rule $X \rightarrow \alpha$ where $|\alpha| \neq 1$, then all such occurrences are replaced by a new non-terminal A_τ and rule $A_\tau \rightarrow_1 \tau$ is added to the productions.

Any rule $X \rightarrow_i Y_1 \dots Y_m$ ($m \geq 3$) is then replaced by the $m-1$ rules

$$\begin{aligned} X &\rightarrow_i Y_1 Z_1 \\ Z_1 &\rightarrow_1 Y_2 Z_2, \dots, Z_{m-3} \rightarrow_1 Y_{m-2} Z_{m-2} \\ Z_{m-1} &\rightarrow_1 Y_{m-1} Y_m \end{aligned}$$

where Z_1, \dots, Z_{m-2} are distinct new non-terminals. Since the weights of all auxiliary productions are 1, these additional steps in a derivation do not affect its total weight. Since the ambiguity of G is preserved in G' , $f_G = f_{G'}$. #

EVALUATION

Cocke's nodal span context-free language recognition algorithm can be generalized in a straightforward way to compute any algebraic power series [2].

Theorem 1. Any algebraic power series $f: T^* \rightarrow R$ can be evaluated at string t in T^* in $O(|t|^3)$ steps on a random access computer.

Proof. By Lemma 1, there exists some Chomsky Normal Form grammar $G = \langle V, T, P, S \rangle$ such that $f = f_G$.

Let t be some string in T^* . We may assume that t is not ϵ , since otherwise $f(t) = f(\epsilon) = 0$ immediately.

We assume the use of a two-dimensional, random access data structure F which may be indexed by each non-terminal X in V and each non-empty subinterval s of t . When specifying indices for F , distinct subintervals s of t are distinct even if their values as substrings are equal. Each element $F[X, s]$ is of type R and is used to tabulate the value of $f_X(s)$.

We begin by initializing F to 0, the identity of \oplus in R . Since there are $|t| \cdot (|t|+1)/2$ substrings of t and $|V|$ is constant, this initialization of F is performed in $O(|t|^2)$ steps.

Suppose $|t| = n$ and s is some substring of t of length 1, i.e., s is some terminal symbol in T . Then, since G is a Chomsky Normal Form grammar, any derivation $X \Rightarrow s$ in G must consist of the single application of some production $X \rightarrow_i s$ in P . Thus,

$$f_X(s) = \sum_{\substack{X \rightarrow_i s \\ \text{in } P}} i.$$

Accordingly, the base step of the algorithm computes $f_X(s)$, for all X in V and symbols s of t , in $n \cdot |P|$ steps:

for each of the n symbols s of t do
for each production $X \rightarrow_i \tau$ in P do
if $s = \tau$ then $F[X, s] := F[X, s] \oplus i$.

Now consider larger substrings s of t . Suppose $|s| = k \geq 2$. Because G is a Chomsky Normal Form grammar, any derivation of s from X in G must be of the form $X \Rightarrow_i YZ \xRightarrow{*} s$. Invoking the distributivity in R of product over sum, we see that:

$$\begin{aligned} f_X(s) &= \sum_{\Gamma: X \xRightarrow{*} s} w(\Gamma) \\ &= \sum_{\substack{X \rightarrow_i YZ \\ \text{in } P}} \sum_{uv=s} \sum_{\Gamma_1: Y \xRightarrow{*} u} \sum_{\Gamma_2: Z \xRightarrow{*} v} i \otimes w(\Gamma_1) \otimes w(\Gamma_2). \end{aligned}$$

$$= \sum_{\substack{X \rightarrow_i YZ \\ \text{in } P}} \sum_{uv=s} i \otimes f_Y(u) \otimes f_Z(v).$$

Thus, from the values of F for all substrings of s shorter than k , we may compute $f_X(s)$. Since there are $k-1$ ways of representing $s = uv$ (we know there are no derivations of ϵ in G), the number of operations required is no more than $|P| \cdot (k-1) \cdot 3$. There are $n-k+1$ distinct substrings s of t of length k . Thus, F may be computed for all substrings of length k and all X in V in $(n-k+1) \cdot |P| \cdot (k-1) \cdot 3$ steps, given the values of F for all shorter substrings:

for each of the $n-k+1$ substrings s of t of length k do
for each production $X \rightarrow_i YZ$ in P do
for each of the $k-1$ partitions $uv = s$ do
 $F[X, s] := F[X, s] \oplus (i \otimes F[Y, u] \otimes F[Z, v])$.

Hence, to compute $f(t) = f_S(t) = F[S, t]$ requires

$$3 \cdot |P| \cdot \sum_{k=2}^n (n-k+1) \cdot (k-1) = |P| \cdot (n-1) \cdot n \cdot (n+1)/2 = O(n^3)$$

steps, plus $O(n^2)$ for initialization and $O(n)$ for the base step. Thus, $f(t)$ is $O(|t|^3)$ computable. #

Corollary 1. Any rational power series $f: T^* \rightarrow R$ can be evaluated at string t in T^* in $O(|t|)$ steps.

Proof. Suppose $X \rightarrow_i YZ$ is a production of the Chomsky Normal Form grammar constructed by Lemma 1 from a right linear grammar for f . Then clearly, Y only derives some terminal symbol in T . Therefore

$$\begin{aligned} f_X(s) &= f_X(\sigma s_1) \\ &= \sum_{X \rightarrow_i YZ} \sum_{Y \rightarrow_j \sigma} \sum_{\Gamma: Z \xRightarrow{*} s_1} i \otimes j \otimes w(\Gamma) \\ &= \sum_{X \rightarrow_i YZ} \sum_{Y \rightarrow_j \sigma} i \otimes j \otimes f_Z(s_1). \end{aligned}$$

Thus, F need only tabulate $f_X(s)$ for suffixes s of t and all X in V . Given the value in F for the next shorter suffix s_1 of t , $f_X(s)$ is computable in a constant number of steps. Thus, proceeding from right to left, $f(t) = F[S, t]$ is $O(|t|)$ computable. #

OPTIMIZING SEMI-RINGS

Many of the closure properties of context-free languages have analogies in the theory of algebraic power series. Because some operations may lead to necessarily improper weighted grammars, the results are often restricted by some limiting condition. But if the semi-ring is such that every induced function is well-defined, even when the underlying grammar is improper, then these restrictions may be relaxed.

A semi-ring R will be called optimizing if 1, the identity of \otimes in R , satisfies $1 \otimes i = i$ for all i in R .

For example, consider the set

$N^+ = \{\text{non-negative integers}\} \cup \{\infty\}$ with operations defined in the following table:

R	\ominus	\oplus	$\frac{1}{}$	$\frac{0}{}$
N^+	+	min	0	∞

Then N^+ is an optimizing semi-ring.

If the context-free grammar G of a given language is identified with the N^+ -weighted grammar where every production is associated with weight 0 in N^+ , then $f_G^{-1}(0)$ is $L(G)$, the error-free language. A quantitative, generative model of errors is provided by the addition to G of productions with non-zero weight. These rules may be language dependent, for example,

<primary> \rightarrow_j begin <conditional expression> end

describes a common mistake in Algol-60. Or the error rules may reflect the uniform error characteristics of the input device, for example, some weighted confusion matrix for characters on a keyboard. The weight of a production may be interpreted as the number of errors incurred in using that rule in a derivation.

The function $f_G(t)$ determines the minimum number of errors in which t can be derived. The strings for which the N^+ -weighted grammar still provides no explanation lie in $f_G^{-1}(\infty)$.

In the light of the following theorem, when adding error productions, the restriction to proper rules can be ignored and the induced minimal error function $f_G: T^* \rightarrow N^+$ remains algebraic (off \in).

Theorem 2. If R is an optimizing semi-ring, then every R -weighted grammar G induces an algebraic power series (off \in).

Construction. We show the bookkeeping of weights required in the construction of a proper grammar G' such that $(\forall \text{ in } T^* - \{\in\})[f_{G'}(t) = f_G(t)]$. The procedure is just a generalization of the removal of \in and chain rules from a context-free grammar.

Clearly, it is never necessary to have two productions which are identical except for their weights. If $X \rightarrow_i \alpha$ and $X \rightarrow_j \alpha$ are both in a set of productions, then for every derivation $\Gamma_1: S \xrightarrow{*} \beta X \gamma \xrightarrow{*} \beta \alpha \gamma \xrightarrow{*} t$ there is a derivation $\Gamma_2: S \xrightarrow{*} \beta X \gamma \xrightarrow{*} \beta \alpha \gamma \xrightarrow{*} t$. Since $f_G(t)$ includes in its sum both $w(\Gamma_1)$ and $w(\Gamma_2)$, by the distributive law, the single production $X \rightarrow_{i \oplus j} \alpha$ will suffice. Therefore, in the construction below, rules may be "appended" to a set of productions in the sense of a set-theoretic union with summation (\oplus) over the weights. For example, the result of appending the weighted production $A \rightarrow_3 a$ to the set $\{A \rightarrow_2 a, A \rightarrow_2 b\}$ is the set $\{A \rightarrow_{3 \oplus 2} a, A \rightarrow_2 b\}$.

From the distributive law and the condition that $1 \oplus i = 1$ for all i in R , it follows that

$$i \oplus (i \ominus j) = i$$

for every i and j in R .

Step 1. Elimination of \in -rules. For each Y in V , let $i_Y = f_Y(\in)$, i.e., $i_Y = \bigoplus_{\Gamma} w(\Gamma)$ where Γ ranges

over all distinct derivation trees for $Y \xrightarrow{*} \in$ in G . If \in cannot be derived from Y , then $i_Y = 0$ by definition. We must show that the value of i_Y is well-defined, even if G is not proper. Any derivation tree for $Y \xrightarrow{*} \in$ which is deeper than the cardinality of V must contain a path with some non-terminal X repeated. But any derivation $X \xrightarrow{*} \alpha \beta \gamma \xrightarrow{*} \alpha \beta \in \xrightarrow{*} \alpha \beta \in$ on such a path may be replaced by $X \xrightarrow{*} \alpha \beta \in \xrightarrow{*} \alpha \beta \in$ - the resulting tree is another a derivation of \in from Y . Suppose the weight of the rest of the derivation tree, the same in both cases, is j . The total weight of both of these derivation trees enters into the formal sum for i_Y . However, by the condition on R , $(j \oplus i_1 \oplus i_2 \oplus i_3) \oplus (j \oplus i_3) = j \oplus i_3$, so the longer derivation has no effect on the value of i_Y . Thus, i_Y can be computed by summing only over derivation trees Γ of \in from Y which are no deeper than the cardinality of V . But there are only a finite number of such "shallow" trees. Therefore, the value of i_Y may be determined by exhaustive examination of all such "shallow" trees.

We construct an intermediate grammar $G'' = \langle V, T, P'' \rangle$ with no \in -rules such that $f_{G''} = f_G$ off \in . Let P'' be initially empty. Then, for each rule $X \rightarrow_i \alpha$ in P , append to P'' all rules $X \rightarrow_{\beta} \alpha$, $\beta \neq \in$, which can be obtained from $X \rightarrow_i \alpha$ by deletion of zero or more non-terminals Y_1, \dots, Y_k from α for which $i_{Y_1}, \dots, i_{Y_k} \neq 0$. If $\alpha = \alpha_0 Y_1 \alpha_1 Y_2 \alpha_2 \dots Y_k \alpha_k$ and Y_1, \dots, Y_k are deleted, then the rule which is appended to P'' is $X \rightarrow_{\alpha_0 \dots \alpha_k}$ with a weight of $i \ominus i_{Y_2} \ominus \dots \ominus i_{Y_k}$.

Step 2. Elimination of rules $X \rightarrow_i Y$. For each X and Y in V , compute $i_{XY} = \bigoplus_{\Gamma} w(\Gamma)$ where Γ ranges

over all distinct derivation trees of Y from X in G'' . Let $i_{XX} = 1$. By reasoning similar to Step 1, i_{XY} is computable. (Because P'' has no \in -rules, every derivation of Y from X in G'' is linear. Thus, by the condition on R , only the finite number of derivations which are no longer than the cardinality of V can affect the value of i_{XY} .)

We construct the proper grammar $G' = \langle V, T, P', S \rangle$ from G'' as follows. Let P' be initially empty. For every X in V and each rule $X \rightarrow_i \alpha$ in P'' , where α not in V and $i_{XY} \neq 0$, the rule $X \xrightarrow{i \ominus i_{XY}} \alpha$ is appended to P' . #

The errors considered in [3,4, and 5], omission, insertion, and replacement of symbols, are easily expressed in N^+ -weighted productions. Assuming (without loss of generality) all occurrences of terminal symbols σ are isolated into productions of their own, we may augment an arbitrary G to G' by adding, for each terminal rule $X \rightarrow \sigma$, the following N^+ -weighted error rules:

omission: $X \rightarrow_1 \in$
insertion: $X \rightarrow_0 XA$
where $A \rightarrow_0 \tau$
and $A \rightarrow_1 \tau A$ ($\forall \tau$ in T)
replacement: $X \rightarrow_1 \tau$ ($\forall \tau$ in T).

Finally, to allow symbols to be inserted erroneously at the very beginning of a string, add

$S \rightarrow_0 A S$. Clearly, $f_{G_0}(t)$ is the fewest insertion, omission and/or replacement errors which permit t to be derived from S in G . By Theorems 2 and 1, $f_{G_0}(t)$ is algebraic, and therefore computable in $O(|t|^3)$ steps. Furthermore, by modifying the Cocke recognition algorithm in the usual way to preserve a derivation tree, we obtain a minimal error parse. From this parse and the natural inverses of each error, we may arrive at a minimal error correction.

But this particular optimal error measure is just a special case of an N^+ -weighted sequential transduction. Because algebraic power series are closed under such mappings, any error measure which may be so expressed is computable in $O(n^3)$ steps.

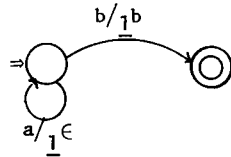
WEIGHTED TRANSDUCERS

An R-transducer is a non-deterministic finite state machine with output and weights associated with each arc [1]. It consists of a set of states Q , distinguished subsets of initial and final states, a finite set of arcs D contained in $Q \times T^* \times T^* \times Q$, and an association of weights to arcs $w: D \rightarrow R$. If (q, t, t', q') is an arc in D , then t and t' are called its input and output, respectively. A sequence ρ in D^* of connected arcs, beginning at an initial state and ending at a final state, is called a path. If d is in D (not necessarily a path), then $w(d)$, input (d) and output (d) are respectively the product (\odot) of the weights, the concatenation of the inputs and the concatenation of the outputs of the constituent arcs in d .

Given an R-transducer M from T^* to T'^* and an algebraic power series $f: T^* \rightarrow R$, we define $Mf: T'^* \rightarrow R$ by

$$Mf(t') = \left(\sum_{\rho} f(\text{input}(\rho)) \odot w(\rho) \right)$$

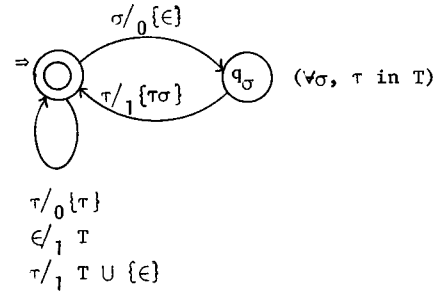
where ρ ranges over all paths of M with output $(\rho) = t'$. If arcs with empty output are allowed in D , then this sum may be infinite and not well-defined. For instance, if G has productions $\{S \rightarrow_1 aS, S \rightarrow_1 b\}$ and M is



then $Mf_G(b) = 1 \oplus 1 \oplus \dots$. M is called proper if no arc in D has ϵ -output. It can be shown that if f is algebraic and M is proper, then Mf is also algebraic [1][†].

When R is an optimizing semi-ring, the restriction to proper transducers can be relaxed. Thus, any minimal error measure which can be expressed as an N^+ -transduction is algebraic (off ϵ) and therefore $O(n^3)$ computable. For example, the errors insertion, omission and replacement are trivially expressed by an N^+ -transduction, as is the non-overlapping transposition of adjacent symbols.

[†]Although Shamir's definition of proper in [1] is that no arc in M has ϵ -input, he probably intended our definition since otherwise the theorem is false.



Theorem 3. If R is an optimizing semi-ring, f an algebraic (rational) power series and M an R -transducer, then Mf is algebraic (rational) off ϵ .

Proof (sketch). We may assume $D \subseteq Q \times T \cup \{\epsilon\} \times T^* \times Q$ (if necessary, by adding states to Q and breaking-up transitions on input strings longer than one). Exactly as in Shamir [1], the function $f_M: D^* \rightarrow R$ such that $f_M(d) = f(\text{input}(d)) \odot w(d)$ if d in D^* is a path and 0 otherwise, is algebraic (rational). To get Mf , apply the homomorphism output: $D^* \rightarrow T'^*$ to f_M . If R were not an optimizing semi-ring, then this step might fail to produce an algebraic (rational) power series, since closure is guaranteed only for ϵ -free homomorphisms. This is why, for arbitrary R , the transducer must be proper, i.e., no ϵ -output arcs. However, because R is an optimizing semi-ring, from Theorem 2, one shows that erasing homomorphisms, such as output, preserve algebraic (rational) power series. #

CONCLUSION

The purpose of this note has been to point out that minimal error analysis, as considered in [3, 4 and 5], is an instantiation of a general result already implicit in the papers of Shamir [1] and Cocke [2]. We have pointed out the relevance of the theory of algebraic power series in non-commuting variables in order to minimize further piecemeal rediscovery. Although our examples have been in N^+ (minimizing the number of errors) we observe that all results apply in the optimizing semi-ring of probabilities under product and max.

REFERENCES

- [1] Shamir, Eliahu, "A Representation Theorem for Algebraic and Context-Free Power Series in Noncommuting Variables," Information and Control (11), 1967, 239-254.
- [2] Aho, A. V. and J. D. Ullman, The Theory of Parsing, Translation, and Compiling, Prentice-Hall, 1972.
- [3] Teitelbaum, Ray, "Diagnosis of Syntax Errors," February 1972, unpublished note.
- [4] Lyon, Gordon, "Least-errors Recognition of Mutated Context Free Sentences in Time $n^3 \log n$," Proc. of the Sixth Annual Princeton Conf. on Info. Sci. and Sys., March 1972.
- [5] Aho, A. V. and T. G. Peterson, "A Minimum Distance Error Correcting Parser for Context Free Languages," SIAM J. on Computing, Dec. 1972.