Complexity of inclusion problems for unambiguous automata and context-free grammars

Lorenzo Clemente,University of Warsaw, Poland

May 1, 2019

1 Intro

Complement of UCFL languages need not be CFL [4].

The "CFG $\subseteq$ NFA" problem is easily shown to be $\mathsf{EXPTIME}$-c.. In fact, the following closely related problem is $\mathsf{EXPTIME}$-c.: Given a CFG G and a family of DFA's $A_1, \ldots, A_k$, decide whether $L(G) \cap \bigcap_{i=1}^k L(A_k) = \emptyset$ [3, 6]. We can reduce the problem above to the "CFG $\subseteq$ NFA" problem, by simply noticing that $L(G) \cap \bigcap_{i=1}^k L(A_k) = \emptyset$ holds iff $L(G) \subseteq \bigcup_{i=1}^k \Sigma^* \setminus L(A_k)$, and that, since the A_k's are deterministic, a polynomial size NFA can be built to recognise the language on the right.

Let $\Sigma_n = \{a_1, \ldots, a_n\}$ be an alphabet of size n and let $m \leq n$ a bound on the number of allowed letters. Then,

$$\mu_{\Sigma_n}(\Sigma_m^*) = \frac{1}{n - m + 1}. \tag{1}$$

Lemma 1 (Representation lemma)**.** *Let $n + 1 \in \mathbb{N}$ with $n \geq 2$ be a base, let $m \in \mathbb{N}$ s.t. $1 \leq m \leq n$, and let $\alpha \in \mathbb{R}$ with $0 \leq \alpha \leq \frac{1}{n-m+1}$ be written in reduced form as*

$$\alpha = \frac{p}{q}, \qquad \text{with } p, q \in \mathbb{N},\ p \leq q.$$

There exists a DFA A over alphabet $\Sigma_n = \{a_1, \ldots, a_n\}$ using only letters from $\Sigma_m \subseteq \Sigma_n$, and thus $L(A) \subseteq \Sigma_m^*$, s.t.

$$\mu_{\Sigma_n}(L(A)) = \alpha.$$

Moreover, if there exists $\ell \in \mathbb{N}$ s.t.

$$q \mid (n+1)^\ell, \tag{2}$$

then A can be taken of size polynomial in $\log q$, n, and ℓ.

Proof. If $\alpha = \frac{1}{n-m+1}$, then just take A to be the automaton recognising Σ_m^*, and we are done.

Otherwise, assume $\alpha < \frac{1}{n-m+1}$. Our aim is to write α as the following infinite geometric sum:

$$\alpha = \sum_{i=0}^{\infty} \frac{\alpha_i}{(n+1)^{i+1}}, \qquad \text{with } 0 \le \alpha_i \le m^i, \tag{3}$$

where α_i intuitively counts the number of words of length i in $L(A)$. Moreover, we would like the sequence $\alpha_0, \alpha_1, \ldots$ to be eventually periodic (with small prefix and period), in order to construct a (small) finite automaton A recognising a language $L(A)$ of measure α. Let $k \in \mathbb{N}$ be a length threshold. The measure of all words of length at most k is

$$\mu_{\Sigma_n}(\Sigma_m^{\le k}) = \sum_{i=0}^{k} \frac{m^i}{(n+1)^{i+1}} = \frac{1}{n-m+1}\left(1 - \left(\frac{m}{n+1}\right)^{k+1}\right), \tag{4}$$

Since the quantity in (4) goes to $\frac{1}{n-m+1} > \alpha$ for large k, fix in the following the unique $k \in \mathbb{N}$ s.t. $\alpha_0 = m^0, \alpha_1 = m^1, \ldots, \alpha_{k-1} = m^{k-1}$, and $0 \le \alpha_k < m^k$ s.t.

$$\mu_{\Sigma_n}(\Sigma_m^{\le k-1}) + \frac{\alpha_k}{(n+1)^{k+1}} \le \alpha < \mu_{\Sigma_n}(\Sigma_m^{\le k-1}) + \frac{\alpha_k + 1}{(n+1)^{k+1}}. \tag{5}$$

For complexity considerations to be made later, we note here that k satisfies $\alpha \le \mu_{\Sigma_n}(\Sigma_m^k)$, and thus $k \ge \frac{\log(1-(n-m+1)\alpha)}{\log m - \log(n+1)} - 1 = \frac{-\log(1-(n-m+1)\alpha)}{\log(n+1) - \log m}$. The minimal denominator is achieved by $m = n$, and the maximal numerator by $m = 1$. Replacing, it suffices to take $k \ge \frac{-\log(1-n\alpha)}{\log(n+1)-\log n}$. Since $-\log(1-n\alpha) = O(\log q)$ and $\log(n+1) - \log n = \log \frac{n+1}{n} = \log(1 + \frac{1}{n}) = O(\frac{1}{n})$, we obtain

$$k = O(n \log q). \tag{6}$$

Let

$$\beta = \alpha - \left(\mu_{\Sigma_n}(\Sigma_m^{\le k-1}) + \frac{\alpha_k}{(n+1)^{k+1}}\right), \tag{7}$$

and thus $0 \le \beta < \frac{1}{(n+1)^{k+1}}$. We can write β in base-$(n+1)$ as

$$\beta = \frac{1}{(n+1)^k} \sum_{j=1}^{\infty} \frac{\beta_j}{(n+1)^{j+1}}, \qquad \text{with } 0 \le \beta_j \le n. \tag{8}$$

Intuitively, we can interpret β_j as the number of words of length $k + j$ that our language contains. Since β is a rational number, the sequence $\beta_1, \beta_2, \ldots$ is ultimately periodic [2], i.e., there exists a threshold $j_1 \in \mathbb{N}$ and a period $l \in \mathbb{N}$ s.t. for every $j \ge j_1$

$$\beta_{j+l} = \beta_j. \tag{9}$$

Let $\gamma_1 = \beta_{j_1}, \gamma_2 = \beta_{j_1+1}, \ldots, \gamma_l = \beta_{j_1+l-1}$. Consequently,

$$\begin{aligned}\beta &= \frac{1}{(n+1)^k} \left(\sum_{j=1}^{j_1-1} \frac{\beta_j}{(n+1)^{j+1}} + \sum_{s=0}^{\infty} \left(\frac{\gamma_1}{(n+1)^{j_1+sl+1}} + \cdots + \frac{\gamma_l}{(n+1)^{j_1+sl+l}} \right) \right) \\ &= \frac{1}{(n+1)^k} \left(\sum_{j=1}^{j_1-1} \frac{\beta_j}{(n+1)^{j+1}} + \frac{1}{(n+1)^{j_1-1}} \sum_{s=0}^{\infty} \frac{\gamma}{(n+1)^{l(s+1)+1}} \right), \quad \text{where}\end{aligned} \tag{10}$$

$\gamma = \gamma_1(n+1)^{l-1} + \cdots + \gamma_l(n+1)^0$.

Intuitively, γ is the number of words of length $k + j_1 - 1 + (s+1)l$, for every $s \in \mathbb{N}$.

We now build a regular expression e recognising a language of measure α. For a given length k and cardinality $0 \leq h \leq m^k$, we build an expression $e_{h,k}$ recognising a language of measure $\frac{h}{(n+1)^{k+1}}$. Write h as a base-m number

$$m = \sum_{i=0}^{k} h_i \cdot m^i, \qquad \text{with } 0 \leq h_i < m. \tag{11}$$

Then, the following regular expression $e_{h,k}$ recognises precisely h words of length k and no other word: If $h = m^k$, take $e_{h,k} = \Sigma_m^k$, and otherwise

$$\begin{aligned} e_{h,k} =&(a_1 + \cdots + a_{h_0}) \cdot \Sigma_m^0 \cdot a_1^{k-1} + \\ &+(a_1 + \cdots + a_{h_1}) \cdot \Sigma_m^1 \cdot a_1^{k-2} + \cdots + \\ &+(a_1 + \cdots + a_{h_{k-1}}) \cdot \Sigma_m^{k-1} \cdot a_1^0. \end{aligned}$$

The size of $e_{h,k}$ is $O(k(m+k)) = O(k(n+k))$.

We represent $\alpha - \beta = \mu(\Sigma_m^{\leq k-1}) + \frac{\alpha_k}{(n+1)^{k+1}}$ as the measure of the language recognised by the regular expression, also of size $O(k(n+k))$,

$$f = \Sigma_m^{\leq k-1} + e_{\alpha_k,k}. \tag{12}$$

Similarly, by (10) we can represent β as

$$g = e_{\beta_1,k+1} + \cdots + e_{\beta_{j_1-1},k+j_1-1} + e_{\gamma,k+j_1-1+l} \cdot e_{1,l}^*. \tag{13}$$

Since $n \leq m^k$ for k sufficiently large and $\beta_j \leq n$ ($k \geq \frac{\log n}{\log m}$ suffices), there are enough words β_j of length $k+j$ over alphabet Σ_m, and thus $e_{\beta_j,k+j}$ is well-defined. Similarly, in order for $e_{\gamma,k+j_1-1+l}$ to be well-defined, we need $\gamma \leq m^{k+j_1-1+l}$, which is satisfied for k large enough since $\gamma = O(n^l)$. The expression g above is of size $O(k(n+k) + j_1^2(n+j_1) + l(n+l))$.

Finally, the sought expression of measure α is $e := f + g$, which is of the same asymptotic size as g above. Note that e is unambiguous, since the length of the string uniquely specifies how it is parsed in $L(e)$. (Perhaps one could even derive a small deterministic automaton...?) By (6), and assuming $j_1, l \geq n$, e is of size

$$O(n^2 \log q + j_1^3 + l^2). \tag{14}$$

For the second part of the claim, assume (2) for some ℓ. By (6) and the assumption above,

$$k = O(n\ell \log(n+1)). \tag{15}$$

Since β is defined as $\alpha - \frac{\cdots}{(n+1)^{k+1}}$, we can write $\beta = \frac{r}{s}$ with $r, s \in \mathbb{N}$ and $r \leq s$, where $s \mid (n+1)^{k+1}$. If we decompose the base $n+1$ in prime factors as

$$n + 1 = p_1^{z_1} p_2^{z_2} \cdots p_m^{z_m}, \qquad \text{with } z_1, \ldots, z_m \in \mathbb{N},$$

then s is of the form $s = p_1^{t_1} p_2^{t_2} \cdots p_m^{t_m}$ with $t_i \leq (k+1)z_i$. By [2, Theorem 136], β can be written in base-$(n+1)$ with a *finite* expansion of length $j_1 = \max\left\{\frac{t_1}{z_1}, \ldots, \frac{t_m}{z_m}\right\} = O(k+1)$ (therefore the period is $l = 0$), and thus by (15)

$$j_1 = O(n\ell \log(n+1)). \tag{16}$$

By applying (14) to this case, we obtain a regular expression e of size

$$O(n^3 \ell^3 \log^3(n+1)), \tag{17}$$

which is polynomial in $\log q = O(\ell \log n)$, ℓ, and n, as required. □

2 Closure properties of UCFL's

We notice that UCFL's are closed under union and intersection with regular languages, with quadratic complexity when the regular language is presented as a DFA. Closure under intersection with a regular language is clear by taking the product of a UPDA and a NFA.

For closure under union, the input is a UCFL L (presented by a UPDA A) and a regular language M (presented by a DFA B). We build a UPDA C recognising $L \setminus M$ and then take the product with B in order to build a PDA $D = C \times B$ recognising $(L \setminus M) \cup M$: Since the union is disjoint, D is unambiguous, as required.

3 Non-trivial lower bounds

The universality problem for UFA can be reduced to solving linear equations and checking that the solution has 1 in a given component. Consequently, this gives both a PTIME and a NC^2 upper bound. Any lower-bound?

4 Complexity of inclusion problems

The "NFA $\subseteq$ DCFG" problem is solved in PTIME by effectively complementing the DCFG, intersecting it with the NFA, and testing non-emptiness of the resulting CFG.

Using the same trick as above, the problem "NFA $\subseteq$ UCFG" reduces to "DFA $\subseteq$ UCFG". Moreover, the latter problem reduces to the universality problem "UCFG $= \Sigma^*$" as follows: Let $L = L(A)$ and $M = L(G)$ for a DFA A

$\subseteq$	DFA	UFA	NFA	DCFG	UCFG	CFG
DFA	PTIME	PTIME	PSPACE-c.	PTIME	UUCFL	undecid.
UFA	PTIME	PTIME [5]	PSPACE-c.	PTIME	UUCFL	undecid.
NFA	PTIME	PTIME 5	PSPACE-c.	PTIME	UUCFL	undecid.
DCFG	PTIME	$\leq$UUCFL	EXPTIME-c.	undecid.	undecid.	undecid.
UCFG	PTIME	$\leq$UUCFL	EXPTIME-c.	undecid.	undecid.	undecid.
CFG	PTIME	$\leq$UUCFL	EXPTIME-c.	undecid.	undecid.	undecid.

Figure 1: Complexity of inclusion problems for various classes of regular and context-free languages

and a UCFG G. Since $L \subseteq M$ is equivalent to $L \subseteq L \cap M$, let G' be a UCFG for $N := L \cap M$. Since $N \subseteq L$ holds by construction, $L \subseteq N$ is equivalent to $N \cup (\Sigma^* \setminus L) = \Sigma^*$. Consequently, let G'' be a UCFG recognising $N \cup (\Sigma^* \setminus L)$. We have that our original problem $L \subseteq M$ is equivalent to $L(G'') = \Sigma^*$, as required. We used the fact that UCFL's are effectively closed under union and intersection with regular languages, and moreover efficiently so when those regular languages are represented as DFA's.

Similarly, "CFG $\subseteq$ UFA" reduces to "DCFG $\subseteq$ UFA", which in turn reduces to UUCFL, as follows: Let L be a DCFL and M a regular language. We have

$$L \subseteq M \quad \text{iff} \quad (M \cap L) \cup (\Sigma^* \setminus L) = \Sigma^*.$$

Notice that $M \cap L$ can be recognised by a UCFG of polynomial size (since L is deterministic and M unambiguous), that $\Sigma^* \setminus L$ can be recognised by a DCFG of polynomial size (since L is deterministic), and since the last two languages are disjoint, their union is also UCFG.

We can also reduce the "CFG $\subseteq$ UFA" problem to

$$\text{DCFG} \subseteq \text{DCFG} \cap \text{UFA},$$

with the further property that the language on the right is included by construction in the language of the left. Let A be a PDA and B a UFA. As before, we lift the alphabet to transitions of A, and obtain A': A transition $\delta = p \xrightarrow{a,\mathsf{op}} q$ in A becomes $p \xrightarrow{\delta,\mathsf{op}} q$ in A', where op is push or pop. We build a UPDA $B' = A \times B$ having as control locations pairs (p, q) with p a control location of A and q a state of B, and transitions of the form $(p, q) \xrightarrow{\delta,\mathsf{op}} (p', q')$ whenever $\delta = p \xrightarrow{a,\mathsf{op}} p'$ is a transition of A and $q \xrightarrow{a} q'$ is a transition of B. We have $L(A) \subseteq L(B)$ iff $L(A') \subseteq L(B')$ and $L(B') \subseteq L(A')$ by construction. Therefore, in order to decide the latter question, it suffices to decide the measure comparison problem

$$\mu(L(A')) \leq \mu(L(B')),$$

where A' is a DCFL and B' is DCFL $\cap$ UFA. The measure on the left can be approximated in PTIME (with Etessami's technique for SCFG; check-me). It remains to:

1. Approximate the measure on the right in PTIME, by extending the method of [1] from a DFA to a UFA.

2. Show that only "few" bits are needed to actually decide the inequality above. This should use the fact that B' is not arbitrary, but of the form

$A \times B$. EDIT: Exponentially many bits are needed, even just to check whether the measure is < 1 (remove a word of exponential length).

5 NFA $\subseteq$ UFA in PTIME

We show that nonetheless, the question whether $L(\mathcal{A}) \subseteq L(\mathcal{B})$ with $\mathcal{A}$ and NFA and $\mathcal{B}$ UFA can be solved in PTIME. We enrich the alphabet from Σ to $\Sigma' = \Delta_{\mathcal{A}}$, by adding information on the transition taken by $\mathcal{A}$. Let $\mathcal{A}'$ be the same as $\mathcal{A}$, except that a transition $\delta = p \xrightarrow{a} q$ in $\mathcal{A}$ becomes a transition $p \xrightarrow{\delta} q$ in $\mathcal{A}'$. Notice that $\mathcal{A}'$ is now deterministic. Let $\mathcal{B}'$ be the same as $\mathcal{B}$, except that a transition $p \xrightarrow{a} q$ in $\mathcal{B}$ is expanded in $\mathcal{B}'$ to a set of transitions $p \xrightarrow{\delta} q$ for every $\delta \in \Delta_{\mathcal{A}}$ of the form $r \xrightarrow{a} s$. Notice that $\mathcal{B}'$ is still unambiguous. Clearly, $L(\mathcal{A}) \subseteq L(\mathcal{B})$ iff $L(\mathcal{A}') \subseteq L(\mathcal{B}')$: For the "only if" direction, if $w' = \delta_1 \cdots \delta_n \in L(\mathcal{A}')$ with $\delta_i = p_i \xrightarrow{a_i} q_i$, then $w = a_1 \cdots a_n \in L(\mathcal{A})$, thus $w \in L(\mathcal{B})$ by assumption, and thus $w' \in L(\mathcal{B}')$ by construction. For the "if" direction, if $w = a_1 \cdots a_n \in L(\mathcal{A})$, then there exists $w' = \delta_1 \cdots \delta_n \in L(\mathcal{A}')$ with $\delta_i = p_i \xrightarrow{a_i} q_i$, thus $w' \in L(\mathcal{B}')$ by assumption, which implies $w \in L(\mathcal{B})$ by construction. Since the inclusion problem for unambiguous automata can be solved in PTIME, our original problem is in PTIME as well.

6 DCFG $\subseteq$ UFA

While "DCFG $\subseteq$ UFA" reduces to UUCFL, which is in PSPACE, this needs not be optimal. Another more direct way could be the following. Let A be a DPDA and B an UFA. Then:

1. Compute in PTIME the measure $\mu(q)$ of every control state q of $L(B)$.

2. Construct a UPDA A' for $L(A) \cap L(B)$. Control locations of A' are of the form $\langle p, q \rangle$ with $p \in A$ and $q \in B$. Notice that by construction $L(\langle p, q \rangle) \subseteq L(q)$ and thus $\mu(\langle p, q \rangle) \leq \mu(q)$. Then, $\mu(\langle p, q \rangle) \geq \mu(q)$ iff $L(\langle p, q \rangle) = L(q)$ iff $L(p) \subseteq L(q)$.

3. Thus it suffices to approximate $\mu(\langle p, q \rangle)$ within $\log(\mu(q))$ bits of precision. The latter problem is on a UPDA obtained by taking the product of a DPDA and a UFA. This in general can be simpler than computing the measure of an arbitrary UPDA/UCFL.

7 Complexity of UUCFL

The UUCFL problem asks whether a given unambiguous context-free grammar recognises the universal language Σ^*.

7.1 SQRTSUM lower bound for the measure upper bound

We show that deciding whether $\mu(L) \geq x$ for a UCFL L and $x \in \mathbb{Q}$ is hard for SQRTSUM. The SQRTSUM problem is the following: Given $d_0, d_1, \ldots, d_n \in \mathbb{N}$, is it the case that the following holds:

$$\sum_{i=1}^{n} \sqrt{d_i} \geq d_0. \tag{18}$$

We construct a UCFG G over a n-ary alphabet $\Sigma = \{a_1, \dots, a_n\}$ and a constant $\alpha \in \mathbb{Q}$ s.t.

$$\mu(L(G)) \geq \alpha \quad \text{iff} \quad (18) \text{ holds.}$$

Let $d = \max_{i=1}^{n} d_i$ and

$$x_i = 1 - \frac{\sqrt{d_i}}{d} = 1 - \sqrt{\frac{d_i}{d^2}}. \tag{19}$$

Note that x_i is the least non-negative solution of the following quadratic equation in x:

$$x = \frac{1}{2}\left(1 - \frac{d_i}{d^2}\right) + \frac{1}{2}x^2. \tag{20}$$

We build a UCFG G_i s.t.

$$\mu(L(G_i)) = x_i. \tag{21}$$

Notice that, since x_i is an irrational number, $|\Sigma| \geq 2$ is necessary, otherwise G would recognise a regular language, and thus its measure would be rational. (However, in general it might still be the case that computing such a rational measure) We assume w.l.o.g. that

(NOT USED) All the d_i's are distinct. If not, say $d_1 = d_2$, then we can replace them by the single $4d_1$, since $\sqrt{d_1} + \sqrt{d_2} = 2\sqrt{d_1} = \sqrt{4d_1}$.

1. We can assume that the maximal d is a square of the form:

$$d = (n+1)^{2h}. \tag{22}$$

If not, add a new integer $d_{n'} = (n'+1)^{2h}$ for h large enough, where $n' = n+1$, and replace d_0 with $d_0 + \sqrt{d_{n'}} = d_0 + (n'+1)^h$.

We look for a grammar G_i with initial nonterminal X_i and a rule of the form:

$$X_i \leftarrow C_i \mid A_i \cdot X_i \cdot a_n \cdot X_i,$$

where $A_i, C_i \subseteq \Sigma_{n-1}^*$ are small finite languages. If we call x the measure $\mu(L(X_i))$, and a the measure of $A_i \cdot a_n$, and c the measure of C_i, under the assumption that G is unambiguous, we obtain (recall that $\mu(LM) = (n+1)\mu(L)\mu(M)$ over an alphabet of size n for an unambiguous product LM)

$$x = c + (n+1)^2 a x^2.$$

By comparing the equation above with (20), we derive

$$a = \frac{1}{2(n+1)^2} = \frac{\frac{(n+1)^{k-1}}{2}}{(n+1)^{k+1}}, \text{ for every } k \geq 1, \text{ and} \tag{23}$$

$$c = \frac{1}{2}\left(1 - \frac{d_i}{d^2}\right). \tag{24}$$

We assume w.l.o.g. that n is odd and, consequently the numerator $\frac{(n+1)^k}{2}$ above is an integer. We choose k large enough s.t. $\frac{(n+1)^{k-1}}{2} \leq (n-1)^{k-1}$, and let $A_i \subseteq \Sigma_{n-1}^{k-1}$ be a finite language containing precisely $\frac{(n+1)^{k-1}}{2}$ strings of length $k-1$.

Since $\mu_{\Sigma_n}(\Sigma_{n-1}^*) = \frac{1}{2}$ and $c < \frac{1}{2}$, by Lemma 1, there exists a DFA C_i recognising a language $L(C_i) \subseteq \Sigma_{n-1}^*$ of measure c. Moreover, since c can be put in the form $c = \frac{\frac{d}{2}(d^2-d_i)}{(n+1)^{6h}} = \frac{p}{q}$ with $p, q \in \mathbb{N}$ relatively prime and $q \mid (n+1)^{6h}$, C_i is of polynomial size.

We argue that G_i is unambiguous. We notice that any word produced by X_i has the property that it has the same number of blocks in A_i's as a_n's. By way of contradiction, suppose $w \in L(X_i)$ is a word with two different derivations. Necessarily w contains a_n, otherwise the first production is applied and $w \in C_i$ can be derived in only one way. Thus, the second production is applied and w can be put in the two forms $w = uva_nt = u'v'a_nt'$ with $u, u' \in A_i$, and $v, t, v', t' \in L(X_i)$. First, $u = u'$ since all strings in A_i have length $k-1$. Assume w.l.o.g. that $|v| < |v'|$, which implies that v' is of the form $v' = va_nz$ with $v = xy$, $x \in A_i$ and $y, z \in L(X_i)$. Since y is in $L(X_i)$, it has the same number of A_i's blocks as well as a_n's. Thus, $v = xy$ cannot be in $L(X_i)$ because it has one more $x \in A_i$, which is a contradiction.

We obtain $L(X_i) \subseteq (\Sigma_0 \cup \Sigma_1)^{\geq 2}$ and $\mu(L(X_i)) = x_i$. By constructing the unambiguous grammar G with initial nonterminal X and productions

$$X \leftarrow a_1 \cdot X_1 \mid \cdots \mid a_n \cdot X_n, \tag{25}$$

we have that $L(X) \subseteq \Sigma \cdot (\Sigma_0 \cup \Sigma_1)^{\geq 2}$ and

$$\begin{aligned}\mu(L(X)) &= \mu(L(a_1 \cdot X_1)) + \cdots + \mu(L(a_n \cdot X_n)) = \\ &= \frac{1}{n+1}(x_1 + \cdots + x_n) = \\ &= \frac{1}{n+1}\left(n - \frac{\sqrt{d_1} + \cdots + \sqrt{d_n}}{d}\right)\end{aligned}$$

and thus

$$\sum_{i=1}^{n} \sqrt{d_i} \geq d_0 \quad \text{iff} \quad \mu(L(X)) \leq \alpha := \frac{1}{n+1}\left(n - \frac{d_0}{d}\right). \tag{26}$$

Are there PTIME approximations for $\mu(L(G))$ within some additive error $\varepsilon > 0$?

7.2 SQRTSUM hardness for UUCFL

??? WORK IN PROGRESS ???

We construct a regular language $L' \subseteq \Sigma_2^*$ of measure $\mu(L') = 1 - \alpha$. Since $0 < \alpha < 1$, the same holds for $1 - \alpha$. We have

$$1 - \alpha = \frac{d + d_0}{(n+1)d} = \frac{(n+1)^{2h} + d_0}{(n+1)^{2h+1}} = \frac{1}{n+1} + \frac{d_0}{(n+1)^{2h+1}},$$

where $d_0 \leq n\sqrt{d} = n(n+1)^h \leq (n+1)^{h+1}$. We can write d_0 in base $n+1$ as $d_0 = \sum_{i=0}^{k} f_i (n+1)^i$, where $k := \lfloor \log_{n+1} d_0 \rfloor \leq h+1$ and, for every $0 \leq j \leq k$, $0 \leq f_i \leq n$. Consequently, we can write $1-\alpha$ as

$$1 - \alpha = \frac{1}{n+1} + \frac{f_0}{(n+1)^{2h+1}} + \frac{f_1}{(n+1)^{2h}} + \cdots + \frac{f_k}{(n+1)^{2h-k+1}}, \tag{27}$$

allowing us to interpret the quantity above as the measure of a regular language $L' \subseteq \Sigma_2^*$ containing the empty word ε, and containing f_i words of length $2h-i+1$ for every $0 \leq i \leq k$. The shortest such word is of length $\geq 2h-(h+1)+1 = h \geq 2$, and there are enough such words $|\Sigma_2^2| = \left(\frac{n-1}{4}\right)^2 \geq n$ for n sufficiently large. Consider the language over Σ recognised by the nonterminal S and a rule

$$S \leftarrow L' \mid X. \tag{28}$$

First L' and $L(X)$ are disjoint since $L' \subseteq \{\varepsilon\} \cup \Sigma_2^{\geq 2}$. Consequently,

$$\mu(S) = \mu(L') + \mu(L(X)) = \frac{d + d_0}{(n+1)d} + \frac{1}{n+1}\left(n - \frac{\sqrt{d_1} + \cdots + \sqrt{d_n}}{d}\right) =$$
$$= 1 + \frac{d_0 - (\sqrt{d_1} + \cdots + \sqrt{d_n})}{(n+1)d}.$$

7.3 Bounded ambiguity

Notice that the universality problem for the union of two DCFL is undecidable: the first DCFL recognises runs of a deterministic Minsky machine that has a mistake on the first counter, and the second DCFL does the same on the second counter, in such a way that their (possibly overlapping union) encodes runs with at least some mistake. Then their union is universal iff the Minsky machine is empty.

This implies that already universality of 2-ambiguous grammars is undecidable.

7.4 PSPACE upper bound

Rewrite the section below for the more specific UUCFL.

Solving CFG $\subseteq$ UFA. The same construction shows that the inclusion problem $L \subseteq M$ for a CFL L and a regular language M reduces to an inclusion problem $L' \subseteq M'$ where L' is a DCFL and M' is in the same regular class as M.

In this section we present an PSPACE algorithm for the inclusion problem $L \subseteq M$ for L a DCFL recognised by a given DCFG and M recognised by a UFA.

Let G be a CFG with m nonterminals $X_1, \ldots, X_m$, and for every nonterminal X_i and length $n \in \mathbb{N}$, let $T_n(X_i) = |L(X_i) \cap \Sigma^n|$ be the number of words of length n generated by X_i. The *generating function* $g_i : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \cap \{\infty\}$ of X_i is defined as

$$g_i(x) = \sum_{n=0}^{\infty} T_n(X_i) \cdot x^n.$$

Since $|T_n(X_i)| \leq \Sigma^n$, the series above converges to a finite value in $\mathbb{R}_{\geq 0}$ for every $x < |\Sigma|$. For two generating functions f, g, we write $f \leq g$ if, for every $x \in \mathbb{R}_{\geq 0}$,

$f(x) \leq g(x)$. Given two nonterminals X_i, X_j of G, the *generating function inequality problem* asks whether $g_i \leq g_j$ holds. This problem is undecidable in general, since the CFL universality problem $L(Y) = \Sigma^*$ is equivalent to $g_X \leq g_Y$ where $L(X) = \Sigma^*$.

Let $g : \mathbb{R}_{\geq 0} \to \mathbb{R}^m_{\geq 0}$ be defined as $g(x) = (g_1(x), \ldots, g_m(x))$. We assume that the grammar is in Chomsky normal form. Let A be the set of continuous functions in $\mathbb{R}_{\geq 0} \to \mathbb{R}^m_{\geq 0}$, and consider the mapping $F : A \to A$ defined as $F(f)(x) = (F_1(f)(x), \ldots, F_m(f)(x))$, where

$$F_i(f)(x) = 1_{X_i \to \varepsilon?} + \sum_{X_i \to a} x + \sum_{X_i \to X_j X_k} f_j(x) \cdot f_k(x) \tag{29}$$

TODO: check that F maps continuous functions to continuous functions.

Lemma 2. *If G is unambiguous, then g is the least fixpoint of F.*

Proof. The fact that g is a fixpoint $F(g) = g$ follows immediately from unambiguity. Notice that F is itself a monotonic and continuous function on A. Monotonicity is clear. If $g_1 \leq g_2 \leq \cdots$ is a non-decreasing sequence of continuous functions $g_n \in A$ with limit $g = \lim_n g_n$, then

$$\begin{aligned}
F_i(\lim_n g_n) &= 1_{X_i \to \varepsilon?} + \sum_{X_i \to a} x + \sum_{X_i \to X_j X_k} (\lim_n g_n)_j(x) \cdot (\lim_n g_n)_k(x) = \\
&= 1_{X_i \to \varepsilon?} + \sum_{X_i \to a} x + \sum_{X_i \to X_j X_k} (\lim_n g_{nj})(x) \cdot (\lim_n g_{nk})(x) = \\
&= 1_{X_i \to \varepsilon?} + \sum_{X_i \to a} x + \sum_{X_i \to X_j X_k} \lim_n g_{nj}(x) \cdot \lim_n g_{nk}(x) = \\
&= 1_{X_i \to \varepsilon?} + \sum_{X_i \to a} x + \sum_{X_i \to X_j X_k} \lim_n \left(g_{nj}(x) \cdot g_{nk}(x)\right) = \\
&= \lim_n \left(1_{X_i \to \varepsilon?} + \sum_{X_i \to a} x + \sum_{X_i \to X_j X_k} g_{nj}(x) \cdot g_{nk}(x) \right) = \\
&= \lim_n (F_i(g_n)).
\end{aligned}$$

Consider the non-decreasing sequence of continuous functions $g_1 \leq g_2 \leq \cdots$ defined as $g_1(x) = (0, \ldots, 0)$, and, for every $i \geq 1$, $g_{i+1} = F(g_i)$. Then 1) $g^* = \lim_n g_n$ is the least fixpoint of F, and 2) $g = g^*$. Regarding 1), g^* is a fixpoint of F since $F(g^*) = \lim_n F(g_n) = \lim_n g_{n+1} = g^*$, and it is clearly the least one since g_1 is the minimal element of A. Regarding 2), it follows directly from the following characterisation of the m-th approximant g_m:

Claim. *Let $T_{mn}(X_i)$ be the number of words of length n that can be derived from X_i by at most m rewriting steps. Then,*

$$g_{ni}(x) = \sum_{n=0}^{\infty} T_{mn}(X_i) \cdot x^n. \tag{30}$$

Clearly, $T_n = \lim_m T_{mn}$, and thus ... TODO □

Lemma 3. *The generating function inequality problem can be solved in* EXPTIME *for unambiguous grammars.*

Proof. Thanks to Lemma 2, the generating function g of an ambiguous grammar is the least fixpoint of F from (29). Let

$$p_i(x, \overline{y}) \equiv y_i - 1_{X_i \to \varepsilon?} - \sum_{X_i \to a} x - \sum_{X_i \to X_j X_k} y_j \cdot y_k, \tag{31}$$

$$\widehat{\varphi}(x, \overline{y}) \equiv \bigwedge_i p_i(x, \overline{y}) = 0, \text{ and} \tag{32}$$

$$\varphi(x, \overline{y}) \equiv \widehat{\varphi}(x, \overline{y}) \wedge \forall \overline{z} \cdot \widehat{\varphi}(x, \overline{z}) \implies \overline{y} \leq \overline{z}. \tag{33}$$

Consequently,

$$g(x) = \overline{y} \quad \text{iff} \quad \varphi(x, \overline{y}).$$

Thus, the inequality problem $g_i \leq g_j$ is equivalent to

$$\forall x, \overline{y} \cdot \varphi(x, \overline{y}) \implies y_i \leq y_j.$$

which is a formula of Tarski algebra of fixed alternation depth, and this fragment is solvable in EXPTIME. □

Let $\|x\| = \max_i x_i$ be the max-norm. A function $F : \mathbb{R}^m \to \mathbb{R}^m$ is *contractive* if there exists a constant $0 \leq \alpha < 1$ s.t. for every vectors $x, y \in \mathbb{R}^m$, $\|F(x) - F(y)\| \leq \alpha \cdot \|x - y\|$. By Banach's Fixpoint Theorem, a contractive function F has a unique fixpoint $x^* = F(x^*)$, which can moreover be found by iterating $x^* = \lim_n F^n(x_0)$ for any initial vector $x_0 \in \mathbb{R}^m$.

Lemma 4. *If the grammar G is linear, then F_x is contractive for x sufficiently small and thus it has a unique fixpoint.*

Proof. If G is a linear grammar, then by letting $y_i = f_i(x)$, we can write

$$F_i(y) = 1_{X_i \to \varepsilon?} + \sum_{X_i \to \alpha X_j \beta} x^{|\alpha\beta|} y_j. \tag{34}$$

Therefore,

$$F_i(y) - F_i(z) = \sum_{X_i \to \alpha X_j \beta} x^{|\alpha\beta|} y_j - \sum_{X_i \to \alpha X_j \beta} x^{|\alpha\beta|} z_j = \sum_{X_i \to \alpha X_j \beta} x^{|\alpha\beta|} (y_j - z_j),$$

and thus $\|F(y) - F(z)\| = \max_i \left(\sum_{X_i \to \alpha X_j \beta} x^{|\alpha\beta|} (y_j - z_j) \right) \leq \gamma \|y - z\|$, where $\gamma = \sum_{X_i \to \alpha X_j \beta} x^{|\alpha\beta|}$. Therefore, F is contractive $\gamma < 1$ provided that

$$x < k^{-l},$$

where k is the number of productions of G and $l = \max_{X_i \to \alpha X_j \beta} |\alpha\beta|$ is the maximum length of a r.h.s. of a production. □

Lemma 5. *The inequaliy problem between an UCFG gf f and an ULCFG gf g can be solved in PSPACE.*

Proof. By Lemma 4, generating functions of ULCFG are unique solutions of polynomial equations, and thus the problem is equivalent to

$$\exists x, \overline{y}, \overline{z} \cdot \varphi_f(x, \overline{y}) \wedge \varphi_g(x, \overline{z}) \to y_i \leq z_j.$$

which is a formula of the existential fragment of Tarski algebra, and thus solvable in PSPACE. □

Corollary 6. *The universality problem for unambiguous grammars is in PSPACE.*

Proof. Let $k = |\Sigma|$. The generating function of Σ^* is $g_{\Sigma^*}(x) = \sum_{n=0}^{\infty} k^n x^n = \frac{1}{1-kx}$, and thus L is universal iff $\forall x \cdot g_L(x) \geq \frac{1}{1-kx}$. This is the same as

$$\forall x, \overline{y} \cdot \bigwedge_i p_i(x, \overline{y}) = 0 \implies y_j \geq \frac{1}{1-kx},$$

which is a formula of the existential fragment of Tarski algebra, and thus solvable in PSPACE. □

TODO: extend to weighted automata and grammars?

References

[1] K. Etessami, A. Stewart, and M. Yannakakis. Stochastic context-free grammars, regular languages, and newton's method. In *In Proc. of ICALP'13*, ICALP'13, pages 199–211, Berlin, Heidelberg, 2013. Springer-Verlag.

[2] W. E. Hardy G.H. *An introduction to the theory of numbers.* OUP, 6ed. edition, 2008.

[3] A. Heußner, J. Leroux, A. Muscholl, and G. Sutre. Reachability analysis of communicating pushdown systems. *LMCS*, 8(3):1–20, September 2012.

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

[5] R. E. Stearns and H. B. Hunt. On the equivalence and containment problems for unambiguous regular expressions, grammars, and automata. In *Proceedings of the 22nd Annual Symposium on Foundations of Computer Science*, SFCS '81, pages 74–81, Washington, DC, USA, 1981. IEEE Computer Society.

[6] J. Swernofsky and M. Wehar. On the complexity of intersecting regular, context-free, and tree languages. In M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, editors, *Automata, Languages, and Programming*, volume 9135 of *Lecture Notes in Computer Science*, pages 414–426. Springer Berlin Heidelberg, 2015.