Reestimation of Reified Rules in Semiring Parsing and Biparsing

Markus Saers and **Dekai Wu**
Human Language Technology Center
Dept. of Computer Science and Engineering
Hong Kong University of Science and Technology
{masaers|dekai}@cs.ust.hk

Abstract

We show that reifying the rules from hyperedge weights to first-class graph nodes automatically gives us rule expectations in any kind of grammar expressible as a deductive system, without any explicit algorithm for calculating rule expectations (such as the inside-outside algorithm). This gives us expectation maximization training for any grammar class with a parsing algorithm that can be stated as a deductive system, for free. Having such a framework in place accelerates turnover time for experimenting with new grammar classes and parsing algorithms—to implement a grammar learner, only the parse forest construction has to be implemented.

1 Introduction

We propose *contextual probability* as a quantity that measures how often something has been used in a corpus, and when calculated for rules, it gives us everything needed to calculate rule expectations for expectation maximization. For labeled spans in context-free parses, this quantity is called *outside probability*, and in semiring (bi-) parsing, it is called *reverse value*. The inside-outside algorithm for reestimating context-free grammar rules uses this quantity for the symbols occurring in the parse forest. Generally, the contextual probability is:

> The *contextual probability* of ***something***
> is the sum of the probabilities
> of all contexts where it was used.

For symbols participating in a parse, we could state it like this:

> The *contextual probability* of ***an item***
> is the sum of the probabilities
> of all contexts where it was used.

...which is exactly what we mean with outside probability. In semiring (bi-) parsing, this quantity is called reverse value, but in this framework it is also defined for rules, which means that we could restate our boxed statement as:

> The *contextual probability* of ***a rule***
> is the sum of the probabilities
> of all contexts where it was used.

This opens up an interesting line of inquiry into what this quantity might represent. In this paper we show that the contextual probabilities of the rules contain precisely the new information needed in order to calculate the expectations needed to reestimate the rule probabilities. This line of inquiry was discovered while working on a preterminalized version of linear inversion transduction grammars (LITGs), so we will use these preterminalized LITGs (Saers and Wu, 2011) as an example throughout this paper.

We will start by examining semiring parsing (parsing as deductive systems over semirings, Section 3), followed by a section on how this relates to weighted hypergraphs, a common representation of parse forests (Section 4). This reveals a disparity between weighted hypergraphs and semiring parsing. It seems like we are forced to choose between the inside-outside algorithm for context-free grammars

Proceedings of SSST-5, Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 70–78,
ACL HLT 2011, Portland, Oregon, USA, June 2011.

on the one side, and the flexibility of grammar formalism and parsing algorithm development afforded by semiring (bi-) parsing. It is, however, possible to have both, which we will show in Section 5. An integral part of this unification is the concept of contextual probability. Finally, we will offer some conclusions in Section 6.

2 Background

A common view on probabilistic parsing—be it bilingual or monolingual—is that it involves the construction of a *weighted hypergraph* (Billot and Lang, 1989; Manning and Klein, 2001; Huang, 2008). This is an appealing conceptualization, as it separates the construction of the parse forest (the actual hypergraph) from the probabilistic calculations that need to be carried out. The calculations are, in fact, given by the hypergraph itself. To get the probability of the sentence (pair) being parsed, one simply have to query the hypergraph for the value of the *goal node*. It is furthermore possible to abstract away the calculations themselves, by defining the hypergraph over an arbitrary *semiring*. When the *Boolean semiring* is used, the value of the goal node will be *true* if the sentence (pair) is a member of the language (or transduction) defined by the grammar, and *false* otherwise. When the *probabilistic semiring* is used, the probability of the sentence (pair) is attained, and with the *tropical semiring*, the probability of the most likely tree is attained. To further generalize the building of the hypergraph—the parsing algorithm—a *deductive system* can be used. By defining a hand-full of deductive rules that describe how *items* can be constructed, the full complexities of a parsing algorithm can be very succinctly summarized. Deductive systems to represent parsers and semirings to calculate the desired values for the parses were introduced in Goodman (1999).

In this paper we will reify the grammar rules by moving them from the meta level to the object level—effectively making them first-class citizens of the parse trees, which are no longer weighted hypergraphs, but *mul/add-graphs*. This move allows us to calculate rule expectations for expectation maximization (Dempster et al., 1977) as part of the parsing process, which significantly shortens turn-over time for experimenting with different grammar formalisms.

Another approach which achieve a similar goal is to use a *expectation semiring* (Eisner, 2001; Eisner, 2002; Li and Eisner, 2009). In this semiring, all values are pairs of probabilities and expectations. The inside-outside algorithm with the expectation semiring requires the usual inside and outside calculations over the probability part of the semiring values, followed by a third traversal over the parse forest to populate the expectation part of the semiring values. The approach taken in this paper also requires the usual inside and outside calculations, but o third traversal of the parse forest. Instead, the proposed approach requires two passes over the rules of the grammar per EM iteration. The asymptotic time complexities are thus equivalent for the two approaches.

2.1 Notation

We will use $\mathbf{w}$ to mean a monolingual sentence, and index the individual tokens from 0 to $|\mathbf{w}| - 1$. This means that $\mathbf{w} = w_0, \ldots, w_{|\mathbf{w}|-1}$. We will frequently use spans from this sentence, and denote them $w_{i..j}$, which is to be interpreted as array slices, that is: including the token at position i, but excluding the token at position j (the interval $[i, j)$ over $\mathbf{w}$, or $w_i, \ldots, w_{j-1}$). A sentence $\mathbf{w}$ thus corresponds to the span $w_{0..|\mathbf{w}|}$. We will also assume that there exists a grammar $G = \langle N, \Sigma, S, R\rangle$ or a transduction grammar (over languages L_0 and L_1) $G = \langle N, \Sigma, \Delta, S, R\rangle$ (depending on the context), where N is the set of nonterminal symbols, Σ is a set of (L_0) terminal symbols, Δ is a set of (L_1) terminal symbols, $S \in N$ is the dedicated start symbol and R is a set of rules appropriate to the grammar. A stochastic grammar is further assumed to have a parameterization function θ, that assigns probabilities to all the rules in R. For general L_0 tokens we will use lower case letters from the beginning of the alphabet, and for L_1 from the end of the alphabet. For specific sentences we will use $\mathbf{e} = e_{0..|\mathbf{e}|}$ to represent an L_0 sentence and $\mathbf{f} = f_{0..|\mathbf{f}|}$ to represent an L_1 sentence.

3 Semiring parsing

Semiring parsing was introduced in Goodman (1999), as a unifying approach to parsing. The gen-

eral idea is that any parsing algorithm can be expressed as a deductive system. The same algorithm can then be used for both traditional grammars and stochastic grammars by changing the semiring used in the deductive system. This approach thus separates the algorithm from the specific calculations it is used for.

Definition 1. *A semiring is a tuple* $\langle \mathbb{A}, \oplus, \otimes, \mathbf{0}, \mathbf{1} \rangle$*, where* $\mathbb{A}$ *is the set the semiring is defined over,* $\oplus$ *is an associative, commutative operator over* $\mathbb{A}$*, with identity element* $\mathbf{0}$ *and* $\otimes$ *is an associative operator over* $\mathbb{A}$ *distributed over* $\oplus$*, with identity element* $\mathbf{1}$*.*

Semirings can be intuitively understood by considering the *probabilistic semiring*: $\langle \mathbb{R}^+, +, \times, 0, 1 \rangle$, that is: the common meaning of addition and multiplication over the positive real numbers (including zero). Although this paper will have a heavy focus on the probabilistic semiring, several other exists. Among the more popular are the *Boolean semiring* $\langle \{\top, \bot\}, \vee, \wedge, \bot, \top \rangle$ and the *tropical semiring* $\langle \mathbb{R}^+ \cup \{\infty\}, min, +, \infty, 0 \rangle$ (or $\langle \mathbb{R}^- \cup \{-\infty\}, max, +, -\infty, 0 \rangle$ which can be used for probabilities in the logarithmic domain).

The deductive systems used in semiring parsing have three components: an *item* representation, a *goal item* and a set of *deductive rules*. Taking CKY parsing (Cocke, 1969; Kasami and Torii, 1969; Younger, 1967) as an example, the items would have the form $A_{i,j}$, which is to be interpreted as the span $w_{i..j}$ of the sentence being parsed, labeled with the nonterminal symbol A. The goal item would be $S_{0,|\mathbf{w}|}$: the whole sentence labeled with the start symbol of the grammar. Since the CKY algorithm is a very simple parsing algorithm, it only has two deductive rules:

$$\frac{A \rightarrow a,\ \mathbb{I}_a(w_{i..j})}{A_{i,j}}\ {\scriptstyle 0 \leq i \leq j \leq |\mathbf{w}|} \tag{1}$$

$$\frac{B_{i,k},\ C_{k,j},\ A \rightarrow BC}{A_{i,j}} \tag{2}$$

Where $\mathbb{I}_a(\cdot)$ is the terminal indicator function for the semiring. The general form of a deductive rule is that the *conditions* (entities over the line) yield the *consequence* (the entity under the line) given that the *side conditions* (to the right of the line) are satisfied. We will make a distinction between conditions that are themselves items, and conditions that are not. The non-item conditions will be called *axioms*, and are exemplified above by the indicator function ($\mathbb{I}_a(w_{i..j})$ which has a value that depends only on the sentence) and the rules ($A \rightarrow a$ and $A \rightarrow BC$ which have values that depends only on the grammar).

The indicator function might seem unnecessary, but allows us to reason under uncertainty regarding the input. In this paper, we will assume that we have perfect knowledge of the input (but for generality, we will not place it as a side condition). The function is defined such that:

$$\forall a \in \Sigma^* : \mathbb{I}_a(w) = \begin{cases} \mathbf{1} & \text{if } a = w \\ \mathbf{0} & \text{otherwise} \end{cases}$$

An important concept of semiring parsing is that the deductive rules also specify how to arrive at the value of the consequence. Since it is the first value computed for a node, we will call it α, and the general way to calculate it given a deductive rule and the α-values of the conditions is:

$$\alpha(b) = \bigotimes_{i=1}^{n} \alpha(a_i) \quad \text{iff} \quad \frac{a_1, \ldots, a_n}{b}\ {\scriptstyle c_1, \ldots, c_m}$$

If the same consequence can be produced in several ways, the values are summed using the $\oplus$ operator:

$$\alpha(b) = \bigoplus_{\substack{n, a_1, \ldots, a_n \\ \text{such that} \\ \frac{a_1, \ldots, a_n}{b}}} \bigotimes_{i=1}^{n} \alpha(a_i)$$

The α-values of axioms depend on what kind of axiom it is. For the indicator function, the α-value is the value of the function, and for grammar rules, the α-value is the value assigned to the rule by the parameterization function θ of the grammar.

The α-value of a consequence corresponds to the value of everything leading up to that consequence. If we are parsing with a context-free grammar and the probabilistic semiring, this corresponds to the inside probability.

3.1 Reverse values

When we want to reestimate rule probabilities, it is not enough to know the probabilities of arriving at different consequences, we also need to know how likely we are to need the consequences as a condition for other deductions. These values are called

$$\frac{S \to A}{A_{0,|\mathbf{e}|,0,|\mathbf{f}|}}, \qquad \frac{A_{s,s,u,u},\ A \to {}^{\epsilon}\!/\!_{\epsilon}}{\mathcal{G}},$$

$$\frac{B_{s',t,u',v},\ B \to [XA],\ X \to {}^{a}\!/\!_{x}\ ,\ \mathbb{I}_{a/x}\left({}^{e_{s..s'}}\!/\!_{f_{u..u'}}\right)}{A_{s,t,u,v}}\ {\scriptstyle 0\le s\le s',\ 0\le u\le u'},$$

$$\frac{B_{s,t',u,v'},\ B \to [AX],\ X \to {}^{a}\!/\!_{x}\ ,\ \mathbb{I}_{a/x}\left({}^{e_{t'..t}}\!/\!_{f_{v'..v}}\right)}{A_{s,t,u,v}}\ {\scriptstyle t'\le t\le|\mathbf{e}|,\ v'\le v\le|\mathbf{f}|},$$

$$\frac{B_{s',t,u,v'},\ B \to \langle XA\rangle,\ X \to {}^{a}\!/\!_{x}\ ,\ \mathbb{I}_{a/x}\left({}^{e_{s..s'}}\!/\!_{f_{v'..v}}\right)}{A_{s,t,u,v}}\ {\scriptstyle 0\le s\le s',\ v'\le v\le|\mathbf{f}|},$$

$$\frac{B_{s,t',u',v},\ B \to \langle AX\rangle,\ X \to {}^{a}\!/\!_{x}\ ,\ \mathbb{I}_{a/x}\left({}^{e_{t'..t}}\!/\!_{f_{u..u'}}\right)}{A_{s,t,u,v}}\ {\scriptstyle t'\le t\le|\mathbf{e}|,\ 0\le u\le u'}$$

Figure 2: Deductive system describing a PLITG parser. The symbols A, B and S are nonterminal symbols, while X represents a *preterminal* symbol.

$$\frac{S \to A}{A_{0,|\mathbf{e}|,0,|\mathbf{f}|}}, \qquad \frac{A_{s,s,u,u},\ A \to {}^{\epsilon}\!/\!_{\epsilon}}{\mathcal{G}},$$

$$\frac{B_{s',t,u',v},\ B \to [{}^{a}\!/\!_{x}\ A],\ \mathbb{I}_{a/x}\left({}^{e_{s..s'}}\!/\!_{f_{u..u'}}\right)}{A_{s,t,u,v}}\ {\scriptstyle 0\le s\le s',\ 0\le u\le u'},$$

$$\frac{B_{s,t',u,v'},\ B \to [A\ {}^{a}\!/\!_{x}],\ \mathbb{I}_{a/x}\left({}^{e_{t'..t}}\!/\!_{f_{v'..v}}\right)}{A_{s,t,u,v}}\ {\scriptstyle t'\le t\le|\mathbf{e}|,\ v'\le v\le|\mathbf{f}|},$$

$$\frac{B_{s',t,u,v'},\ B \to \langle {}^{a}\!/\!_{x}\ A\rangle,\ \mathbb{I}_{a/x}\left({}^{e_{s..s'}}\!/\!_{f_{v'..v}}\right)}{A_{s,t,u,v}}\ {\scriptstyle 0\le s\le s',\ v'\le v\le|\mathbf{f}|},$$

$$\frac{B_{s,t',u',v},\ B \to \langle A\ {}^{a}\!/\!_{x}\rangle,\ \mathbb{I}_{a/x}\left({}^{e_{t'..t}}\!/\!_{f_{u..u'}}\right)}{A_{s,t,u,v}}\ {\scriptstyle t'\le t\le|\mathbf{e}|,\ 0\le u\le u'}$$

Figure 1: Deductive system describing an LITG parser.

reverse values in Goodman (1999), and outside probabilities in the inside-outside algorithm (Baker, 1979). In this paper we will call them contextual values, or β-values (since they are the second value we calculate).

The way to calculate the reverse values is to start with the goal node and work your way back to the axioms. The reverse value is calculated to be:

$$\beta(x) = \bigoplus_{\substack{n,i,b,a_1,\ldots,a_n \\ \text{such that} \\ \frac{a_1,\ldots,a_n}{b} \wedge x = a_i}} \beta(b) \otimes \bigotimes_{\{j \mid 1\le j\le n, j\ne i\}} \alpha(a_j)$$

That is: the reverse value of the consequence combined with the values of all sibling conditions is calculated and summed for all deductive rules where the item is a condition.

3.2 SPLITG

After we introduced stochastic preterminalized LITGs (Saers, 2011, SPLITG), the idea of expressing them in term of semiring parsing occurred. This is relatively straight forward, producing a compact set of deductive rules similar to that of LITGs. For LITGs, the items take the form of bispans labeled with a symbol. We will represent these bispans as $A_{s,t,u,v}$, where A is the label, and the two spans being labeled are $e_{s..t}$ and $f_{u..v}$. Since we usually do top-down parsing, the goal item is a virtual item ($\mathcal{G}$) than can only be reached by rewriting a nonterminal to the empty bistring (${}^{\epsilon}\!/\!_{\epsilon}$). Figure 1 shows the deductive rules for LITG parsing.

A preterminalized LITG promote preterminal symbols to a distinct class of symbols in the grammar, which is only allowed to rewrite into biterminals. Factoring out the terminal productions in this fashion allows the grammar to define one probability distribution over all the biterminals, which is useful for bilexica induction. It also means that the LITG rules that produce biterminals have to be replaced by two rules in a PLITG, resulting in the deductive rules in Figure 2.

4 Weighted hypergraphs

A hypergraph is a graph where the nodes are connected with *hyperedges*. A hyperedge is an edge that can connect several nodes with one node—it has

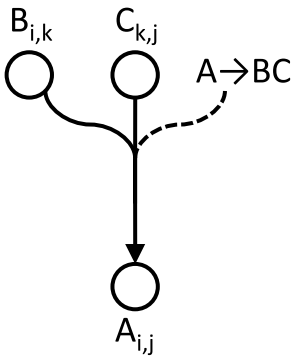

Figure 3: A weighted hyperedge between three nodes, based on the rule $A \rightarrow BC$. The tip of the arrow points to the head of the edge, and the two ends are the tails. The dashed line idicates where the weight of the edge comes from.

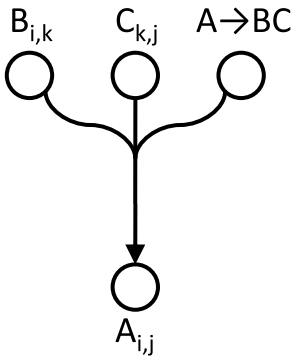

Figure 4: The same hyperedge as in Figure 3, where the rule has been promoted to first-class citizen. The hyperedge is no longer weighted.

one head, but may have any number of tails. Intuitively, this is a good match to context-free grammars, since each rule connects one symbol on the left hand side (the head of the hyperedge) with any number of symbols on the right hand side (the tails of the hyperedge). During parsing, one node is constructed for each labeled (bi-) span, and the nodes are connected with hyperedges based on the valid applications of rules. A hyperedge will be represented as $[h : t_1, \ldots, t_n]$ where h is the head and t_i are the tails.

When this is applied to weighted grammar, each hyperedge can be associated with a weight, making the hypergraph weighted. Every time an edge is traversed, its weight is combined with the value travelling through the edge. Weights are assigned to hyperedges via a weighting function $w(\cdot)$.

Figure 3 contains an illustration of a weighted hyperedge. The arrow indicates the edge itself, whereas the dotted line indicates where the weight comes from. Since each hyperedge corresponds to exactly one rule from a stochastic context-free grammar, we can use the inside-outside algorithm (Baker, 1979) to calculate inside and outside probabilities as well as to reestimate the probabilities of the rules. What we cannot easily do, however, is to change the parsing algorithm or grammar formalism.

If the weighted hyperedge approach was a one-to-one mapping to the semiring parsing approach, we could, but it is not. The main difference is that rules are part of the object level in semiring parsing, but part of the meta level in weighted hypergraphs. To address this disparity, we will reify the rules in the weighted hypergraph to make them nodes. Figure 4 shows the same hyperedge as Figure 3, but with the rule as a proper node rather than a weight associated with the hyperedge. These hyperedges are agnostic to what the tail nodes represent, so we can no longer use the inside-outside algorithm to reestimate the rule probabilities. We can, however, still calculate inside probabilities. In the weighted hyperedge approach, the inside probability of a node is:

$$\alpha(p) = \bigoplus_{\substack{n,q_1,\ldots,q_n \\ \text{such that} \\ [p:q_1,\ldots q_n]}} w([p : q_1, \ldots, q_n]) \otimes \bigotimes_{i=1}^{n} \alpha(q_i)$$

Whereas with the rules reified, the weight simply moved into the tail product:

$$\alpha(p) \bigoplus_{\substack{n,q_1,\ldots,q_n \\ \text{such that} \\ [p:q_1,\ldots q_n]}} \bigotimes_{i=1}^{n} \alpha(q_i)$$

By virtue of the deductive system used to build the hypergraph, we also have the reverse values, which correspond to outside probability:

$$\beta(x) = \bigoplus_{\substack{i,p,n,q_1,\ldots,q_n \\ \text{such that} \\ [p:q_1,\ldots q_n] \wedge x=q_i}} \beta(p) \otimes \bigotimes_{\{j|1\leq j\leq n, j\neq i\}} \alpha(q_j)$$

This means that we have the inside and outside probabilities of the nodes, and we could shoe-horn it into the reestimation part of the inside-outside algorithm.

It also means that we have β-values for the rules, which we are calculating as a side-effect of moving them into the object level. In Section 5, we will take a closer look at the semantics of the contextual probabilities that we are in fact calculating for the reified rules, and see how they can be used in reestimation of the rules.

4.1 SPLITG

Using the hypergraph parsing framework for SPLITGs turns out to be non-trivial. Where the standard LITG uses one rule to rewrite a nonterminal into another nonterminal and a biterminal, the SPLITG rewrites a nonterminal to a preterminal and a nonterminal, *and* rewrites the preterminal into a biterminal. This causes problems within the hypergraph framework, where each rule application should correspond to one hyperedge. As it stands we have two options:

1. Let each rule correspond to one hyperedge, which means that we need to introduce preterminal nodes into the hypergraph. This has a clear drawback for bracketing grammars,[1] since it is now necessary to keep different symbols apart. It also produces larger hypergraphs, since the number of nodes is inflated.

2. Let hypergraphs be associated with one or two rules, which means that we need to redefine hyperedges so that there are two different weighting functions: one for the nonterminal weight and one for the preterminal weight. Although all hyperedges are associated with one nonterminal rule, some hyperedges are not associated with any preterminal rule, making the preterminal weighting function partly defined.

Both of these approaches work in practice, but neither is completely satisfactory since they both represent work-arounds to shoe-horn the parsing algorithm (as stated in the deductive system) into a formalism that is not completely compatible. By reifying the rules into the object level, we rid ourselves of this inconvenience, as we no longer differentiate between different types of conditions.

[1] A bracketing grammar is a grammar where $|N| = 1$.

5 Reestimation of reified rules

As has been amply hinted at, the contextual probabilities (outside probabilities, reverse values or β-values) contain all new information we need about the rules to reestimate their probability in an expectation maximization (Dempster et al., 1977) framework. To show that this is indeed the case, we will rewrite the reestimation formulas of the inside-outside algorithm (Baker, 1979) so that they are stated in terms of contextual probability for the rules.

In general, a stochastic context-free grammar can be estimated from examples of trees generated by the grammar by means of relative frequency. This is also true for expectation maximization with the caveat that we have multiple hypotheses over each sentence (pair), and therefore calculate expectations rather than discrete frequency counts. We thus compute the updated parameterization function $\hat{\theta}$ based on expectations from the current parameterization function:

$$\hat{\theta}\left(\varphi|p\right) = \frac{E_\theta\left[p \rightarrow \varphi\right]}{E_\theta\left[p\right]}$$

Where $p \in N$ and $\varphi \in \{\Sigma \cup N\}^+$ (or $\varphi \in \{(\Sigma^* \times \Delta^*) \cup N\}^+$ for transduction grammars). The expectations are calculated from the sentences in a corpus $\mathcal{C}$:

$$E_\theta\left[x\right] = \sum_{\mathbf{w} \in \mathcal{C}} E_\theta\left[x|\mathbf{w}\right]$$

The exact way of calculating the expectation on x given a sentence depends on what x is. For nonterminal symbols, the expectations are given by:

$$\begin{aligned} E_\theta\left[p|\mathbf{w}\right] &= \frac{E_\theta\left[p, \mathbf{w}\right]}{E_\theta\left[\mathbf{w}\right]} \\ &= \frac{\sum_{0 \leq i \leq j \leq |\mathbf{w}|} \Pr\left(p_{i,j}, \mathbf{w}|G\right)}{\Pr\left(\mathbf{w}|G\right)} \\ &= \frac{\sum_{0 \leq i \leq j \leq |\mathbf{w}|} \alpha(p_{i,j})\beta(p_{i,j})}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})} \end{aligned}$$

For nonterminal rules, the expectations are shown in Figure 5. The most noteworthy step is the last one, where we use the fact that the summation is over the equivalence of the rule's reverse value. Each

$$
\begin{aligned}
E_\theta\left[p \to qr|\mathbf{w}\right] &= \frac{E_\theta\left[p \to qr, \mathbf{w}\right]}{E_\theta\left[\mathbf{w}\right]} \\
&= \frac{\sum_{0\le i\le k\le j\le|\mathbf{w}|} \Pr\left(w_{0..i}, p_{i,j}, w_{j..|\mathbf{w}|}\middle|G\right) \Pr\left(w_{i..k}|q_{i,k}, G\right) \Pr\left(w_{k..j}|r_{k,j}, G\right) \theta\left(qr|p\right)}{\Pr\left(\mathbf{w}|G\right)} \\
&= \frac{\sum_{0\le i\le k\le j\le|\mathbf{w}|} \beta(p_{i,j})\alpha(q_{i,k})\alpha(r_{k,j})\theta\left(qr|p\right)}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})} \\
&= \frac{\theta\left(qr|p\right)\sum_{0\le i\le k\le j\le|\mathbf{w}|} \beta(p_{i,j})\alpha(q_{i,k})\alpha(r_{k,j})}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})} = \boxed{\frac{\alpha(p \to qr)\beta(p \to qr)}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})}}
\end{aligned}
$$

Figure 5: Expected values for nonterminal rules in a specific sentence.

$$
\begin{aligned}
E_\theta\left[p \to a|\mathbf{w}\right] &= \frac{E_\theta\left[p \to a, \mathbf{w}\right]}{E_\theta\left[\mathbf{w}\right]} \\
&= \frac{\sum_{0\le i\le j\le|\mathbf{w}|} \Pr\left(w_{0..i}, p_{i,j}, w_{j..|\mathbf{w}|}\middle|G\right) \mathbb{I}_a(w_{i..j})\theta\left(a|p\right)}{\Pr\left(\mathbf{w}|G\right)} \\
&= \frac{\sum_{0\le i\le j\le|\mathbf{w}|} \beta(p_{i,j})\mathbb{I}_a(w_{i..j})\theta\left(a|p\right)}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})} \\
&= \frac{\theta\left(a|p\right)\sum_{0\le i\le j\le|\mathbf{w}|} \beta(p_{i,j})\mathbb{I}_a(w_{i..j})}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})} = \boxed{\frac{\alpha(p \to a)\beta(p \to a)}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})}}
\end{aligned}
$$

Figure 6: Expected values of terminal rules in a specific sentence.

$\beta(p_{i,j})\alpha(q_{i,k})\alpha(r_{k,j})$ term of the summation corresponds to one instance where the rule was used in the parse. Furthermore, the β value is the outside probability of the consequence of the deductive rule applied, and the two α values are the inside probabilities of the sibling conditions of that deductive rule. The entire summation thus corresponds to our definition of the reverse value of a rule, or its outside probability.

In Figure 6, the same process is carried out for terminal rules. Again, the summation is over all possible ways that we can combine the inside probability of the sibling conditions of the rule with the outside probability of the consequence.

Since the expected values of both terminal and nonterminal rules have the same form, we can generalize the formula for any production φ:

$$
E_\theta\left[p \to \varphi|\mathbf{w}\right] = \frac{\alpha(p \to \varphi)\beta(p \to \varphi)}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})}
$$

Finally, plugging it all into the original rule estimation formula, we have:

$$
\begin{aligned}
\hat{\theta}\left(\varphi|p\right) &= \frac{E_\theta\left[p \to \varphi\right]}{E_\theta\left[p\right]} \\
&= \frac{\sum\limits_{\mathbf{w}\in\mathcal{C}} \frac{\alpha(p\to\varphi)\beta(p\to\varphi)}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})}}{\sum\limits_{\mathbf{w}\in\mathcal{C}} \frac{\sum\limits_{0\le i\le j\le|\mathbf{w}|} \alpha(p_{i,j})\beta(p_{i,j})}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})}} \\
&= \alpha(p \to \varphi)\frac{\sum\limits_{\mathbf{w}\in\mathcal{C}} \frac{\beta(p\to\varphi)}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})}}{\sum\limits_{\mathbf{w}\in\mathcal{C}} \frac{\sum\limits_{0\le i\le j\le|\mathbf{w}|} \alpha(p_{i,j})\beta(p_{i,j})}{\alpha(S_{0,|\mathbf{w}|})\beta(S_{0,|\mathbf{w}|})}}
\end{aligned}
$$

Rather than keeping track of the expectations of nonterminals, they can be calculated from the rule expectations by marginalizing the productions:

$$
E_\theta\left[p\right] = \sum_\varphi E_\theta\left[p \to \varphi\right]
$$

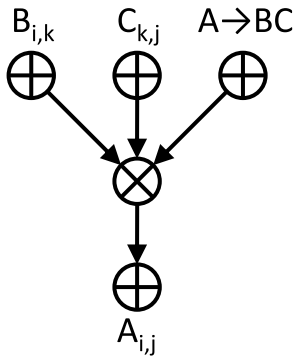

Figure 7: The same hyperedge as in Figures 3 and 4, represented as a mul/add-subgraph.

5.1 SPLITG

Since this view of EM and parsing generalizes to deductive systems with multiple rules as conditions, we can apply it to the deductive system of SPLITGs. It is, however, also interesting to note how the hypergraph view of parsing is changed by this. We effectively removed the weights from the edges, but kept the feature that values of nodes depend entirely on the values connected by incoming hyperedges. If we assume the values to be from the Boolean semiring, the hypergraphs we ended up with are in fact *and/or-graphs*. That is: each node in the hypergraph corresponds to an *or-node*, and each hyperedge corresponds to an *and-node*. We note that this can be generalized to any semiring, since *or* is equivalent to $\oplus$ and *and* is equivalent to $\otimes$ for the Boolean semiring, we can express a hypergraph over an arbitrary semiring as a *mul/add-graph*.[2] Figure 7 shows how a hyperedge looks in this new graph form. The α-value of a node is calculated by combining the values of all incoming edges using the operator of the node. The β-values are also calculated using the operator of the node, but with the edges reversed. For this to work properly, the *mul-nodes* need to behave somewhat different from *add-nodes*: each incoming edge has to be reversed one at a time, as illustrated in Figure 8.

6 Conclusions

We have shown that the reification of rules into the parse forest graphs allows for a unified framework where all calculations are performed the same way,

[2]Because it is much easier to pronounce than $\otimes/\oplus$-graph.

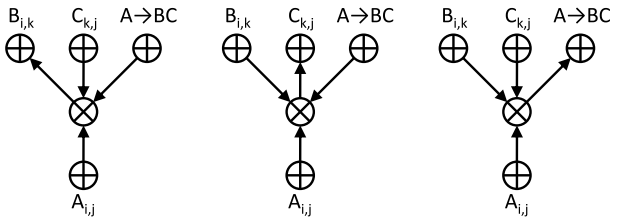

Figure 8: Reverse values (β) are calculated by tracking backwards through all possible paths. This produces three different paths for the mul/add-subgraph from Figure 7. Arrows pointing downward propagate α-values while arrows pointing upward propagate β-values.

and where the calculations for the rules encompass all information needed to reestimate them using expectation maximization. The contextual probability of a rule—its outside probability—holds all information needed to calculate expectations, which can be exploited by promoting the rules to first-class citizens of the parse forest. We have also seen how this reification of the rules helped solve a real translation problem—induction of stochastic preterminalized linear inversion transduction grammars using expectation maximization.

Acknowledgments

This work was funded by the Defense Advanced Research Projects Agency (DARPA) under GALE Contract Nos. HR0011-06-C-0023 and HR0011-06-C-0023, and the Hong Kong Research Grants Council (RGC) under research grants GRF621008, GRF612806, DAG03/04.EG09, RGC6256/00E, and RGC6083/99E. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency. We would also like to thank the three anonymous reviewers, whose feedback made this a better paper.

References

James K. Baker. 1979. Trainable grammars for speech recognition. In *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, pages 547–550, Cambridge, Massachusetts.

Sylvie Billot and Bernard Lang. 1989. The structure of shared forests in ambiguous parsing. In *Proceedings*

of the 27th annual meeting on Association for Computational Linguistics, ACL'89, pages 143–151, Stroudsburg, Pennsylvania, USA.

John Cocke. 1969. *Programming languages and their compilers: Preliminary notes*. Courant Institute of Mathematical Sciences, New York University.

Arthur Pentland Dempster, Nan M. Laird, and Donald Bruce Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Jason Eisner. 2001. Expectation semirings: Flexible EM for finite-state transducers. In Gertjan van Noord, editor, *Proceedings of the ESSLLI Workshop on Finite-State Methods in Natural Language Processing (FSMNLP)*. Extended abstract (5 pages).

Jason Eisner. 2002. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1–8, Philadelphia, July.

Joshua Goodman. 1999. Semiring parsing. *Computational Linguistics*, 25(4):573–605.

Liang Huang. 2008. *Forest-based Algorithms in Natural Language Processing*. Ph.D. thesis, University of Pennsylvania.

Tadao Kasami and Koji Torii. 1969. A syntax-analysis procedure for unambiguous context-free grammars. *Journal of the Association for Computing Machinery*, 16(3):423–431.

Zhifei Li and Jason Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 40–51, Singapore, August.

Christopher D. Manning and Dan Klein. 2001. Parsing and hypergraphs. In *Proceedings of the 2001 International Workshop on Parsing Technologies*.

Markus Saers and Dekai Wu. 2011. Principled induction of phrasal bilexica. In *Proceedings of the 15th Annual Conference of the European Association for Machine Translation*, Leuven, Belgium, May.

Markus Saers. 2011. *Translation as Linear Transduction: Models and Algorithms for Efficient Learning in Statistical Machine Translation*. Ph.D. thesis, Uppsala University, Department of Linguistics and Philology.

Daniel H. Younger. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208.