On the unary equivalence of stochastic languages and stochastic context-free languages.

Mahesh Viswanathan* V. S. P. Vijay Bhattiprolu†

December 10, 2013

Abstract

We show that every unary stochastic context-free grammar with polynomially bounded ambiguity, has an equivalent probabilistic automaton.

1. Introduction

Probabilistic Automata. A ***Probabilistic Automaton (PA)*** is an extension of a non-deterministic finite automaton wherein each transition has an associated probability. The probability of a given path in the automaton is given by the product of the corresponding probabilities of the transitions. The probability of acceptance of a string s in the automaton is given by the sum of the probabilities of every distinct path to a final state with the string s as yield. If no path to a finish state yielding s, its probability is defined to be zero. As such we may view a probabilistic automaton as a mapping from Σ^* to $[0, 1]$.

Stochastic Grammars. Similar in concept to probabilistic automata, a ***Stochastic Context-Free Grammar (SCFG)*** is an extension of a context-free grammar wherein each production rule has an associated probability. The probability of a given derivation tree of the grammar is given by the product of the corresponding probabilities of the rules applied. The probability of generating a string s in the grammar is given by the sum of the probabilities of every distinct derivation tree with the string s as frontier. If there is no such tree, then the probability of s is defined to be zero. Again, we may view a probabilistic automaton as a mapping from Σ^* to $[0, 1]$.

Ambiguity Function. A context-free grammar G is ambiguous if and only if there is a word that can be generated by G with at least two different derivation trees. The ambiguity function of G as defined by Wich [Wic99], is a function mapping every natural number n to the maximal number of derivation trees which a word of length at most n may have. A grammar is said to have ***exponential ambiguity*** if its ambiguity function is in $2^{\Theta(n)}$, and it is said to have ***polynomially bounded ambiguity*** if its ambiguity function is in $O(n^d)$ for some $d \in \mathbb{N}_0$. Wich [Wic99] showed that any grammar G has either exponential ambiguity or polynomially bounded ambiguity. It is clear that these results extend to SCFGs.

*Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; http://www.cs.uiuc.edu/~vmahesh/.

†Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; http://vspvijay.com.

Our Results. In this paper we show how unary stochastic context-free languages generated by some stochastic grammar of polynomial ambiguity, are stochastic languages. In fact we prove a stronger result, namely, that for every unary stochastic grammar G with polynomially bounded ambiguity, there is a probabilistic automaton M that realizes the same mapping from Σ^* to $[0,1]$ as G.

2. Preliminaries

2.1. Probabilistic Automata

Formally, a ***Probabilistic Automaton*** is a tuple $\mathrm{M} = (\mathrm{Q}, \Sigma, \mathrm{q}_0, \delta, \mathrm{F})$ where $(\mathrm{Q}, \Sigma, \mathrm{q}_0, \mathrm{F})$ are as in the standard NFA description and δ is a set of quadruples $\mathrm{Q} \times \Sigma^* \times \mathrm{Q} \times (0,1]$ where we allow words in transitions under the usual definition in NFAs. For any triple in $(\mathrm{q}, \mathrm{w}, \mathrm{q}') \in \mathrm{Q} \times \Sigma^* \times \mathrm{Q}$, that determines a transition, $\mathrm{P}(\mathrm{q}, \mathrm{w}, \mathrm{q}')$ denotes the probability of that transition. There is an additional constraint wherein, for a particular state q and word w the probabilities of all corresponding transitions add up to one. A path ψ in the automaton is a sequence of states and transitions starting at q_0, and represented by $\mathrm{q}_0 \xrightarrow{\mathrm{w}_1} \mathrm{q}_1 \cdots \mathrm{q}_{k-1} \xrightarrow{\mathrm{w}_k} \mathrm{q}_k$, where for all $1 \leq i \leq k, \mathrm{q}_i \in \mathrm{Q}$, and $(\mathrm{q}_{i-1}, \mathrm{w}, \mathrm{q}_i, \mathrm{p})$ is a transition, for some $\mathrm{p} \in (0,1]$. The probability of a path $\psi = \mathrm{q}_0 \xrightarrow{\mathrm{w}_1} \mathrm{q}_1 \cdots \mathrm{q}_{k-1} \xrightarrow{\mathrm{w}_k} \mathrm{q}_k$, denoted by $p(\psi)$, is given by,

$$p(\psi) = \prod_{i=0}^{k} \mathrm{P}(\mathrm{q}_{i-1}, \mathrm{w}, \mathrm{q}_i)$$

The yield of ψ, is given by, $y(\psi) = \mathrm{w}_1\mathrm{w}_2 \cdots \mathrm{w}_k$. The trace of a word $\mathrm{TR}(\mathrm{M}, \mathrm{w})$, is the set of every path to an accept state ψ, such that $y(\psi) = \mathrm{w}$. The probability of a word $\mathrm{w} \in \Sigma^*$, denoted by $\mathcal{P}_\mathrm{M}(\mathrm{w})$, is given by,

$$\mathcal{P}_\mathrm{M}(\mathrm{w}) = \sum_{\psi \in \mathrm{TR}(\mathrm{M},\mathrm{w})} p(\psi)\,.$$

A probabilistic automaton M paired with a cut-point η defines a language which is given by the set of strings accepted by M with probability greater than η. The set of languages recognized by probabilistic automata are called ***stochastic languages***.

2.2. Stochastic grammars

Formally, an ***SCFG*** is a tuple $(\mathrm{V}, \Sigma, \mathrm{R}, \mathrm{S})$ where $(\mathrm{V}, \Sigma, \mathrm{S})$ is as in the standard CFG description, while R is a set of triples of the form $\mathrm{V} \times (\mathrm{V} \cup \Sigma)^* \times (0,1]$, representing a production rule paired with a probability. There is an additional constraint, that the probabilities of the production rules of any fixed variable A, add up to one. An SCFG G paired with a cut-point η defines a language which is given by the set of strings generated by G with probability greater than η. The set of languages generated by SCFGs are called ***Stochastic Context-Free Languages (SCFL)***.

Let $\mathrm{G} = (\mathrm{V}, \Sigma, \mathrm{R}, \mathrm{P}, \mathrm{S})$ be an SCFG. We assume there are no non-generating cycles in G.

Definition 2.1. A ***single production variable*** is a variable $\mathrm{A} \neq \mathrm{S}$ such that there is exactly one production rule with A on the left hand side.

Observation 2.2. For any SCFG G_1, there is an equivalent SCFG G_2, with no single production variables. This is because for any single production variable A of G_1, we may simply replace every occurrence of A on the right hand sides of the rules by its own single production. Since there are no non-generating cycles, this process will always terminate.

By Observation 2.2, we may assume without loss of generality that G has no single production variables, and we do so throughtout this paper.

A sentence α, is any string in the set $(\mathrm{V} \cup \Sigma)^*$. For a sentence α, $\alpha_{/\mathrm{V}}$ is the projection of α onto V, and $\alpha_{/\Sigma}$ is the projection of α onto Σ.

We denote by Δ_{G}, the set of derivation trees of G. A variable node is any node represented by a variable and a terminal node is any node represented by a terminal. For any variable node n of a derivation tree τ, $\mathcal{V}(\mathsf{n})$ denotes the variable in τ representing n. The ***frontier*** of a tree τ is the set of leaf nodes of τ, and is denoted by $Fr(\tau)$. Two distinct trees τ_1, τ_2 are said to be ambiguous if $Fr(\tau_1) = Fr(\tau_2)$. A set of trees is said to be ambiguous if it contains some pair of ambiguous trees. The set of complete derivation trees, denoted by $\Delta_{\mathrm{G}}^{\Sigma}$, is the set of derivation trees that have only terminals in the frontier. For a variable A, the set of derivation trees that have A as the root and as the only variable in the frontier, is the set of A-Pumping trees and is denoted by $\Delta_{\mathrm{G}}^{p}(\mathrm{A})$. The set of Pumping trees denoted by Δ_{G}^{p}, is given by, $\Delta_{\mathrm{G}}^{p} = \bigcup_{\mathrm{A} \in \mathrm{V}} \Delta_{\mathrm{G}}^{p}(\mathrm{A})$.

A derivation step under the usual definition, for the sentences α_1, α_2, is denoted by $\alpha_1 \overset{\pi}{\Rightarrow} \alpha_2$, where $\pi \in \mathrm{R}$. A derivation sequence is a sequence of derivation steps beginning at S, for the derived sentence α, it is denoted by $\mathrm{S} \overset{*}{\Rightarrow} \alpha$. The yield of a derivation sequence $y\left(\mathrm{S} \overset{*}{\Rightarrow} \alpha\right)$, is given by $\alpha_{/\Sigma}$.

Definition 2.3. Let $\alpha, \alpha_1, \alpha_2 \in (\mathrm{V} \cup \Sigma)^*$, $\mathrm{A} \in \mathrm{V}$ and $(\pi = (\mathrm{A}, \alpha, \mathrm{p}) \in \mathrm{R}$ for some $\mathrm{p} \in (0, 1]$. The step $\alpha_1 \mathrm{A} \alpha_2 \overset{\pi}{\Rightarrow} \alpha_1 \alpha \alpha_2$ is a ***leftmost derivation step*** if $\alpha_1 \in \Sigma^*$, and it is denoted by $\alpha_1 \mathrm{A} \alpha_2 \overset{\pi}{\Rightarrow}_{lm} \alpha_1 \alpha \alpha_2$.

A leftmost derivation sequence, is a sequence of leftmost derivation steps beginning at S, and for a derived sentence α, it is denoted by $\mathrm{S} \overset{*}{\Rightarrow}_{lm} \alpha$. The parse PARSE(G, w), of a word w, is the set of every leftmost derivation sequence Ψ, such that $y(\Psi) = \mathrm{w}$. It is well known that every leftmost sequence corresponds to a single derivation tree and vice-versa. Thus we may define the ambiguity of a word w, as $|\mathrm{PARSE}(\mathrm{G}, \mathrm{w})|$. The ambiguity function of G denoted by μ, is a function mapping every natural number n, to the maximal ambiguity which a word of length at most n may have.

Consider any tree τ, and any corresponding derivation sequence Ψ of τ. Every sentence in the sequence Ψ, corresponds to a set of nodes in τ, which we term as a cut $\mathcal{C}$, of τ. It is useful to observe that no node in $\mathcal{C}$ is an ancestor of another node. Further, every node that is neither in $\mathcal{C}$, nor is an ancestor of any node in $\mathcal{C}$, must have some node in $\mathcal{C}$ as an ancestor. For a set of trees T and a variable $\mathrm{A} \in \mathrm{V}$, we denote by $\sup_{\Pi}(\mathrm{A}, T)$, the maximum number of occurrences of A, in any cut of any tree τ in T. Observe that any derivation sequence corresponding to a tree τ in T can have at most $\sup_{\Pi}(\mathrm{A}, T)$ occurrences of the variable A in any sentence.

The probability of a derivation tree is the product of the probabilities of every production rule applied in the tree. Equivalently, the probability of a derivation sequence Ψ, denoted by $p(\Psi)$, is the product of the probabilities of every production in the sequence. The probability of a word $\mathrm{w} \in \Sigma^*$, denoted by $\mathcal{P}_{\mathrm{G}}(\mathrm{w})$, is given by,

$$\mathcal{P}_{\mathrm{G}}(\mathrm{w}) = \sum_{\Psi \in \mathrm{PARSE}(\mathrm{G}, \mathrm{w})} p(\Psi)\,.$$

A probabilistic automaton M, and SCFG G are said to be equivalent if and only if $\mathcal{P}_{\mathrm{M}} = \mathcal{P}_{\mathrm{G}}$.

The following was shown by Wich [Wic99], which clearly extends to SCFGs:

Theorem 2.4. *Let* G *be a context-free grammar. Then* G *has polynomially bounded ambiguity if and only if* Δ_{G}^{p} *is unambiguous.*

3. Bounding the Parikh Supremum of a Variable

Let $G = (V, \{a\}, R, S)$ be a unary SCFG. We show that if G has polynomially bounded ambiguity, then every variable $A \in V$, has a bounded parikh supremum. Let m represent the maximum number of variables on the right hand side of any rule of G, and let $h = |V|$.

3.1. Extracting pumping trees

We show that there exists, for any complete derivation tree $\tau \in \Delta_G^\Sigma$, a derivation tree τ' of bounded size, such that τ can be obtained by replacing every variable node n in τ' with some $\mathcal{V}(\mathsf{n})$-pumping tree.

3.1.1. The algorithm ExtractPumpTree

ExtractPumpTree(τ) :
(A) Let n be the root of τ and let $A = \mathcal{V}(\mathsf{n})$.
(B) Find any node n' in τ, such that $\mathcal{V}(\mathsf{n}') = A$ and there is no occurrence of A having n' as an ancestor (Note that $\mathsf{n}' = \mathsf{n}$ is valid).
(C) Let τ' be the subtree of τ rooted at n'. Observe that there can be no occurrence of the variable A in any non-root node of τ'.
(D) Clearly, n and n' determine a A-pumping tree and replacing this pumping tree with the variable A, gives us τ'.
(E) In τ', for every child n_1 of n', replace the subtree τ_1 rooted at n_1, with the tree ExtractPumpTree(τ_1). Let this new tree be τ_2.
(F) Output τ_2.

3.1.2. Correctness

Lemma 3.1. *There exists a constant c, such that for any complete derivation tree $\tau \in \Delta_G^\Sigma$, there is a derivation tree τ' of at most c nodes, such that τ can be obtained by replacing every variable node n in τ' with some $\mathcal{V}(\mathsf{n})$-pumping tree.*

Proof: Let $\tau \in \Delta_G^\Sigma$ be any complete derivation tree. We claim that, $\tau' = \text{ExtractPumpTree}(\tau)$ is the desired tree for $c = O(m^h)$. Indeed, τ can be obtained by replacing every variable node n in τ' with some $\mathcal{V}(\mathsf{n})$-pumping tree. This is due to the easily verifiable property that every variable node in τ' is the result of exactly one replacement of a A-pumping tree by A, for some variable $A \in V$ (This could also be a trivial replacement).

We are only left to prove the claim on the size of τ'. To this end, let n' be the root of τ'. Observe that the only occurrence of the variable $\mathcal{V}(\mathsf{n}')$ in τ', is at n'. A stronger property holds, i.e. for any node n_1 in τ', the only occurrence of the variable $\mathcal{V}(\mathsf{n}_1)$ in the subtree rooted at n_1, is at n_1. Thus any path from the root to a leaf in τ' can have at most one occurrence of each variable, implying that τ' has height at most h, and consequently, $O(m^h)$ nodes. ∎

3.2. Bounding Variables in a Cut

Lemma 3.2. *Let A be a variable such that $\Delta_G^p(A)$ is unambiguous. Then for every variable $A_1 \in V$, $\sup_\Pi(A_1, \Delta_G^p(A)) \leq 2$.*

Proof: Consider any variable $\mathrm{A}_1 \in \mathrm{V}$ and any A-pumping tree $\tau \in \Delta^p_{\mathrm{G}}(\mathrm{A})$. Let $\mathcal{C}$ be any cut of τ. Since τ is an A-pumping tree it must contain A in its frontier. Then there must be some node n in $\mathcal{C}$ such that the subtree of τ rooted at n contains A in its frontier. Let $\mathrm{A}_2 = \mathcal{V}(\mathsf{n})$. Since A_1 is not a single production variable, there exist derivation trees τ_1, τ_2 rooted at A_1, such that $Fr(\tau_1), Fr(\tau_2) \in \Sigma^*$ and $\tau'_1 \neq \tau'_2$. Assume for purpose of contradiction, that there are three nodes $\mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3$ in $\mathcal{C}$ such that $\mathcal{V}(\mathsf{n}_1) = \mathcal{V}(\mathsf{n}_2) = \mathcal{V}(\mathsf{n}_3) = \mathrm{A}_1$. We have two cases:

If $\mathsf{n} \in \{\mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3\}$: Without loss of generality we may assume that $\mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3$ are arranged from left to right in τ. Let τ_1 be the tree obtained by replacing the subtree at n_1 by τ'_1, the subtree at n_2 by τ'_2 and the subtree at n_3 by τ'. Let τ_2 be the tree obtained by replacing the subtree at n_1 by τ'_2, the subtree at n_2 by τ'_1 and the subtree at n_3 by τ'. Now that τ_1 and τ_2 are ambiguous derivation trees that can be generated by G - a contradiction.

If $\mathsf{n} \notin \{\mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3\}$: Observe that at least two nodes of $\{\mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3\}$ must lie to one side of n in τ. Without loss of generality we may assume that n_1 and n_2 are those nodes. Let τ_1 be the tree obtained by replacing the subtree at n_1 by τ'_1, and the subtree at n_2 by τ'_2. Let τ_1 be the tree obtained by replacing the subtree at n_1 by τ'_2, and the subtree at n_2 by τ'_1. Now τ_1 and τ_2 are ambiguous derivation trees that can be generated by G - a contradiction.

Thus no variable can occur more than twice in any cut of τ, and the claim follows. ∎

Lemma 3.3. *If* G *has polynomially bounded ambiguity, then there is a constant c, such that for any variable* $\mathrm{A} \in \mathrm{V}$, $\sup_{\Pi}(\mathrm{A}, \Delta^p_{\mathrm{G}}) \leq c$.

Proof: Consider any cut $\mathcal{C}$, of any complete derivation tree $\tau \in \Delta^{\Sigma}_{\mathrm{G}}$. Let G have polynomially bounded ambiguity. Then, by Theorem 2.4, Δ^p_{G} is unambiguous. Observe that by Lemma 3.1, there exists a set of $O\big(\mathrm{m}^{\mathrm{h}}\big)$ pumping trees embedded in τ, such that any node n of τ, is contained in exactly one such pumping tree. This implies that $\mathcal{C}$ is the union of $O\big(\mathrm{m}^{\mathrm{h}}\big)$ cuts of pumping trees. Then by Lemma 3.2, no variable $\mathrm{A} \in \mathrm{V}$, can occur more than $O\big(\mathrm{m}^{\mathrm{h}}\big)$ times in $\mathcal{C}$, and the claim follows. ∎

4. The Result

Let $\mathrm{G} = (\mathrm{V}, \{\mathrm{a}\}, \mathrm{R}, \mathrm{S})$ be a unary SCFG with polynomially bounded ambiguity. We show how to construct a probabilistic automaton M, such that $\mathcal{P}_{\mathrm{G}} = \mathcal{P}_{\mathrm{M}}$. To do this, we employ an approach similar to that of Esperaza et. al. [EGKL10], wherein the bound on the number of variables in any cut, is exploited.

4.1. Construction

Let $\mathrm{G} = (\mathrm{V}, \{\mathrm{a}\}, \mathrm{R}, \mathrm{S})$ be a unary SCFG with polynomially bounded ambiguity. Consider the automaton $\mathrm{M}_k = (\mathrm{Q}_k, \{\mathrm{a}\}, \mathrm{q}_0, \delta_k, \mathrm{F})$ where,

$$\mathrm{Q}_k = \{\alpha \in \mathrm{V}^* \mid \textit{no variable } \mathrm{A} \in \mathrm{V} \textit{ occurs more than } k \textit{ times in } \alpha\}$$

$$\mathrm{q}_0 = \mathrm{S}$$

$$T(\mathrm{A}, \alpha, \mathrm{w}) = \big\{\mathrm{p}_1 \;\big|\; (\mathrm{A}, \alpha_1, \mathrm{p}_1) \in \mathrm{R},\ \alpha_{1/\mathrm{V}} = \alpha,\ \alpha_{1/\Sigma} = \mathrm{w}\big\} \quad \textit{for any } \alpha \in \mathrm{V}^*,\ \mathrm{w} \in \mathrm{a}^*$$

$$\delta_k = \left\{ (\mathrm{A}\alpha, \mathrm{w}, \mathrm{p}, \alpha'\alpha) \;\middle|\; \mathrm{A}\alpha, \alpha'\alpha \in \mathrm{Q}_k,\ \mathrm{w} \in \{\mathrm{a}, \epsilon\}^{\mathrm{t}},\ \ \mathrm{p} = \sum_{\mathrm{p}_1 \in T(\mathrm{A}, \alpha', \mathrm{w})} \mathrm{p}_1 \right\}$$

$$\mathrm{F} = \{\epsilon\}$$

Let c be as in Lemma 3.3. We claim that M_c is the desired automaton.

4.2. Correctness

For any leftmost derivation sequence $\Psi = \left(\mathrm{S} \overset{\pi_1}{\Rightarrow}_{lm} \alpha_1 \cdots\cdots \alpha_{k-1} \overset{\pi_k}{\Rightarrow}_{lm} \alpha_k\right)$, where π_i represents the production rule $(\mathrm{A}_i, \alpha'_i)$, let,

$$f(\Psi) = \left(\mathrm{S}_{/\mathrm{V}} \xrightarrow{\alpha'_{1/\Sigma}} \alpha_{1/\mathrm{V}} \cdots\cdots \alpha_{k-1/\mathrm{V}} \xrightarrow{\alpha'_{k/\Sigma}} \alpha_{k/\mathrm{V}}\right)$$

Lemma 4.1. *For any leftmost derivation sequence Ψ, $f(\Psi)$ is a valid path in M_c, and $y(f(\Psi)) = y(\Psi)$.*

Proof: Any derivation sequence of length zero, is simply the sentence S. Then, $f(\mathrm{S}) = \mathrm{S}$ which by construction, is a valid trivial path in M_c. Additionally, $y(\mathrm{S}) = y(f(\mathrm{S})) = \epsilon$. Given this base case, assume inductively that any derivation sequence Ψ of length less than i, is such that $f(\Psi)$ is a valid path in M_c and $y(f(\Psi)) = y(\Psi)$.

Consider any leftmost derivation sequence $\Psi = \left(\mathrm{S} \overset{\pi_1}{\Rightarrow}_{lm} \alpha_1 \cdots\cdots \alpha_{i-1} \overset{\pi_i}{\Rightarrow}_{lm} \alpha_i\right)$ of length i, where π_j represents the production rule $\left(\mathrm{A}_j, \alpha'_j, \mathrm{p}_j\right)$. Note that for A_j must be the leftmost variable of α_j. Let Ψ' be the leftmost derivation sequence representing the first $i-1$ steps of Ψ. By the inductive hypothesis, $f(\Psi')$ is a valid path in M_c and $y(f(\Psi')) = y(\Psi')$. Now, $y(f(\Psi)) = y(f(\Psi'))\, \alpha'_{i/\Sigma} = y(\Psi')\, \alpha'_{i/\Sigma} = y(\Psi)$.

Thus, we are only left to show that $\alpha_{k-1/\mathrm{V}} \xrightarrow{\alpha'_{k/\Sigma}} \alpha_{k/\mathrm{V}}$ is a valid transition. To this end, observe that by Lemma 3.3, $\alpha_{i/\mathrm{V}} \in \mathrm{Q}_c$. Further, since $\mathrm{p}_i > 0, (\mathrm{A}_i, \alpha'_i, \mathrm{p}_i) \in \mathrm{R}$, and A_i is the leftmost variable of α_i, by construction, there is some $\mathrm{p} \geq \mathrm{p}_j$, such that $\left(\alpha_{i-1/\mathrm{V}}, \alpha'_{i/\Sigma}, \mathrm{p}, \alpha_{i/\mathrm{V}}\right) \in \delta_k$. The claim follows. ∎

For any path ψ of M_c, let $f^{-1}(\psi) = \{\Psi \mid f(\Psi) = \psi\}$. For any $\mathrm{A} \in \mathrm{V}, \alpha \in \mathrm{V}^*$ and $\mathrm{w} \in \mathrm{a}^*$, let $Q(\mathrm{A}, \alpha, \mathrm{w}) = \left\{\pi \;\middle|\; \pi = (\mathrm{A}, \alpha_1, \mathrm{p}_1) \in \mathrm{R},\, \alpha_{1/\mathrm{V}} = \alpha,\, \alpha_{1/\Sigma} = \mathrm{w}\right\}$

Lemma 4.2. *Let ψ be any path in M_c. Then* $\sum_{\Psi \in f^{-1}(\psi)} p(\Psi) = p(\psi)$.

Proof: Consider the path in M_c of length zero. It is the trivial path S. Clearly, $f^{-1}(\mathrm{S}) = \{\mathrm{S}\}$, $p(\mathrm{S}) = 1$, and the sum of the derivation sequence probabilities from the set $\{\mathrm{S}\}$, is one. Given this base case, assume inductively that for any path ψ of length less than i, $\sum_{\Psi \in f^{-1}(\psi)} p(\Psi) = p(\psi)$.

Let $\psi = \left(\mathrm{S} \xrightarrow{\mathrm{w}_1} \mathrm{A}_1\alpha_1 \cdots\cdots \mathrm{A}_{i-1}\alpha_{i-1} \xrightarrow{\mathrm{w}_i} \alpha'_i\alpha_{i-1}\right)$ be any path in M_c of length i, where for every j, $\mathrm{A}_j \in \mathrm{V}$. Let ψ' be the path representing the first $i-1$ steps of ψ. From the inductive hypothesis, we have $\sum_{\Psi \in f^{-1}(\psi')} p(\Psi) = p(\psi')$. Let $Z = T(\mathrm{A}_{i-1}, \alpha'_i, \mathrm{w}_i)$. We then have,

$$\begin{aligned}
p(\psi) &= p(\psi') \times \sum_{\mathrm{p} \in Z} \mathrm{p} \\
&= \sum_{\mathrm{p} \in Z} \mathrm{p} \times p(\psi) \\
&= \sum_{\mathrm{p} \in Z} \sum_{\Psi \in f^{-1}(\psi')} p(\Psi) \times \mathrm{p} \\
&= \sum_{\Psi \in f^{-1}(\psi')} \sum_{\mathrm{p} \in Z} p(\Psi) \times \mathrm{p}
\end{aligned}$$

Observe that by construction, $f^{-1}(\psi)$ is the set of leftmost derivation sequences, that can be obtained exactly by appending every production in $Q(\mathrm{A}_{i-1}, \alpha'_i, \mathrm{w}_i)$ to every sequence in $f^{-1}(\psi')$. Now because Z is simply the set of probabilities of the productions in $Q(\mathrm{A}_{i-1}, \alpha'_i, \mathrm{w}_i)$, and from the equality above, we get,

$$\sum_{\Psi \in f^{-1}(\psi)} p(\Psi) = \sum_{\Psi \in f^{-1}(\psi')} \sum_{\mathrm{p} \in Z} p(\Psi) \times \mathrm{p} = p(\psi)$$

■

Lemma 4.3. *Consider any word* $\mathrm{w} \in \mathrm{a}^*$. $\mathcal{P}_{\mathrm{G}}(\mathrm{w}) = \mathcal{P}_{\mathrm{M}_c}(\mathrm{w})$.

Proof: Observe by construction, that a leftmost derivation sequence Ψ is a generating sequence in G if and only if $f(\Psi)$ is a path to the accept state in M_c. Thus if w has no paths to the accept state in M_c, then $\mathcal{P}_{\mathrm{G}}(\mathrm{w}) = \mathcal{P}_{\mathrm{M}_c}(\mathrm{w}) = 0$.

If w has some path to an accept state ψ in M_c, then $p(\psi) > 0$, and consequently by Lemma 4.2 and Lemma 4.1, there must be some leftmost derivation sequence Ψ in G, generating w, such that $f(\Psi) = \psi$. Further by Lemma 4.1, if Ψ is any leftmost derivation sequence generating w, then there is a unique path to an accept state ψ, with yield w, such that $f(\Psi) = \psi$. Now clearly we have,

$$\begin{aligned}
\mathrm{PARSE}(\mathrm{G}, \mathrm{w}) &= \bigcup_{\psi \in \mathrm{TR}(\mathrm{M}_c, \mathrm{w})} f^{-1}(\psi) \\
\Rightarrow \mathcal{P}_{\mathrm{G}}(\mathrm{w}) &= \sum_{\Psi \in \mathrm{PARSE}(\mathrm{G}, \mathrm{w})} p(\Psi) \\
&= \sum_{\psi \in \mathrm{TR}(\mathrm{M}_c, \mathrm{w})} \sum_{\Psi \in f^{-1}(\psi)} p(\Psi) \\
&= \sum_{\psi \in \mathrm{TR}(\mathrm{M}_c, \mathrm{w})} p(\psi) && \text{(by Lemma 4.2)} \\
&= \mathcal{P}_{\mathrm{M}_c}(\mathrm{w})
\end{aligned}$$

■

Theorem 4.4. *Let* G *be a unary stochastic context-free grammar with polynomially bounded ambiguity. Then, there exists a probabilistic automaton* M *equivalent to* G.

Proof: By Lemma 4.3, the desired automaton is M_c from Section 4.1. ■

References

[EGKL10] Javier Esparza, Pierre Ganty, Stefan Kiefer, and Michael Luttenberger. Parikh's theorem: A simple and direct construction. *CoRR*, abs/1006.3825, 2010.

[Wic99] Klaus Wich. Exponential ambiguity of context-free grammars. In Grzegorz Rozenberg and Wolfgang Thomas, editors, *Developments in Language Theory*, pages 125–138. World Scientific, 1999.