Separating Words With Automata and Grammars

Jeffrey Shallit
Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada
shallit@graceland.uwaterloo.ca
http://math.uwaterloo.ca/~shallit

This is joint work with

- J. Currie (University of Winnipeg)
- H. Petersen (University of Stuttgart)
- J. M. Robson (University of Bordeaux)

An electronic copy of these slides can be found at
http://math.uwaterloo.ca/~shallit/talks.html

Separating Automata With Words

The following theorem says, roughly speaking, if two DFA's accept different languages, then there is a "short" string accepted by one but not the other.

Theorem (Moore, 1956). *Suppose* $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ *and* $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ *are two deterministic finite automata such that* $L(M_1) \neq L(M_2)$. *Then there exists a string*

$$w \in (L(M_1) - L(M_2)) \ \cup \ (L(M_2) - L(M_1))$$

with

$$|w| \leq \mathrm{card}(Q_1) + \mathrm{card}(Q_2) - 2.$$

By $\mathrm{card}(Q)$ we mean the cardinality of the set Q.

The bound in Moore's theorem is best possible, even over a unary alphabet.

Separating Grammars with Words

There is no counterpart of Moore's theorem for context-free grammars. We have

Proposition. *There is no computable* $f : \mathbb{N} \to \mathbb{N}$ *with the following property: if* G_1, G_2 *are context-free grammars with* $L(G_1) \neq L(G_2)$ *and* $\mathrm{ds}(G_1), \mathrm{ds}(G_2) \leq n$, *then there exists* w *with* $|w| \leq f(n)$ *such that*

$$w \in (L(G_1) - L(G_2)) \ \cup \ (L(G_2) - L(G_1)).$$

Proof. Assume that such a function exists. Then a Turing machine could test if $L(G_1) = L(G_2)$ by first computing $b = f(\max(\mathrm{ds}(G_1), \mathrm{ds}(G_2)))$ and then testing membership of w in $L(G_1)$ and $L(G_2)$ for all $|w| \leq b$. If such a w is found with

$$w \in (L(G_1) - L(G_2)) \ \cup \ (L(G_2) - L(G_1)),$$

then accept; otherwise reject. This would give a decision procedure for testing equality of context-free grammars, which is well-known to be recursively unsolvable. ∎

The above proof works if ds denotes any measure of descriptional complexity of context-free grammars that is computable and for which there are only a finite number of distinct grammars G with $\mathrm{ds}(G) \leq n$.

The Inverse Problem: Separating Words with Automata

- We are given two distinct words w and x with $|w|, |x| \leq n$.
- We want to find a "small" DFA that accepts one word but not the other.
- Such a DFA is called a separating DFA.
- Problem first studied by Goralčík and Koubek [1986].
- They proved a separating DFA exists with $O(\log n)$ states if $|w| \neq |x|$. This is best possible.
- They sketched a proof that a separating DFA exists with $o(n)$ states if $|w| = |x|$.
- This upper bound was later improved by Robson [1989] to

$$O(n^{2/5}(\log n)^{3/5}).$$

The case of unequal lengths

Theorem. If $|w| \neq |x|$ and $|w|, |x| \leq n$, then there exists a DFA with $O(\log n)$ states accepting w but not x.

Proof. We need the following lemma:

If i, j are unequal non-negative integers $\leq n$ and $n \geq 2$, then there exists a prime number $p \leq 4.4 \log n$ with $i \not\equiv j \pmod{p}$.

- Let $i = |w| \bmod p$.
- Then $w \in (\Sigma^p)^* \Sigma^i$ and $x \notin (\Sigma^p)^* \Sigma^i$.
- Then we can accept w and reject x by using a "cycle" automaton with p states, one for each residue class, with the state corresponding to i made accepting. ∎

The case of unequal lengths: lower bound

Theorem. Let N be an integer ≥ 1, and let $n = \mathrm{lcm}(1, 2, \ldots, N)$. Then no DFA with $\leq N$ states can separate a^n from a^{2n}.

Proof.

- Suppose M is a DFA that separates a^n from a^{2n}.
- Without loss of generality we may assume M accepts a^n (if not, interchange final and nonfinal states).
- If M has $\leq N$ states, then by the pumping lemma we can write $a^n = a^i a^j a^{n-(i+j)}$ with $1 \leq j \leq N$.
- Then $a^{n+(k-1)j} \in L(M)$ for all $k \geq 0$.
- Now $j \mid n$, so we can take $k = \frac{n}{j} + 1$ to see that $a^{2n} \in L(M)$. ∎

The lower bound of $\Omega(\log n)$ in the case of unequal lengths now follows when we observe that the prime number theorem gives $n = e^{N(1+o(1))}$.

The case of equal lengths

As stated above, Robson can separate equal-length strings of length $\leq n$ using $O(n^{2/5}(\log n)^{3/5})$ states. The ideas behind this proof are quite intricate. However, the idea behind a slightly weaker bound is easier to state.

Theorem. Any two distinct strings $w, x \in (0+1)^n$ can be separated by a DFA with $2m$ states, where $m = O(\sqrt{n})$.

Proof Sketch. Given two distinct strings $w, x \in (0+1)^n$, there exists an integer $m = O(\sqrt{n})$ and an integer y, $0 \leq y < m$ such that the total number of 1's at positions congruent to $y \pmod m$ is of different parity in w and x.

We can now use a mod-m counter combined with a mod-2 counter (and hence $2m$ states) to separate w and x. ∎

Separating words with grammars: Definition of description size

Let $G = (V, \Sigma, P, S)$ be a context-free grammar. We define the description size of G to be

$$\mathrm{ds}(G) = 1 + \mathrm{card}(V) + \mathrm{card}(\Sigma) + \sum_{(A,\beta)\in P} (|\beta| + 3).$$

Roughly speaking, this is the number of symbols needed to write down a description of G.

Separating words with grammars: unequal lengths

Theorem. Suppose $w, x \in \Sigma^*$ with $|w|, |x| \leq n$ and $|w| \neq |x|$. Let $k = \mathrm{card}(\Sigma)$. Then there exists a CFG G that separates w from x with description size $\mathrm{ds}(G) = O(k + \log\log n)$.

Proof. As in the case with automata, it suffices to show how to generate $(\Sigma^p)^*\Sigma^i$ where $|w| \not\equiv |x| \pmod{p}$ and $|w| \equiv i \pmod{p}$. We can do this using $O(k + \log_2 p)$ productions using the "binary method".

For example, suppose $p = 19$, $i = 6$. Then we can generate $((0+1)^{19})^*(0+1)^6$ as follows:

$$\begin{aligned}
S &\to CA_6 \\
C &\to \epsilon \mid A_{19}C \\
B &\to 0 \mid 1 \\
A_{19} &\to BA_{18} \\
A_{18} &\to A_9A_9 \\
A_9 &\to BA_8 \\
A_8 &\to A_4A_4 \\
A_4 &\to A_2A_2 \\
A_2 &\to BB \\
A_6 &\to A_3A_3 \\
A_3 &\to BA_2
\end{aligned}$$

Separating words with grammars: equal lengths

Theorem. Suppose $w, x \in \Sigma^*$ with $|w|, |x| \leq n$ and $|w| = |x|$. Let $k = \mathrm{card}(\Sigma)$. Then there exists a CFG G that separates w from x with description size $\mathrm{ds}(G) = O(k + \log n)$.

Proof.

- If $w \neq x$ and $|w| = |x|$ then there must exist a position j, $1 \leq j \leq n$ in which $a = w_j \neq x_j = b$ for $a, b \in \Sigma$.
- Then $w \in \Sigma^{j-1} a \Sigma^{n-j}$ and $x \notin \Sigma^{j-1} a \Sigma^{n-j}$.
- Using the binary technique as before, we get a grammar with description size $O(k + \log n)$. ∎

Lower bounds: separating collections

Let S be a finite set. We call a finite collection

$$\mathcal{U} = \{U_1, U_2, \ldots, U_j\}$$

of subsets of S a *separating collection* if for all $x, y \in S$ with $x \neq y$, there exists a set $C_{xy} \in \mathcal{U}$ such that

$$\mathrm{card}(C_{xy} \cap \{x, y\}) = 1.$$

Lower bounds: a useful lemma

Separating Lemma. Suppose S is a finite set of cardinality $m \geq 1$. If $\mathcal{U} = \{U_1, U_2, \ldots, U_j\}$ is a separating collection for S, then $\mathrm{card}(\mathcal{U}) \geq \lceil \log_2 m \rceil$. Furthermore, this bound is best possible.

Proof. For each $x \in S$ consider the set of indices of members of $\mathcal{U}$ to which x belongs, that is,

$$V_x = \{i : x \in U_i\}.$$

Then we claim that all the sets V_x are distinct; for if not we would have $V_x = V_y$ for some $y \neq x$, and then every set in $\mathcal{U}$ containing x would also contain y. Hence $\mathcal{U}$ is not a separating collection. It follows that $2^{\mathrm{card}(\mathcal{U})} \geq m$, and hence $\mathrm{card}(\mathcal{U}) \geq \log_2 m$. Since the cardinality of a set must be an integer, we obtain $\mathrm{card}(\mathcal{U}) \geq \lceil \log_2 m \rceil$.

We now show the bound is best possible. Without loss of generality, we may assume $S = \{0, 1, 2, \ldots, m-1\}$. Then define

$$U_i = \{r \in S \ : \text{the } i\text{'th least significant bit of the binary expansion of } r \text{ is } 1\}$$

for $0 \leq i < \lceil \log_2 m \rceil$, and set $\mathcal{U} = \{U_i \ : \ 0 \leq i < \lceil \log_2 m \rceil\}$. It works. ∎

Lower bounds

Theorem. Let $k = \mathrm{card}(\Sigma)$ be fixed. For all $n \geq 1$ there exists a pair of distinct words $w, x \in \Sigma^n$ requiring a context-free grammar of description size $\Omega(\frac{\log n}{\log \log n})$ to separate them.

Proof.

- Wlog $\Sigma = \{1, 2, \ldots, k\}$.
- Let $G = (V, \Sigma, P, S)$ be a CFG separating w from x.
- Wlog $V = \{A_1, A_2, \ldots, A_r\}$ and $A_1 = S$.
- encode G as a string s over the alphabet $V \cup \Sigma \cup \{\#\}$ as follows: if

  $$P = \{B_1 \to \beta_1, B_2 \to \beta_2, \ldots, B_t \to \beta_t\}$$

  then

  $$s = B_1 \# \beta_1 \# B_2 \# \beta_2 \# \cdots \# B_t \# \beta_t \#.$$

- Then

  $$|s| = \sum_{1 \leq i \leq t} (|\beta_i| + 3) \leq \mathrm{ds}(G),$$

  and each position in s can take on at most $|\Sigma| + |V| + 1 \leq \mathrm{ds}(G)$ different values.
- Suppose $\mathrm{ds}(G) \leq d$.

- There are at most d^d different strings encoding such a grammar, and hence at most d^d different grammars with description size $\leq d$.
- Hence there are at most d^d different languages generated by CFG's with description size $\leq d$.
- Let $\mathcal{U}$ be the collection of all these languages. Now apply the Separating Lemma.
- There are k^n distinct words of length n.
- If $\mathcal{U}$ is a separating collection for the set of words of length $\leq n$, then we have

$$d^d \geq \mathrm{card}(\mathcal{U}) \geq \log_2 k^n.$$

- It follows that

$$d \log d \geq \log n + \log \log_2 k.$$

- Hence for fixed k we have $d = \Omega(\frac{\log n}{\log \log n})$. ∎

Lower bound for unequal lengths

Theorem. Let $k = \mathrm{card}(\Sigma)$ be fixed. For all $n \geq 1$ there exists a pair of words w, x with $|w| \neq |x|$ and $|w|, |x| \leq n$ requiring a context-free grammar with description size $\Omega(\frac{\log\log n}{\log\log\log n})$ to separate them.

Proof.

- Suppose G is a context-free grammar separating w and x, where $|w| \neq |x|$.
- Wlog we may assume $k = 1$.
- There are $n + 1$ strings in 0^* of length $\leq n$.
- Applying the Separating Lemma as before, we find that $d^d \geq \log_2(n + 1)$
- Hence $d = \Omega(\frac{\log\log n}{\log\log\log n})$. ∎

The Separating Lemma Alone Can't Give Better Lower Bounds

The lower bound technique using the separating lemma cannot improve the lower bounds without a significant new idea. This is because there are $d^{\Omega(d)}$ distinct languages generable by context-free grammars with description size $\leq d$ — even over a unary alphabet.

Theorem. Any subset $S \subseteq \{\epsilon, 0, 0^2, \ldots, 0^{n \cdot 2^n - 1}\}$ can be generated using a grammar with description size $O(2^n)$. Hence $2^{n \cdot 2^n} = (2^n)^{2^n}$ different languages can be generated by grammars with description size $O(2^n)$.

Proof Sketch.

- The idea is to use a "four Russians" style approach.
- Take a subset

$$S \subseteq \{\epsilon, 0, 0^2, \ldots, 0^{n \cdot 2^n - 1}\}.$$

- Then write S as the disjoint union of 2^n pieces S_j, each containing the strings in S of length between jn and $(j+1)n$, for $0 \leq j < 2^n$.
- Let $S_j = W_j 0^{jn}$, so that

$$S_j \subseteq A = \{\epsilon, 0, 0^2, \ldots, 0^{n-1}\}.$$

- Now we need to generate all nonempty subsets of A and all strings 0^{jn}, $0 \leq j < 2^n$, with $O(2^n)$ description size.
- In both cases we can do this incrementally.
- Each new subset is produced by adding one new member to a subset produced by a previous variable.
- Each new 0^{jn} is produced by concatenating a variable that produces 0^j with one that produces $0^{j(n-1)}$.
- The total description size is $O(2^n)$. ∎

A matching lower bound for unequal lengths

We now prove a matching lower bound of $O(\log \log n)$ in the case of unequal lengths.

- We assume that the grammars in the section are in "binary normal form"; that is, that every production is of the form $A \to BC$, or $A \to B$, or $A \to a$, or $A \to \epsilon$.
- We can convert every grammar to one in binary normal form with only a linear increase in description size.
- Fix a grammar $G = (V, T, P, S)$. A word a^i is called a *pumping word* if there exist a variable A and integers $i_1, i_2 \geq 0$ such that $i = i_1 + i_2$ and $A \underset{G}{\overset{*}{\Longrightarrow}} a^{i_1} A a^{i_2}$.
- A *cutting operation* or *cut* can be performed on a parse tree T if there exist two occurrences of the same variable A at nodes N_1 and N_2, with N_2 a descendant of N_1.
- The cut is performed by replacing the subtree rooted at N_1 by that rooted at N_2, obtaining a new tree T'.
- If the yield of T is w, and the yield of T' is w', then we say w' is *obtained by cutting* from w.
- If $w = a^j$ and $w' = a^k$, then a^{j-k} is a pumping word.
- A *pasting operation* is the result of undoing a cut.

- We say w is $obtained\ by\ pasting$ from w'.
- We say a cut is $basic$ if there is no other possible cut, possibly involving other variables, within the subtree rooted at N_1.
- A pumping word is called $basic$ if it is cut out by some basic cut.
- If a cut is possible, then a basic cut is also possible.
- Assume $G = (V, T, P, S)$ is in binary normal form, with $s := \mathrm{card}(V)$ and $m := 2^s$.

Lemma 1. If a parse tree T has yield x with $|x| \geq m$, then it is possible to perform a cut in it.

Proof.

- Like the proof of the pumping lemma.
- Since $|x| \geq 2^s$, we have $|x| > 2^{s-1}$.
- Since every node of the parse tree has outdegree ≤ 2, there is a path from the root of T to a leaf of length $\geq s + 1$.
- Such a path has $\geq s + 2$ nodes, all of which but the last are labeled with variables.
- Hence some variable occurs twice on this path. ∎

Lemma 2. Every basic pumping word is of length $< 2m$.

Proof.

- Let a^i be a word with $i \geq 2m$, and consider an arbitrary cut operation which cuts out a^i.
- Consider the subtree T_1 rooted at node N_1 of the cut.
- The yield of T_1 is of length $\geq 2m$.
- Since G is in binary normal form, the node N_1 has at most two children.
- Thus N_1 has a child which is the root of a subtree T_2 with yield of length $\geq m$.
- By Lemma 1, it is possible to perform a cut in T_2.
- Hence the cut at N_1 is not basic.
- Since we considered an arbitrary cut, a^i is not cut out by any basic cut and hence a^i is not a basic pumping word. ∎

Lemma 3. Given a derivation $S \stackrel{*}{\Longrightarrow} w$, there is a word w' with $|w'| < 2ms$, obtained from w by a sequence of zero or more basic cuts, such that w' has a derivation in G using all the variables used in the derivation of w.

Proof.

- Let T be the parse tree corresponding to the given derivation $S \stackrel{*}{\Longrightarrow} w$.
- Consider reducing T to a tree T'' with yield w'' by a sequence of basic cuts $c_1, c_2, \ldots, c_j$, such that no further cuts are possible in T''.
- By Lemma 1 we have $|w''| < m$.
- Suppose that the basic cuts which removed the last occurrence of some variable are $c_{n_1}, \ldots, c_{n_k}$, in that order.
- Then we can perform pasting operations to T', reversing the effects of cuts $c_{n_k}, \ldots, c_{n_1}$, and obtaining a parse tree T' with yield w'.
- Then all the other pasting operations corresponding to the remaining c_i can be performed in any order to obtain a tree with yield w.
- Since w has been obtained from w' by pasting basic pumping words, w' can be obtained from w by basic cuts.

- Then, since $k \leq s-1$, we have

$$|w'| < |w''| + 2m(s-1) < m + 2m(s-1) < 2ms.$$

■

We now fix one such sequence of basic cuts, and define $r(w)$ to be the word w' obtained from w by this sequence of basic cuts.

Theorem. Let $G = (V, T, P, S)$ be a context-free grammar in binary normal form with $s = \mathrm{card}(V)$. Define $m = 2^s$, $m' = \mathrm{lcm}(1, 2, \ldots, 2m)$, and $M = 4m^2m'$. Then G does not separate a^{2M} from a^{3M}.

Proof. We show $a^{2M} \in L(G)$ iff $a^{3M} \in L(G)$.

$\Longrightarrow$:

- Suppose $w = a^{2M} \in L(G)$, and consider a parse tree T for a^{2M}.
- Since $|w| = 2M \geq m$, we can perform a pasting operation on T, which increases the length of the resulting yield by i, where a^i is a basic pumping word.
- By Lemma 2 we have $i < 2m$, and so $i \mid m'$.
- But $m' \mid M$, so we can perform this pasting operation $\frac{M}{i}$ times to get a derivation of a^{3M}.

$\Longleftarrow$:

- Consider the set $B = \{a^{b_1}, a^{b_2}, \ldots, a^{b_j}\}$ of all nonempty basic pumping words, where $b_1 \leq b_2 \leq \cdots \leq b_j$.
- Define $g := \gcd_{x \in B} |x|$.
- Now if t is an integer linear combination of the b_i, then $g \mid t$.
- By a well-known result if $u \geq (b_1/g - 1)(b_j/g - 1)$, then gu is a non-negative integer linear combination of the b_i.
- Since $b_i < 2m$ for $1 \leq i \leq j$, it follows that if $i \geq 4m^2$, then $a^i \in B^*$ iff $g \mid i$.
- Now suppose $w = a^{3M} \in L(G)$.
- Wlog assume that every variable in V is used in the derivation of w; for if not, we could replace V by V', where V' consists only of those variables appearing in the derivation of w.
- Consider the word $a^x = r(a^{3M})$, where r is the function defined after Lemma 3.
- Since a^x was obtained from a^{3M} by a sequence of basic cuts, we must have $a^{3M-x} \in B^*$.
- By Lemma 3, we have $x < 2ms < M$.
- Hence $3M - x > 2M - x > M \geq 4m^2$.

- Since $g \mid M$, it follows that $a^{2M-x} \in B^*$.
- By Lemma 3, the parse tree for a^x uses all variables of V, so we can perform any sequence of paste operations.
- Thus we can perform pasting operations that add $2M - x$ a's to a^x, obtaining a^{2M}, and so $a^{2M} \in L(G)$. ∎

Corollary. Let $k = \mathrm{card}(\Sigma)$ be fixed. For all $n \geq 1$ there exist words w, x with $|w| \neq |x|$ and $|w|, |x| \leq n$ requiring a context-free grammar with description size $\Omega(\log \log n)$ to separate them.

Proof.

- Let G be a context-free grammar with description size d.
- We may convert G to an equivalent grammar G' in binary normal form with description size $\leq \alpha d$ for some constant α.
- By the previous theorem, G' (and hence G) fails to separate a^{2M} from a^{3M}.
- But $m = 2^s \leq 2^{\alpha d}$.
- Also, $m' = \mathrm{lcm}(1, 2, \ldots, 2m) \leq e^{2.08m}$ by a well-known estimate.
- It follows that $3M \leq 12 \cdot 2^{2\alpha d} \cdot e^{2.08 \cdot 2^{\alpha d}}$, and so $d = \Omega(\log \log n)$, where $n = 3M$. ∎

Open Problems

1. Improve Robson's bound of $O(n^{2/5}(\log n)^{3/5})$ states for separating same-length words of length $\leq n$ with a DFA.
2. Find better bounds for separating with an NFA.
3. Close the gap between the upper bound of $O(\log n)$ and the lower bound of $\Omega\left(\frac{\log n}{\log\log n}\right)$ for separation by grammars.

For Further Reading

1. P. Goralčík and V. Koubek. On discerning words by automata. In L. Kott, editor, *Proc. 13th Int'l Conf. on Automata, Languages, and Programming (ICALP)*, Vol. 226 of *Lecture Notes in Computer Science*, pp. 116–122. Springer-Verlag, 1986.

2. J. Gruska. On the size of context-free grammars. *Kybernetika* 8 (1972), 213–218.

3. E. F. Moore. Gedanken-experiments on sequential machines. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, Vol. 34 of *Annals of Mathematics Studies*, pp. 129–153. Princeton University Press, Princeton, 1956.

4. J. M. Robson. Separating strings with small automata. *Inform. Process. Lett.* **30** (1989), 209–214.

5. J. M. Robson. Separating words with machines and groups. *RAIRO Inform. Théor. App.* **30** (1996), 81–86.

An electronic copy of these slides can be found at
`http://math.uwaterloo.ca/~shallit/talks.html`