JACOBS UNIVERSITY

Comparison of the Expressive Powers of Weighted Grammars and OOMs

Josip Djolonga

Computer Science
Jacobs University Bremen
College Ring 7
28759 Bremen
Germany

Type: Guided Research Final Report
Date: May 17, 2011
Supervisor: Prof. Dr. H. Jaeger

Executive Summary

In many problems in computer science we have to come up with models that assign probabilities to finite strings composed of symbols from some finite alphabet. Moreover, we would like to devise mechanisms that will allow us to empirically learn the parameters of the model from some available corpus of data. For example, in speech processing we need a probabilistic model of the underlying language, and in computational biology we are interested in the probabilities of protein and DNA sequences which can be also represented as strings of letters. Some of the most widely used such models that have been applied to many areas of science are Hidden Markov Models (*HMM*). A natural generalization of *HMM*s that have been also extensively researched are Observable Operator Models (*OOM*). Similarly to *HMM*s, they assign probabilities by processing the input string from left to right. However, there is a crucial difference between these two formalisms, namely *OOM*s relax some of the restrictions *HMM*s pose and as a result have gained several benefits. They have been shown to be properly more powerful and moreover, unlike *HMM*s, there exists a learning algorithm that can efficiently converge to the best model. Another widely used model, that gives us a hierarchical multi-scale insight of the language being modelled are Weighted Context Free Grammars (*WCFG*). For each word that belongs to the language, the grammar has a set of hierarchically structured derivation trees from which the weight of the word can be inferred. The difference between *OOM*s and *WCFG*s is not only in the way they assign the weights, but also in the power of the methods to learn their parameters from data. As already mentioned, *OOM*s have a very efficient algorithm that can find the best parameters, while the algorithms used to infer the grammatical models do not have such properties. Given that these two models provide completely different views on the modelled language and have learning algorithms that possess different characteristics, we are interested in the relationship between their expressive power. In this thesis report, I first develop the mathematical framework needed to precisely state the question. Then, two theorems are proven that decide the expressivity question, namely, there is no proper inclusion between the two models.

CONTENTS

1 DESCRIPTION

Observable Operator Models (*OOM*) are a formalism of modelling symbolic, stationary stochastic time series that have been invented as the natural extension of the well known Hidden Markov Models (*HMM*). They have been extensively researched in the past which resulted in a rich theory based on linear algebra, and the *Efficiency Sharpening (ES)* (Jaeger et al., 2005) learning algorithm that has been proven to efficiently converge to the best parameters. These two models keep an internal state that is iteratively updated as the input string is processed from left to right. Probabilities are then assigned as a function of the internal state. However, the states of these two models represent different things. While *HMM*s' states are closely related to the states of the physical system being modelled, the states of an *OOM* keep the necessary information to predict the future observations and have nothing to do with the states of the physical system. Additionally and most importantly, it has been shown that *OOM*s are properly more expressive than *HMM*s (Jaeger, 2000). While originally invented for stochastic processes, with some small modifications these models can be also used to model probabilities over sets of finite words (Thon, 2011). Such probability distributions are of crucial importance in several fields, most notably speech recognition where they are used as language models, and in computational biology where they can model protein and DNA sequences.

Another approach that gives us a different insight into the stochastic language being modelled are Weighted Context Free Grammars (*WCFG*). Originally inspired by the broadly applied Context Free Grammar (*CFG*), they provide a hierarchical multi-scale view of the language. The *CFG*s are extended by assigning weights to each production rule. These weights can be then used to assign probabilities to words. *WCFG*s have been known for quite some time in the speech processing community (Lari and Young, 1990) and have been successfully used in many domains (Jurafsky et al., 1995; Sakakibara et al., 1994). Unlike *OOM*s that possess a provably optimal algorithm, *WCFG*s are trained using the *Inside-Outside* procedure (Lari and Young, 1990), member of the class of *Expectation Maximization* algorithms which do not have the convergence properties of *ES*. An interesting question that arises, and has numerous consequences to the learnability and expressibility of the model, is where to take the weights of the production rules from. While generally they can be assumed to come from some semi-ring (Goodman, 1999), in practice they are taken to be positive reals or from $(0, 1]$ (Smith and Johnson, 2007), so that some interpretation can be given. It has been already shown that under some normalization conditions, if we allow positive weights we can not represent more languages than if we just take weights from $(0, 1]$ and we moreover restrict them to have a probabilistic interpretation (Smith and Johnson, 2007).

In this thesis, I decide the question if there is a relationship between *OOM*s and unambiguous positively weighted grammars, more specifically if one of them properly subsume the other. The document is structured as follows. First, in Section 2, the mathematical framework needed to formally state the question is presented. Then, in the next two Sections 3 and 4, two theorems are proven which decide the question of inclusion between *OOM*s and unambiguous *WCFG*s. Finally, a summary of the results with suggestions for future work is presented in Section 5.

2 THEORETICAL SETTING

In the following I present the formal theoretical setting needed to state the problem in a rigorous mathematical manner. First, formal languages are discussed in general and then their extension using a weighting function is introduced. Then, the theory of weighted grammars is presented together with some important results which are crucial to the discussion in the subsequent sections. Finally, *OOM*s are precisely formulated in a manner suitable for modelling distributions over words and also their connection to other formalisms is outlined.

2.1 PROBABILISTIC LANGUAGES

The theory of formal languages has been extensively researched in the past, by both linguists for modelling natural human languages, and by computer scientists as the tool for analysing and specifying programming languages. Here I give only a short introduction, for a better overview see the standard text on the subject (Hopcroft et al., 1979).

An *alphabet* is a finite set of symbols and throughout this document it will be denoted by Σ. If we concatenate symbols from the alphabet into a finite sequence, we call the sequence a *string* over Σ. A special string is the string with zero occurrences of symbols, known as the *empty string* and will be denoted by ε. The length of a string w, denoted by $|w|$, is the number of positions for symbols in the string. As a convention, we set $|\varepsilon| = 0$. We denote by Σ^k the set of all words over Σ of length k. Furthermore, we define $\Sigma^+ = \cup_{k=1}^{\infty} \Sigma^k$ and $\Sigma^* = \Sigma^0 \cup \Sigma^+$. The concatenation of strings w and v will be denoted as wv, and as a shorthand for concatenating multiple copies of the same string we define $w^k = \underbrace{ww \cdots w}_{k \text{ copies}}$ if $k > 0$, and $w^0 = \varepsilon$.

The most important notion is that of a deterministic formal language, which is simply a collection of words over some alphabet.

Definition 2.1 (Language)**.** *A language over an alphabet Σ is a subset of Σ^*.*

Example. $\{b, bb, ba\}$ and $\{a^i b^i \mid i \in \mathbb{N}\}$ are languages over $\Sigma = \{a, b\}$.

Because this definition only covers the deterministic case where the membership of a word is a boolean function, we have to extend it so that we can assign weights to each word. This is better formalized in the next definition adapted from (Booth and Thompson, 1973).

Definition 2.2 (Word Function)**.** *A word function is a mapping $f : \Sigma^* \to \mathbb{R}$.*

Let $w_1, w_2, \cdots$ be the lexicographic ordering of Σ^* induced by some order $\prec$ on the symbols Σ. As a shorthand, we denote $\sum_{w \in \Sigma^*} f(w) = \sum_{i=1}^{\infty} f(w_i)$. If we add the restriction that $\sum_{w \in \Sigma^*} f(w)$ converges , we call f a measurable word function.

Note. We can still obtain a deterministic language $L \subset \Sigma^*$ of all words that appear in the language from a word function f by setting $L = \operatorname{supp} f = \{w \mid f(w) \neq 0\}$.
Remark. If f maps to $\mathbb{R}^{\geq 0}$ then the series converges iff it converges absolutely, so the ordering does not matter and it will not be explicitly defined.

This definition of a language distribution is very broad and general, and even allows negative word weights which do not have a clear intuitive meaning. As we are mostly interested in the case where

we can give a probabilistic interpretation of frequency, or rather likelihood, to the words, we can further restrict the set of word functions to include only those that can be potentially utilized in many applications.

Definition 2.3 (Probabilistic Word Function)**.** *A measurable word function f is called a probabilistic word function if it maps to $[0, 1]$ and $\sum_{w\in\Sigma^*} f(w) = 1$.*

Note. Any non-negative measurable word function can be normalized to a probabilistic word function if we divide by $\sum_{w\in\Sigma^*} f(w)$.
Example. To better illustrate this definition, we proceed with an example of a probabilistic word function over $\Sigma = \{a, b\}$. We define a word function f equivalent to the well-known Poisson distribution as

$$f(w) = \begin{cases} \frac{\lambda^k e^{-\lambda}}{k!}, & w \text{ is of the form } a^k b^k \text{ for some } k \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

where λ is a positive constant. It is trivially true that this is indeed a probabilistic word measure, because $f(a^k b^k) = Poisson(k; \lambda)$, hence it must hold that $\sum_{k=0}^{\infty} f(a^k b^k) = 1$.

2.2 Weighted Grammars

One formalism that can be used to assign weights to words using a hierarchical multi-scale view of the language is that of Weighted Context Free Grammars (*WCFG*). They are a generalization of classical grammars by assigning a weight to each of the production rules. It is important to note that we follow the standard literature (Smith and Johnson, 2007) and limit the weights to positive reals [1] . One reason for this restriction is that positive weights can be more easily interpreted and estimated from data. There exists a more general theory of these grammars when the weights and the operations come from a semi-ring (Goodman, 1999), but we only consider the special case when we use $(\mathbb{R}^{>0}, +, \times)$ as the underlying semi-ring.

The definition below is adapted from the one for *CFG*s in (Hopcroft et al., 1979), which is the standard introductory book for formal grammars.

Definition 2.4 (WCFG,GWCFG)**.** *A weighted context free grammar is a structure $G = (V, \Sigma, R, S, \Theta)$, where*

- *$V = \{A_1, A_2, \cdots, A_k\}$ is a finite set of non-terminals (variables).*
- *Σ is the alphabet of terminals.*
- *$R = \{r_1, r_2, \cdots, r_m\}$ is a finite set of rules of the form $A_i \to \alpha$, where $A_i \in V$ and $\alpha \in (V \cup \Sigma)^*$.*
- *$S \in V$ is the starting symbol.*
- *$\Theta : R \to \mathbb{R}^{>0}$ assigns positive weights to each rule in R.*

For convenience, we write $\theta_{A\to\alpha}$ for the weight of the rule $(A \to \alpha) \in R$. Moreover, if we allow weights from $\mathbb{R}$, rather than $\mathbb{R}^{>0}$, we say that the grammar is a General WCFG (GWCFG).

[1] Zero weights are not considered, because those rules can be simply removed from the grammar.

An important subclass of *WCFG*s can be obtained by restricting the weights in such a way to get a probabilistic interpretation of the branchings of the variables in the derivation trees.

Definition 2.5 (PCFG)**.** *A probabilistic context free grammar is a WCFG $G = (V, \Sigma, R, S, \Theta)$ with the restrictions that Θ maps to $(0, 1]$, and for each non-terminal $A \in V$ it holds that*

$$\sum_{(A \to \alpha_i) \in R} \theta_{A \to \alpha_i} = 1$$

Given the restrictions on the grammar which take away some of the degrees of freedom, it seems that *PCFG*s should be less expressive than *WCFG*s. However, it has been shown that under some normalization conditions *WCFG*s and *PCFG*s represent the same set of functions.

Theorem 2.1. *If $f \in \mathsf{WCFG}$ is a probabilistic measure function, then $f \in \mathsf{PCFG}$.*

A proof of the claim can be found in (Smith and Johnson, 2007).

Once we have assigned weights to each rule, we can use them to weigh derivations. There are two ways equivalent ways of looking at the extension of the weights from production rules to weighted derivations - either using leftmost derivations, or using derivation trees. I have chosen the latter approach as I find it more intuitive and better portrays the hierarchical nature of the grammars.

Definition 2.6 (Derivation Tree)**.** *A derivation tree for a WCFG $G = (V, \Sigma, R, S, \Theta)$ is a tree τ that satisfies the following conditions:*

1. *Each interior node is labelled with a variable from V and the root is labelled S.*
2. *Each leaf is labelled by an element of $V \cup \Sigma \cup \{\varepsilon\}$. However, if the node is labelled ε, it must be the only child of its parent.*
3. *If a parent is labelled A and its children are labelled $X_1, X_2, \cdots, X_k$ from left to right, then $A \to X_1 X_2 \cdots X_k$ is a production rule. We count this as one occurrence of the rule $A \to X_1 X_2 \cdots X_k$ in the tree τ.*

The yield of the tree τ, denoted by by $yield(\tau)$ is the concatenation of the labels of the leaves of the tree from left to right (in a depth-first manner). For a word w, we write $trees(w)$ for the set of all trees with yield w.

Every tree has a weight associated to it, which is equal to the product of all the rules used in its derivation. This is analogous to the product rule in probability, where the product of an intersection of independent events is equal to the product of their probabilities.

Definition 2.7 (Tree Weight)**.** *For a tree τ define its weight to be*

$$\theta(\tau) = \prod_{(A \to \alpha) \in R} (\theta_{A \to \alpha})^{g(A \to \alpha, \tau)}$$

where $g(A \to \alpha, \tau)$ is the number of occurrences of rule $A \to \alpha$ in the tree τ.

Example. Consider the grammar $G = (\underbrace{\{S, A, B\}}_{V}, \underbrace{\{a, b\}}_{\Sigma}, R, S, \Theta)$ with rules

$$\{S \xrightarrow{0.5} AB, \qquad S \xrightarrow{0.5} S, \qquad A \xrightarrow{0.3} Aa, \qquad A \xrightarrow{0.7} \varepsilon, \qquad B \xrightarrow{0.2} aBb, \qquad B \xrightarrow{0.8} b\}$$

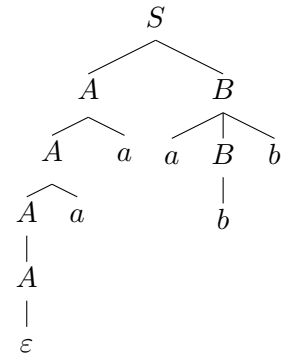

Figure 1: A derivation tree of $aaaabb$

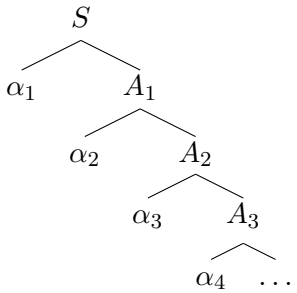

Figure 2: A typical right-linear derivation tree

A possible derivation tree is shown on Figure 1. The corresponding tree weight is

$$\underbrace{\theta_{S\to AB}}_{0.5} \times \underbrace{\theta_{A\to Aa}}_{0.3}{}^{2} \times \underbrace{\theta_{A\to\varepsilon}}_{0.7} \times \underbrace{\theta_{B\to aBb}}_{0.2} \times \underbrace{\theta_{B\to b}}_{0.8} = 0.00504$$

Definition 2.8 (Right-Linear Grammar)**.** *A grammar is called right-linear if every rule is of the form* $A \to \alpha B$ *or* $A \to \alpha$*, where* $A, B \in V$ *and* $\alpha \in \Sigma^*$*.*

The derivation trees that are associated with linear grammars have a characteristic form. They only branch on one side and typically look like Figure 2. Their importance comes from the classical theory of formal languages, because they are known to be equivalent with finite deterministic automata (Hopcroft et al., 1979). Analogously, as will be shown in the next subsection, there is a relationship between these grammars and the probabilistic counterparts of finite automata.

The weights of trees can be used to define weights over words, which is our primary interest. We define the weight of a word as the sum of the weights of all derivation trees for that word.

Definition 2.9 (Induced Word Measure)**.** *The weight of* $w \in \Sigma^*$*, denoted by* $\theta(w)$ [2] *under the WCFG* G *is* $\sum_{\tau \in trees(w)} \theta(\tau)$ *where* $trees(w)$ *is the set of all trees with yield* w [3]*. From this it clearly follows that every weighted grammar* G *induces a word measure* f_G *on* Σ^* *by setting* $f_G(w)$ *to be equal to the weight of* w *under* G*.*

The set of all word measures that can be induced by some WCFG will be denoted by WCFG*.*

[2] As a convenience we use $\theta(\cdot)$ to denote weights of words and trees. The meaning should be clear from the context.

[3] This implies that if w has no derivation trees, then $\theta(w) = 0$ because the sum has zero terms.

Remark. An important observation is that while the grammar might be a *PCFG*, it is not generally true that the induced word distribution is a probabilistic word measure. This is best shown by a very simple counter-example. Consider the grammar with a single non-terminal S, and a single production rule $S \xrightarrow{1} S$. It is clear that this is indeed a *PCFG*, but it assigns the weight of 0 to all words.

There is a special kind of grammars, where each word has at most one derivation tree. This means that in order to compute the weight for the word we do not have to consider all possibles parses of the word, which can be infinitely many for some grammars. The expressive power of this family of grammars is considered in Section 3.

Definition 2.10 (Unambiguous Grammar)**.** *A grammar is unambiguous if for all $w \in \Sigma^*$ it holds that $|trees(w)| \leq 1$. Or stated differently, if each word has at most one derivation.*

We denote with $\mathsf{WCFG}_{unambiguous}$ the set of all word measures that can be induced by some unambiguous WCFG.

2.3 Pumping Lemma for WCFGs

Naturally, there are some languages that *CFG*s can not represent and we need a tool to show that that is indeed the case. A combinatorial argument on the height of the derivation trees results in a very powerful lemma, known as the *Pumping Lemma*, which in some cases can be used to prove that some language is not context-free. The version of the lemma presented here is slightly different than the classical deterministic statement, because it applies to *WCFG*s, rather than *CFG*s. It will be a cornerstone of the proof in Section 3.

Lemma 2.2 (Pumping Lemma for unambiguous WCFGs)**.** *Let $f : \Sigma^* \to [0, 1]$ be a probabilistic word measure induced by an unambiguous WCFG $G = (V, \Sigma, R, S, \Theta)$. Then, there exists a constant N such that every $w \in \Sigma^*$ with $|w| \geq N$ and $\theta(w) \neq 0$ can be decomposed as $w = uvxyz$*

- $|vy| > 0$
- $|vxy| \leq N$
- $\theta(uv^n xy^n z) = k^n \theta(w)$ *for all $n \in \mathbb{N}^+$ and some $k > 0$.*

I do not provide a proof of the claim, because it is a well known fact and detailed proofs can be found in many standard books (Sipser, 1996). The only deviations from the standard statement are the constant k, which is the weight of the subtree being repeated (*"pumped"*), and the stronger third conclusion, which follows from the fact that the grammar is assumed to be unambiguous.

2.4 Multiplicity Automata

In the same way classical grammars can be altered to assign probabilities to words, finite automata (Hopcroft et al., 1979) can be also extended to their weighted counterparts - Multiplicity Automata (*MA*) (Schützenberg, 1961). As it will be later shown they subsume *OOM*s, and thus provide a different perspective of looking at *OOM*s as automata , rather than algebraic methods for modelling stochastic systems. Not only *MA*s put *OOM*s in this framework of probabilistic models inspired

from the theory of formal languages, but this representation of *OOM*s will also make some of the proofs easier and more intuitive.

Definition 2.11 (MA,PFA)**.** *A multiplicity automaton is a structure* $M = (\Sigma, Q, \phi, \iota, \tau)$ *where*

- Σ *is a finite alphabet.*
- Q *is a finite set of states.*
- $\phi : Q \times \Sigma \times Q \to \mathbb{R}$ *is the state transition function. We can inductively extend* ϕ *to words by setting*

$$\forall w \in \Sigma^*,\ a \in \Sigma\ :\ \phi(q, wa, q') = \sum_{s \in Q} \phi(q, w, s)\phi(s, a, q')$$

  and $\phi(q, \varepsilon, q') = \delta(q, q')$ [4]*.*
- $\iota : Q \to \mathbb{R}$ *is the initialization function.*
- $\tau : Q \to \mathbb{R}$ *is the termination function.*

If we further pose the following restrictions [5] ϕ, ι, τ *have codomain* $[0, 1]$*;* $\forall q \in Q : \tau(q) + \phi(q, \Sigma, Q) = 1$ *and* $\iota(Q) = 1$ *we get probabilistic finite automation (PFA).*

To a MA M *we assign the word function*

$$f_M : \Sigma^* \to \mathbb{R} : \quad f_M(x) = \sum_{q, q' \in Q} \iota(q)\phi(q, x, q')\tau(q')$$

Note. The function ϕ has $\mathbb{R}$ as a codomain, more specifically it can take negative values which is crucial to their ability to represent *OOM*s.

If we look closely at how f_M is defined, it takes the sum of all possible paths in the automaton. Because it is convenient to use paths when working with automata, we formally define them.

Definition 2.12 (Path in MA)**.** *We say that the sequence of states and symbols* $q_1 \overset{a_1}{\to} q_2, \cdots \overset{a_{n-1}}{\to} q_n \overset{a_n}{\to} q_{n+1}$ *is a path for the word* $w = a_1 a_2 \cdots a_n$ *with* n *symbols if* $\forall i \in \{1, 2, \cdots n\}$ *it holds that* $\phi(q_i, a_i, q_{i+1}) \neq 0$*,* $\iota(q_1) \neq 0$ *and* $\tau(q_{n+1}) \neq 0$*.*

The set of all induced functions is not of interest to this project, because we want those functions that can be used as probabilistic distributions. Hence, we define a special kind of *MA*s that satisfy this requirement, whose expressive power will be analysed in the next section.

Definition 2.13 (SMA)**.** *If for some MA* M *it is the case that* f_M *is a probabilistic word function, we say that* M *is a stochastic multiplicity automaton and we say that* $f = f_M$ *is a rational stochastic language. We shall denote the set of all probabilistic word functions that can be represented by an SMA by* SMA*.*

As already mentioned, similarly to the deterministic case, *MA*s can be converted to linear grammars. The following theorem proves this fact.

[4] The delta function is defined as $\delta(x, y) = 1$ if $x = y$ and 0 otherwise.

[5] For any function if some parameter is a set of values, then it denotes the sum over all possible values. For example $\phi(q, \Sigma, Q) = \sum_{a \in \Sigma} \sum_{q' \in Q} \phi(q, a, q')$

Theorem 2.3. *Given a MA $M = (\Sigma, Q, \phi, \iota, \tau)$, there exists an equivalent unambiguous right-linear* ***general*** [6] *WCFG with with $|Q| + 1$ non-terminals.*

Proof. Construct a grammar $G = (V, \Sigma, R, S, P)$ as follows

- $V = \{v_q \mid q \in Q\} \cup \{S\}$.
- $R = \{v_q \to a v_{q'} \mid \forall q, q' \in Q,\ a \in \Sigma \text{ such that } \phi(q, a, q') > 0\}$.
- For each rule $v_q \to a v_{q'}$, set a corresponding weight $\phi(q, a, q')$.
- Add the initialization rules $S \to v_q$ with weight $\iota(q)$ for all $q \in Q$ such that $\iota(q) \neq 0$.
- Finally, add the termination $v_q \to \varepsilon$ with weight $\tau(q)$ for all $q \in Q$ such that $\tau(q) \neq 0$.

It can be easily seen that the induced word function will agree on every word. First, which I will show below, there is a one-to-one correspondence between paths in the *MA* and derivation trees in the grammar, and moreover corresponding pairs have the same weight. The conclusion follows from the fact that both *MA*s and *WCFG*s assign weights to words in a similar manner. More specifically, the weight given to a word is the sum of the weights of all paths in a *MA*, and the sum of all derivations in a *WCFG*.

Let $w = a_1 a_2 a_3 \cdots a_n \in \Sigma^*$ be a word with n symbols.

($\Rightarrow$) If $q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_{n+1}$ is a path in the MA, then by construction there exists a derivation tree with internal nodes $S, v_{q_1}, v_{q_2}, \cdots, v_{q_n}, v_{q_{n+1}}$ and weight

$$\theta_{S \to v_{q_1}} \theta_{v_{q_1} \to a_1 v_{q_2}} \cdots \theta_{v_{q_n} \to a_n v_{q_{n+1}}} \theta_{v_{q_{n+1}} \to \varepsilon} = \iota(q_1)\phi(q_1, a_1, q_2) \cdots \phi(q_n, a_n, q_{n+1})\tau(q_{n+1})$$

which is the weight of the path.

($\Leftarrow$) Let τ be a derivation tree with internal nodes $S, v_{q_1}, v_{q_2}, \cdots, v_{q_{n+1}}$ ordered increasingly by the distance from the root. Then $q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_{n+1}$ has to be a path in the MA because for each q_i, q_{i+1} we only add a rule $q_i \to a_i q_{i+1}$ if there is such a transition in the MA. By an analogous argument of the previous part, the weights are equal.

□

2.5 Symbolic Stochastic Processes and OOMs

In this subsection, a brief introduction is given to the theory of finite-dimensional *OOM*s, which is only a special case of the more abstract theory presented in (Jaeger et al., 2005). The approach taken here follows (Thon, 2011).

As already mentioned, *OOM*s were originally designed to model stationary symbolic stochastic processes. This is quite different from probabilities over words, because stochastic processes are in general infinite.

Definition 2.14 (Symbolic Stochastic Process)**.** *A function $f : \Sigma^* \to [0, 1]$ that satisfies $f(\varepsilon) = 1$ and $\forall w \in \Sigma^* : f(w) = \sum_{a \in \Sigma} f(wa)$ is said to be a symbolic stochastic process.*

Definition 2.15 (SS)**.** *A d-dimensional sequential system M is a structure $(\sigma, \{\tau_x\}_{x \in \Sigma}, \omega_\varepsilon)$ where*

[6] The question about the relationship between positive *WCFG*s and *SMA*s is discussed in the next two sections.

- *σ is a linear evaluation map $\mathbb{R}^d \to \mathbb{R}$.*
- *Each τ_x is a matrix (i.e. a linear operator) from $\mathbb{R}^{d\times d}$.*
- *$\omega_\varepsilon \in \mathbb{R}^d$ is the initial state* [7] *.*

To each M we assign a word function, called the external function, which is defined as

$$f_M : \Sigma^* \to R : \quad f_M(x_1 x_2 \cdots x_n) = \sigma\tau_{x_n} \cdots \tau_{x_2}\tau_{x_1}\omega_\varepsilon$$

The main idea behind *SS*s is that observations are identified with the operators themselves. More specifically, for every possible symbol $a \in \Sigma$ that can be observed there exists a corresponding operator τ_a that updates the current state of the model. Hence, after observing a sequence $a_1 a_2 \cdots a_n$, the new state of the model will be updated to $\tau_{a_n}\tau_{a_{n-1}} \cdots \tau_{a_2}\tau_{a_1}\omega_\varepsilon$. As already mentioned, unlike *HMM*s, this state is not bound to the physical states of the observed system. We are rather interested in keeping the necessary information which will allow us to represent the probability function of future observations. This can be best expressed in the language of linear algebra - we can think of the components of the state vector as the representation of the probability function for future observations with respect to some basis. Thus, we can model all processes whose distribution functions span a finite dimensional vector space by choosing sufficiently large dimension d.

As already hinted in the previous section, there is a formal equivalence relationship between *MA*s and *SS*s. This fact can be easily seen by rewriting *MA*s in terms of matrices and vectors (Thon, 2011). We can say that they are just different perspectives of the same mathematical mechanism.

Besides the fact that they were originally invented for modelling infinite sequences, every stochastic process can be used to model distributions over words by the addition of another stopping symbol which we shall denote by \$. This of course applies to *SS*s and the following definition characterizes the set of languages that can be represented in such a way.

Definition 2.16 (OOM)**.** *An Observable Operator Model is a sequential system M such that the induced word function f_M is symbolic stochastic process. We say that a word function f can be represented by an OOM if there exists an OOM M with an extra symbol \$ such that*

$$\forall w \in \Sigma^* \ : \ f(w) = f_M(w\$)$$

The set of all probabilistic word functions that can be represented in such a way will be denoted by OOM [8] *.*

It turns out that *OOM*s and *MA*s (or equivalently *SS*s) model the same set of probabilistic languages.

Lemma 2.4.

$$\mathsf{OOM} = \mathsf{SMA}$$

A proof of the claim can be found in (Thon, 2011). As there is an equivalence between these two sets, in the following discussion I only consider OOM.

[7] We treat elements of $\mathbb{R}^d$ as column vectors.

[8] In (Thon, 2011) *OOM*s that induce a probabilistic word function are called *terminating*.

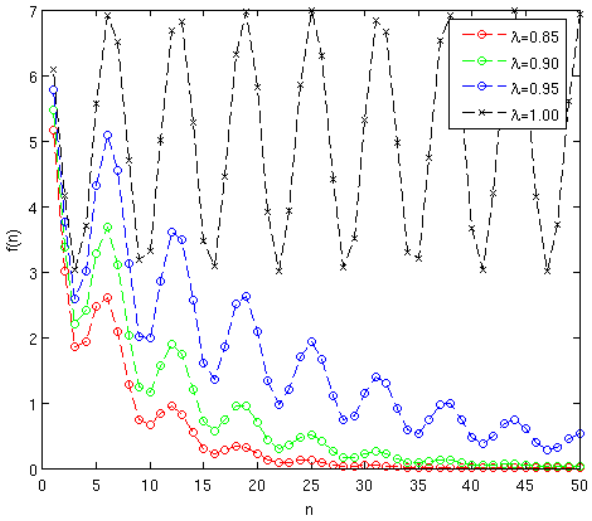

Figure 3: Non-scaled probability clock and example PCL functions.

3 UNAMBIGUOUS WCFGS ARE NOT A SUPERSET OF OOMS

In this section I present the proof that unambiguous *WCFG*s do not subsume *OOM*s. This is done by constructing an example that can be easily seen to be a member of OOM, but I will show that it is not a member of $\mathsf{WCFG}_{unambiguous}$. First, a special set of probabilistic languages, named *clock languages*, are defined in a manner similar to the definition of the clock distribution in (Jaeger, 2000) that is used to show that *OOM*s properly extend *HMM*s. After proving that they can be indeed normalized to a probabilistic word distribution, I proceed with the proof of the main claim.

Definition 3.1 (Rotation Matrix)**.** *A rotation matrix R acting on $\mathbb{R}^3$ of an angle of θ around the x-axis has the form*

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

Definition 3.2 (PCL)**.** *Let R be a rotation matrix of an angle θ which is a non-rational multiple of π, $\lambda \in [0,1)$ and $\mathbf{w_0} \in (\mathbb{R}^+ \cup \{0\})^3$ the initial state. We furthermore restrict $\mathbf{w_0}$ to satisfy* [9] *$w_{0,1} - 2w_{0,2} - 2w_{0,3} > 0$. For each such constants, we define a word measure* [10] [11]

$$f(a^n) = \mathbf{1}\lambda^n R^n \mathbf{w}_0$$

over the singleton alphabet $\Sigma = \{a\}$. We say that f has parameters $(R, \lambda, \mathbf{w}_0)$. The set of all such functions shall be called probabilistic clock languages, and denoted by PCL.

[9] The second sub-index of $\mathbf{w}_0$ is used to denote the vector component.

[10] The convergence is proven in 3.1.

[11] We use $\mathbf{1}$ to denote the row vector of ones $(1, 1, 1)$.

Some example members of PCL can be seen on Figure 3. As one can see, the probability oscillates all the time, and intuitively we need negative values which are non-allowed as weights for *WCFG*s to represent this behaviour. Moreover, we add a decaying factor λ^n which should guarantee that the resulting distribution can be normalized, so we can have a probabilistic word function that exhibits this oscillatory behaviour. This last fact is proven in the next theorem.

Theorem 3.1 (PCL convergence)**.** *For every $f \in \mathsf{PCL}$ it holds that $\sum_{n=1}^{\infty} f(a^n)$ converges.*

Proof. Let $(R, \lambda, \mathbf{w_0})$ be the parameters of $f \in \mathsf{PCL}$. For readability, we denote $f(a^n)$ by a_n. First, we show that $\lim_{n\to\infty} \sqrt[n]{|a_n|} = \lambda \in [0, 1)$. The claim of the proof then follows from the *Root Test* (Rudin, 1964).

To simplify the analysis of the sequence, using simple algebra and the fact that R^n is a rotation matrix around the x-axis of an angle of $n\theta$, we can write the analytical formula for the n-th term

$$a_n = \lambda^n(w_{0,1} + w_{0,2}(\cos(n\theta) + \sin(n\theta)) + w_{0,3}(\cos(n\theta) - \sin(n\theta)))$$

We define two sequences a_n^+ and a_n^- that bound a_n from above and below respectively.

(i)

$$\begin{aligned} a_n &= \lambda^n(w_{0,1} + w_{0,2}(\underbrace{\cos(n\theta)}_{\leq 1} + \underbrace{\sin(n\theta))}_{\leq 1} + w_{0,3}(\underbrace{\cos(n\theta)}_{\leq 1} - \underbrace{\sin(n\theta)))}_{\geq -1} \\ &\leq \lambda^n(w_{0,1} + 2w_{0,2} + 2w_{0,3}) \\ &= \lambda^n C \end{aligned}$$

We can always pick $C > 0$. Hence, if we set $a_n^+ = \lambda^n C$ we get the upper bound.

(ii)

$$\begin{aligned} a_n &= \lambda^n(w_{0,1} + w_{0,2}(\underbrace{\cos(n\theta)}_{\geq -1} + \underbrace{\sin(n\theta))}_{\geq -1} + w_{0,3}(\underbrace{\cos(n\theta)}_{\geq -1} - \underbrace{\sin(n\theta)))}_{\leq 1} \\ &\geq \lambda^n(w_{0,1} - 2w_{0,2} - 2w_{0,3}) \\ &= \lambda^n F \end{aligned}$$

Because of the strict inequality restriction on $\mathbf{w}_0$ in the Definition 3.2, it directly follows that $F > 0$. By defining $a_n^- = \lambda^n F$ we get the lower bound.

Two important limits, that also turn out to be easy to calculate are

$$\lim_{n\to\infty} \sqrt[n]{|a_n^+|} \overset{a_n^+ > 0}{=} \lim_{n\to\infty} \sqrt[n]{\lambda^n C} = \lim_{n\to\infty} \lambda C^{1/n} = \lambda \lim_{n\to\infty} C^{1/n} = \lambda \tag{1}$$

and

$$\lim_{n\to\infty} \sqrt[n]{|a_n^-|} \overset{a_n^- > 0}{=} \lim_{n\to\infty} \sqrt[n]{\lambda^n F} = \lim_{n\to\infty} \lambda F^{1/n} = \lambda \lim_{n\to\infty} F^{1/n} = \lambda \tag{2}$$

Because $0 < a_n^- \leq a_n \leq a_n^+$ it follows that $\sqrt[n]{|a_n^-|} \leq \sqrt[n]{|a_n|} \leq \sqrt[n]{|a_n^+|}$. By the *Squeeze Theorem* we can conclude that $\lim_{n\to\infty} \sqrt[n]{|a_n|} = \lambda < 1$.

The claim of the theorem follows from the *Root Test*. □

Corollary. *Every $f \in \mathsf{PCL}$ can be normalized to a probabilistic word function by appropriately scaling the initial vector $\mathbf{w_0}$.*

By construction of PCL, we know that there is an *SMA*, and hence an *OOM*, that can model this language. The following theorem completes the proof that unambiguous *WCFG*s are not more powerful than *OOM*s.

Theorem 3.2. *The probabilistic clock language p with parameters $(R, \lambda, \underbrace{(5,1,1)/Z}_{\mathbf{w}_0})$ can not be generated by an* ***unambiguous grammar*** *($Z \in \mathbb{R}^{>0}$ is chosen as the normalization constant which was proven to exist in Theorem 3.1).*

Proof. Assume the language can be generated by some unambiguous *WCFG* G and let N be the pumping constant from Theorem 2.2. Then, for all $w = a^n$ with $n \geq N$, $(\exists\, T \in \mathbb{N}).\ 1 \leq T \leq N$ and $(\exists r).\ r \in \mathbb{R}^{>0}$ such that the following holds

$$\forall k \in \mathbb{N}^+ : \quad r^k p(a^n) = p(a^{n+Tk})$$

Where r is the weight of the subtree being repeated. If we expand p using its definition we get

$$\begin{aligned} r^k \mathbf{1}\lambda^n R^n \mathbf{w_0} &= \mathbf{1}\lambda^{n+Tk} R^{n+Tk} \mathbf{w_0} \\ (\frac{r}{\lambda^T})^k \mathbf{1} R^n \mathbf{w_0} &= \mathbf{1} R^{n+Tk} \mathbf{w_0} \end{aligned}$$

Because n can be fixed, the expression can be simplified by setting $\hat{\lambda} = r/\lambda^T$ and $K = \mathbf{1}R^n\mathbf{w}_0$. This results in

$$\hat{\lambda}^k K = \mathbf{1} R^{n+Tk} \mathbf{w_0}$$

where K and $\hat{\lambda}$ are positive constants. This equality must hold for all $k \in \mathbb{N}^+$. I will show that this is not possible by analysing the asymptotic behaviour of both sides as $k \to \infty$.

(i) If $\hat{\lambda} > 1$, then left hand side goes to ∞ as $k \to \infty$, but the right hand side is bounded from above by the constant $C > 0$ which was defined for a_n^+.

(ii) If $\hat{\lambda} < 1$ then the left hand side goes to 0 as $k \to \infty$, but the right hand side is bounded from below by $F > 0$ which was defined for a_n^-.

(iii) If $\hat{\lambda} = 1$ then the left hand side is constant, but the right hand side is not constant. This can be easily seen, because in the previous theorem we have shown that for every m it holds that

$$\begin{aligned} \mathbf{1}R^m\mathbf{w}_0 = b_m &= \underbrace{w_{0,1}}_{=5} + \underbrace{w_{0,2}}_{=1}(\cos(m\theta) + \sin(m\theta)) + \underbrace{w_{0,3}}_{=1}(\cos(m\theta) - \sin(m\theta)) \\ &= 5 + 2cos(m\theta) \end{aligned}$$

In this case $m = n + Tk$, so the right hand side becomes $5 + 2cos(n\theta + Tk\theta)$. The angle θTk is always a non-rational multiple of π, so the cosine component can not be constant.

□

Corollary. $\mathsf{OOM} \not\subset \mathsf{WCFG}_{unambiguous}$

4 OOMs ARE NOT A SUPERSET OF WCFGs

In this section I consider the different direction of the inclusion relationship between the two sets of probabilistic languages. Given that *OOM*s are similar to finite automata and the fact that deterministic finite automata are a proper subset of *CFG*s, this result is not surprising. We see that the addition of negative weights still does not give us enough expressive power to represent some languages from WCFG. For the proof, a very simple word measure over the alphabet $\Sigma = \{a, b\}$ is considered. We define the word function

$$p(w) = \begin{cases} 1/2^{i+1}, & w \text{ is of the form } b^i a^i \text{ for some } i \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

It can be easily checked that p is a probabilistic measure, because $\sum_{n=1}^{\infty} 1/2^n = 1$.

Theorem 4.1. *The probabilistic word measure p defined above can be modelled by a PCFG, but can not be modelled using an OOM.*

Proof. The measure is clearly modelled with an unambiguous *PCFG* G with single non-terminal S, and productions $\{S \overset{0.5}{\to} \varepsilon,\ S \overset{0.5}{\to} bSa\}$.

Assume that $p \in \mathsf{OOM}$. Then there exists a d-dimensional *SMA* $M = (\sigma, \{\tau_a, \tau_b\}, \mathbf{w}_0)$ that can model the language. I will show that this leads to a contradiction.

Consider the $d + 1$ vectors $\mathbf{v}_0 = \mathbf{w}_0,\ \mathbf{v}_1 = \tau_b \mathbf{w}_0,\ \mathbf{v}_2 = \tau_b^2 \mathbf{w}_0, \cdots,\ \mathbf{v}_d = \tau_b^d \mathbf{w}_0$. It can be easily shown that they are all non-zero and distinct. Assume that for some $i \in \{1, 2, \cdots, d\}$ it holds that $\mathbf{v}_i = \mathbf{0}$. By definition, $p(b^i a^i) = \sigma \tau_a^i \mathbf{v}_i = \sigma \tau_a^i \mathbf{0} = 0$, which is a contradiction. Assume that for some $i, j \in \{1, 2, \cdots, d\}$ such that $i \neq j$ it is true that $\mathbf{v}_i = \mathbf{v}_j$. This would imply that $p(b^i a^i) = \sigma \tau_a^i \mathbf{v}_i = \sigma \tau_a^i \mathbf{v}_j \overset{i \neq j}{=} \mathbf{0}$, which is again a contradiction.

Because we can not have a set of $d+1$ independent vectors in a d-dimensional vector space (Axler, 1997), they must be dependent. By definition, there exist scalars $\alpha_i \in \mathbb{R}$, not all zero, such that

$$\alpha_0 \mathbf{v}_0 + \alpha_1 \mathbf{v}_1 + \cdots + \alpha_d \mathbf{v}_d = \mathbf{0}$$

Let i be the index such that $\alpha_i \neq 0$. Then using simple algebra we get

$$\mathbf{v}_i = -\frac{1}{\alpha_i} \sum_{j=0, j \neq i}^{d} \alpha_j \mathbf{v}_j$$

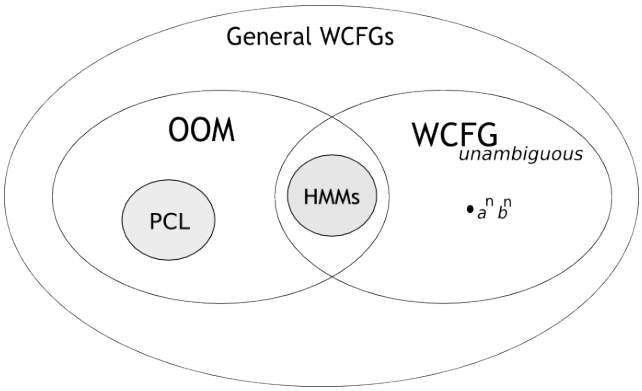

Figure 4: Relationships between the classes of probabilistic languages.

If we multiply both sides from the left by $\sigma\tau_a^i$ and use the linearity of the map, the equation becomes

$$\sigma\tau_a^i\tau_b^i\mathbf{w}_0 = -\frac{1}{\alpha_i}\sum_{j=0,j\neq i}^{d}\alpha_j\sigma\tau_a^i\tau_b^j\mathbf{w}_0$$

By the definition of the language it must hold that $p(b^j a^i) = \sigma\tau_a^i\tau_b^j\mathbf{w}_0 \neq \mathbf{0}$ iff $i = j$. Thus, the left hand side of the equation is non-zero, but all the terms in the summation on the right hand side are zero. This is a contradiction, hence the claim of the theorem must hold. □

Corollary. $\mathsf{WCFG}_{unambiguous} \not\subset \mathsf{OOM}$

5 Conclusion and Future Work

In this report I developed the mathematical framework needed to rigorously analyse the expressive powers of *OOM*s and *WCFG*s when used to assign probabilities to finite sequences of symbols. Despite their different formulation, one derived as the natural extension of *HMM*s and the other inspired from the theory formal languages, it turned out that there is a relation between the two. Namely, *OOM*s can be seen as a special kind of weighted grammars with weights that can also take negative values. With the two theorems in Sections 3 and 4 it was shown that that neither of them is strictly more expressive. The relationship between these sets of languages can be best understood from Figure 4.

There are many possibilities to extend the results from this thesis. One obvious question that is raised is if ambiguous weighted grammars subsume *OOM*s. A simple extension of the argument in Section 3 does not work, because instead of equality we get inequalities which can not be easily proven to be contradictory based on their asymptotic behaviour. Another important that is of a more practical interest and deserves attention is how well can *OOM*s approximate languages generated by a *WCFG*.

All things considered, this project made a connection between two different areas of machine learning research and answered an important question on the relationship of their expressive powers. It has also set up the theoretical basis needed to conduct further research in this area.

6 ACKNOWLEDGEMENTS

I owe my deepest gratitude to my research mentor Professor Dr. Herbert Jeager for his continuous support and guidance during this semester. I would also like to thank Michael Thon for sharing with me the draft of his paper, and for his in-depth discussion during the meetings which were of great help to me.

REFERENCES

Axler, S., 1997. Linear algebra done right. Springer Verlag.

Booth, T., Thompson, R., 1973. Applying probability measures to abstract languages. Computers, IEEE Transactions on 100 (5), 442–450.

Goodman, J., 1999. Semiring parsing. Computational Linguistics 25 (4), 573–605.

Hopcroft, J., Motwani, R., Ullman, J., 1979. Introduction to automata theory, languages, and computation. Vol. 3. Addison-wesley Reading, MA.

Jaeger, H., 2000. Observable operator models for discrete stochastic time series. Neural Computation 12 (6), 1371–1398.

Jaeger, H., Zhao, M., Kretzschmar, K., Oberstein, T., Popovici, D., Kolling, A., 2005. Learning observable operator models via the ES algorithm. New directions in statistical signal processing: From systems to brains.

Jurafsky, D., Wooters, C., Segal, J., Stolcke, A., Fosler, E., Tajchaman, G., Morgan, N., 1995. Using a stochastic context-free grammar as a language model for speech recognition. In: icassp. IEEE, pp. 189–192.

Lari, K., Young, S., 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. Computer speech & language 4 (1), 35–56.

Rudin, W., 1964. Principles of mathematical analysis. Vol. 1976. McGraw-Hill New York;.

Sakakibara, Y., Brown, M., Hughey, R., Mian, I., Sjölander, K., Underwood, R., Haussler, D., 1994. Stochastic context-free grammers for tRNA modeling. Nucleic Acids Research 22 (23), 5112.

Schützenberg, M., 1961. On the definition of a family of automata. Information and control 4 (2-3).

Sipser, M., 1996. Introduction to the Theory of Computation. International Thomson Publishing.

Smith, N., Johnson, M., 2007. Weighted and probabilistic context-free grammars are equally expressive. Computational Linguistics 33 (4), 477–491.

Thon, M., 2011. Links between multiplicity automata, observable operator models and predictive state representations, draft.