Term Graphs and the NP-completeness of the Product-Free Lambek Calculus

Timothy A. D. Fowler
tfowler@cs.toronto.edu

University of Toronto

Abstract. We provide a graphical representation of proofs in the product-free Lambek calculus, called term graphs, that is related to several other proof net presentations. The advantage of term graphs is that they are very simple compared to the others. We use this advantage to provide an NP-completeness proof of the product-free Lambek Calculus that uses the reduction of [8]. Our proof is more intuitive due to the fact that term graphs allow arguments that are graphical in nature rather than using the algebraic arguments of [8].

1 Introduction

The Lambek calculus [4], is a variant of categorial grammar that is of interest to computational linguists because of its ability to capture a wide range of the semantics of natural language. In this paper we will only be concerned with the product-free fragment because of its simplicity and the paucity of linguistic uses of the product connective. The product-free Lambek calculus has a number of interesting computational properties. First, it is weakly equivalent to context-free grammars [6] but not strongly equivalent [9]. Second, it has recently been proven to have an NP-complete sequent derivability problem [8]. Finally, recent research has shown that as long as the order of categories is bounded, polynomial time parsing is possible [2, 3].

With these latter two results, we have precisely determined where the intractability of the Lambek calculus lies, but the NP-completeness proof of [8] and the polynomial time algorithm of [3] seem to use entirely different methods for proving correctness. The former uses a primarily algebraic approach whereas the latter uses a graphical approach based on the graphical LC-Graphs of [5].

The purpose of this paper will be to introduce a new representation of proof nets, called *term graphs*, that are similar to the LC-Graphs of [5] but simpler in that they require two less correctness conditions and they avoid the introduction of terms from the lambda calculus. Term graphs are important because they bridge the gap between the methods of [8] and [3]. That is, despite being superficially very different from the structures of [8], they are fundamentally quite similar, as we will see. In addition, the *abstract term graphs* of [3] are an abstraction over graphical structures that are essentially identical to our term graphs. Once we have introduced term graphs, we will use them to provide a proof of the NP-completeness of the sequent derivability problem for the product-free Lambek calculus that is more intuitive than that of [8] giving us insight into his reduction. This NP-completeness proof also allows us to consider these two results in the same language which will help future research in the area.

This paper will proceed as follows: In section 2, we will introduce the Lambek calculus and term graphs and prove the correctness of term graphs. Then, in section 3 we will introduce the polynomial reduction of [8] and provide an NP-completeness proof that is graphical in nature.

2 The Lambek Calculus and Term Graphs

In this section, we introduce the Lambek calculus in its sequent presentation and then quickly ignore the sequent presentation in favour of term graphs. Term graphs, like other proof net methods, allow easier analysis of computational problems pertaining to these sorts of logics. We prove the correctness of term graphs via the LC-graphs of [5].

2.1 The Lambek Calculus

The sequent derivability problem for the Lambek calculus takes as input a sequent made up of *categories*. The categories for the product-free fragment are built up from a set of *atoms* and the two binary connectives $/$ and $\backslash$. A sequent is a sequence of categories known as the antecedent together with the $\vdash$ symbol and one additional category called the succedent. The sequent derivability problem asks whether an input sequent is logically derivable from the axioms and rules shown in figure 1.

We will be considering two closely related variants of the Lambek calculus: The original Lambek calculus (L) and the Lambek calculus allowing empty premises (L^*). A sequent is *derivable in* L^* if and only if it has a proof according to the sequent calculus in figure 1. In addition, we say that the sequent is derivable in L if and only if it is derivable in L^* such that Γ is non-empty when applying the rules $\backslash R$ and $/R$.

$$\frac{}{\alpha \vdash \alpha}$$

$$\frac{\Gamma \vdash \alpha \qquad \Delta\beta\Theta \vdash \gamma}{\Delta\Gamma\alpha\backslash\beta\Theta \vdash \gamma}\,\backslash L \quad \frac{\alpha\Gamma \vdash \beta}{\Gamma \vdash \alpha\backslash\beta}\,\backslash R$$

$$\frac{\Gamma \vdash \alpha \qquad \Delta\beta\Theta \vdash \gamma}{\Delta\beta/\alpha\Gamma\Theta \vdash \gamma}\,/L \quad \frac{\Gamma\alpha \vdash \beta}{\Gamma \vdash \beta/\alpha}\,/R$$

Fig. 1. Axioms and rules of the Lambek calculus (from [4]).

In figure 1, lowercase Greek letters represent categories and uppercase Greek letters represent sequences of categories.

2.2 Term Graphs

In this section, we will introduce *term graphs*[1] which are a simplification of the LC-Graphs of [5], which in turn are based on the proof nets of [7]. The advantage of

[1] We call them term graphs because they are a graphical representation of the semantic term.

term graphs over LC-Graphs is that we have only two correctness conditions instead of four and the fact that we avoid the introduction of lambda terms. Furthermore, both of the term graph correctness conditions are conditions on the existence of certain paths whereas the LC-Graph correctness conditions are conditions on the existence *and* absence of certain paths.

Definition 1. *A* term graph *for a sequent is a directed graph whose vertices are category occurrences and whose edges are introduced in four groups. Like other proof net presentations, we will proceed with a deterministic step first and a non-deterministic step second.*

First, we assign polarities to category occurrences by assigning negative polarity to occurrences in the antecedent and positive polarity to the succedent. Then, the first two groups of edges are introduced by decomposing the category occurrences via the following vertex rewrite rules:

$$(\alpha/\beta)^- \Rightarrow \alpha^- \rightarrow \beta^+ \tag{1}$$
$$(\beta\backslash\alpha)^- \Rightarrow \beta^+ \leftarrow \alpha^- \tag{2}$$
$$(\alpha/\beta)^+ \Rightarrow \beta^- \dashleftarrow \alpha^+ \tag{3}$$
$$(\beta\backslash\alpha)^+ \Rightarrow \alpha^+ \dashrightarrow \beta^- \tag{4}$$

Each vertex rewrite rule specifies how to rewrite a single vertex on the left side to two vertices on the right side. The neighbourhood of the vertex on the left side of each rule is assigned to α on the right side. Dashed edges are referred to as Lambek edges *and non-dashed edges are referred to as* regular edges*. These two groups of edges will also be referred to as* rewrite *edges.*

After decomposition via the rewrite rules, we have an ordered set of polarized vertices, with edges between some of them. We say that a vertex belongs *to a category occurrence in the sequent if there is a chain of rewrites going back from the one that introduced this vertex to the one that rewrote the category occurrence.*

Lemma 1. *After decomposition via the rewrite rules has terminated, there is a unique vertex with in-degree 0 belonging to each category occurrence in the sequent.*

Proof. *By induction over the rewrite rules.*

A third group of edges is introduced such that there is one Lambek edge from the unique vertex with in-degree 0 in the succedent to each unique vertex with in-degree 0 in each of the antecedent category occurrences. These edges are referred to as rooted Lambek edges*. This completes the deterministic portion of term graph formation.*

A matching *is a planar matching of these vertices in the half plane where atom occurrences are* matched *to atoms occurrences with the same atom but with opposite polarity. The fourth group of edges are introduced as regular edges from the positive vertices to the negative vertices they are matched to. If α and β are matched in a matching M then we write $M(\alpha, \beta)$. See figures 2 and 3 for an example.*

The two edge types of a term graph induce two distinct graphs, the *regular term graph* and the *Lambek term graph*. We will prefix the usual graph theory terms with *regular* and *Lambek* to distinguish paths and edges in these graphs.

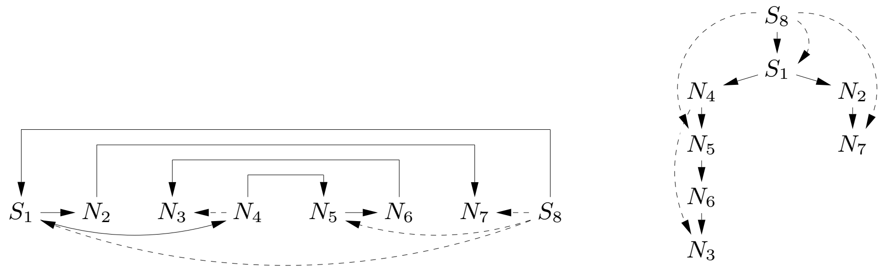

Fig. 2. Two depictions of an integral term graph for the sequent $(S/N)/(N/N), N/N, N \vdash S$.

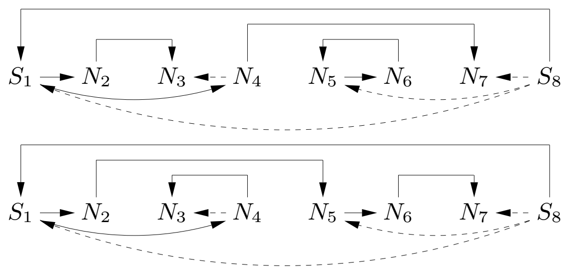

Fig. 3. An L^*-integral term graph that is not integral (top) and a term graph that is not L^*-integral (bottom) for the sequent $(S/N)/(N/N), N/N, N \vdash S$.

Lemma 2. *In a term graph, there is a unique vertex with in-degree* 0.

Proof. By lemma 1, each category in the antecedent and succedent has a unique vertex with in-degree 0 before the introduction of the rooted Lambek edges. However, after the introduction of those edges, the only vertex with in-degree 0 is the one in the succedent.

Definition 2. *In a term graph, the unique vertex of in-degree* 0 *is denoted by* τ.

Lemma 3. *The vertices in a term graph have the following restrictions on incident edges:*

	Negative vertices	*Positive vertices* $\neq \tau$	τ
regular in-degree	*1*	*1*	*0*
regular out-degree	*Arbitrary*	*1*	*1*
Lambek in-degree	*1*	*0*	*0*
Lambek out-degree	*0*	*Arbitrary*	*Arbitrary*

Proof. By induction on the term graph formation process.

Because of this result, we can determine the polarity of a vertex by its incident edges. Therefore, we will often simplify our diagrams by omitting polarities.

Definition 3. *We define two conditions on term graphs:*

T*: For all Lambek edges* $\langle s, t\rangle$ *there is a regular path from* s *to* t.
T(CT)*: For each Lambek edge* $\langle s, t\rangle$*, there exists a negative vertex* x *such that there is a regular path from* s *to* x *and there is no rewrite Lambek edge* $\langle s', x\rangle$ *such that there is a regular path from* s *to* s'.[2].

[2] T(CT) requires that we distinguish rewrite Lambek edges from rooted Lambek edges in the representation. To avoid clutter, we will not mark this difference in our figures.

A matching and its corresponding term graph are L^*-integral *iff they satisfy* ***T***. *A matching and its corresponding term graph are* integral *iff they satisfy* ***T*** *and* ***T****(CT).*

A partial matching is a matching which matches a subset of the polarized vertices and a partial term graph is the term graph of a partial matching. We will extend the notions of integrity to partial matchings by requiring that the integrity conditions are true of Lambek edges whose source and target are matched in the matching. Then, we can prove that the union of two integral partial matchings is an integral partial matching by considering those Lambek edges with a source matched by one matching and a target matched by the other do not violate the integrity conditions.

We will prove that integrity corresponds to sequent derivability in section 2.3. As discussed in the introduction, there are a number of connections between term graphs and the structures of [8]. For example:

- The set of negative vertices in a term graph corresponds to the set $\mathcal{N}_{\mathbb{W}}$ of [8].
- A matched edge from s to t in a term graph corresponds to $\pi(t) = s$ for $t \in \mathcal{N}_{\mathbb{W}}$ of [8]
- A rewrite edge from s to t in a term graph corresponds to $\varphi(t) = s$ of [8].
- A regular edge from s to t in a term graph corresponds to $\psi(t) = s$ of [8].
- The requirement that that matchings be planar and be between like atoms of opposite polarity correspond to the first three correctness conditions of [8]. **T** corresponds to the fourth correctness condition and **T**(CT) corresponds to the fifth.

2.3 Term Graph Correctness

We will prove the correctness of term graphs with respect to the Lambek calculus via the LC-Graphs of [5]. Since LC-Graphs and term graphs are constructed using similar algorithms, we will define LC-Graphs in terms of how they differ from term graphs rather than from scratch.

LC-Graphs are graphs whose vertex set V is a set of lambda calculus variables, which are introduced during the equivalent of the rewrite rule process. During this process, atom occurrences are associated with lambda terms. The leftmost variable in each lambda term is a unique identifier for the atom. This correspondance between lambda variables and atoms establishes a correspondence between LC-Graphs and term graphs. In addition to this superficial difference, LC-Graphs differ from term graphs structurally in the following three ways:

1. The lambda variables in an LC-Graph are locally rearranged relative to the corresponding atom occurrences in a term graph as seen in the mapping in figures 4 and 5.
2. LC-Graphs do not distinguish between Lambek edges and regular edges.
3. LC-Graphs do not introduce any equivalent to the rooted Lambek edges.

The first of these three differences is required for term graphs to avoid the introduction of lambda terms. The second difference allows us to express our correctness conditions more concisely and simplifies the presentation of our proofs. This is accomplished by no longer needing to identify the edges in an LC-Graph that are the

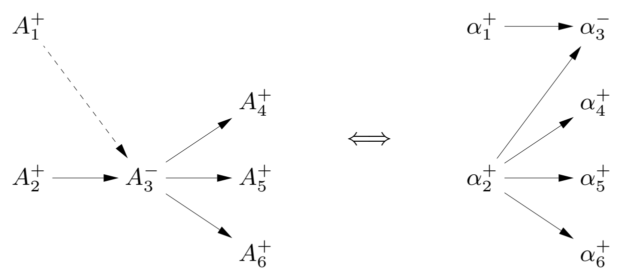

Fig. 4. The mapping between term graphs and LC-Graphs for neighbourhoods of negative vertices. The lambda variable α_i corresponds to the atom occurrence A_i for $1 \leq i \leq 6$.

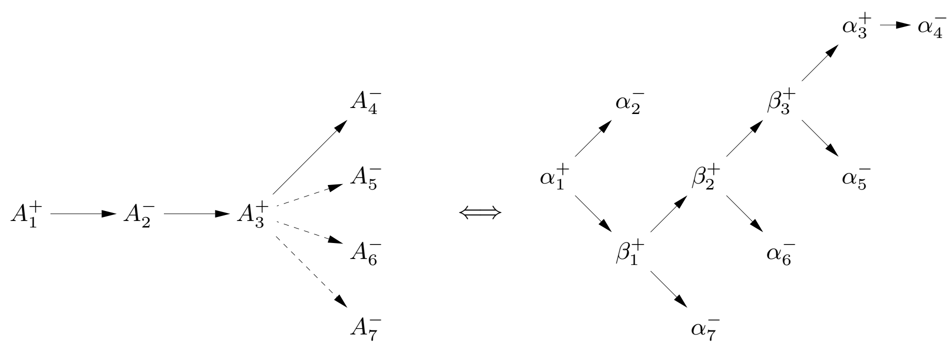

Fig. 5. The mapping between term graphs and LC-Graphs for neighbourhoods of positive vertices. The positive vertex A_3 in a term graph is represented by the vertices $\beta_1, \beta_2, \beta_3$ and α_3 in an LC-Graph. The lambda variable α_i corresponds to the atom occurrence A_i for $1 \leq i \leq 7$.

equivalent of Lambek edges by their endpoint vertices. The last difference will allow us to eliminate one of the correctness conditions.

[5] defines the following terms, necessary to understand the correctness conditions:

Definition 4. *A* lambda-node *is a positive vertex in an LC-Graph with two regular out-neighbours, one of which is positive and one of which is negative. The positive one is its* plus-daughter *and the negative one is its* minus-daughter.

For example, B_1, B_2 and B_3 in figure 5 are lambda-nodes. The intuition is that they correspond to Lambek edges. We can now define the correctness conditions on LC-Graphs and state theorem 1 (proven in [5]):

I(1): There is a unique node in G with in-degree 0, from which all other nodes are path-accessible.
I(2): G is acyclic.
I(3): For every lambda-node $v \in V$, there is a path from its plus-daughter u to its minus-daughter w
I(CT): For every lambda-node $v \in V$, there is a path in G, $v \rightsquigarrow x$ where x is a terminal node and there is no lambda-node $v' \in V$ such that $v \rightsquigarrow v' \rightarrow x$.

Theorem 1. *A sequent is derivable in L^* iff it has an LC-Graph satisfying conditions **I**(1-3). A sequent is derivable in L iff it has an LC-Graph satisfying conditions **I**(1-3) and **I**(CT).*

The equivalence between the **I** conditions and the **T** conditions begins with the following two lemmas.

Lemma 4. *In a term graph, the rewrite edges and the rooted Lambek edges form a tree with τ as its root.*

Proof. By induction on the rewrite rules together with the way that the rooted Lambek edges are introduced.

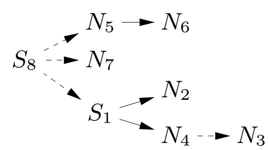

Fig. 6. The tree of rewrite and rooted Lambek edges for $(S/N)/(N/N), N/N, N \vdash S$

Lemma 5. *In an L^*-integral term graph, if there is a path from a to b, then there is a regular path from a to b.*

Proof. The path from a to b may contain a Lambek edge $\langle s, t \rangle$, but by **T**, there is a regular path from s to t. Replacing the Lambek edge with the regular path and repeating gives us a regular path from a to b.

We next define a new condition on term graphs that is a counterpart to **I**(2).

T(C): The term graph is acyclic.

We now prove that any L^*-integral term graph is necessarily acyclic.

Proposition 1. ***T*** $\Rightarrow$ ***T****(C)*

Proof. Suppose there is a cycle. Then, by lemma 5, there is a regular cycle. That cycle cannot contain τ since τ has no in-edges. However, by lemma 4, there is a path from τ to every vertex and by lemma 5 that path is regular. But, by lemma 3 all vertices have regular in-degree at most one. Therefore, no cycles can exist.

Proposition 2. ***T*** $\Rightarrow$ ***I****(1)*

Proof. By lemma 4, there is a path from τ to every node in the term graph and by lemma 5 there is a regular path from τ to every node. Then, by the mapping of term graphs to LC-Graphs, these paths have equivalents in the LC-Graph.

Proposition 3. ***T*** $\Rightarrow$ ***I****(2)*

Proof. By proposition 1, **T** $\Rightarrow$ **T**(C) and by inspection of the mapping from term graphs to LC-Graphs, no new cycles can be introduced.

Proposition 4. ***T*** $\Rightarrow$ ***I****(3)*

Proof. By the mapping.

Proposition 5. ***I****(1),* ***I****(3)* $\Rightarrow$ ***T***

Proof. **I**(3) requires that all nodes be accessible from the root, which means that for the rooted Lambek edges $\langle \tau, t \rangle$, there is a regular path from τ to t. Then, because of the way that positive vertices in a term graph are mapped from their equivalents in LC-Graphs, enforcing **I**(3) requires that for rewrite Lambek edges $\langle s, t \rangle$, there be a regular path from s to t.

Proposition 6. ***T****(CT)* $\Leftrightarrow$ ***I****(CT)*

Proof. The only point of interest is the fact that the Lambek edges whose source is τ could rule out some term graphs when are provable in L^*. However, such Lambek edges are specifically ruled out by **T**(CT).

Theorem 2. *A sequent is derivable in L iff it has a term graph satisfying **T**. A sequent is derivable in L^* iff it has a term graph satisfying conditions **T** and **T**(CT).*

3 NP-Completeness Proof

Now that we have defined a simple graphical representation of proofs in the Lambek calculus, we can proceed with a graphical proof of NP-Completeness that in some ways mirrors the proof of [8]. We begin with the same reduction from SAT as [8].

Definition 5. *Let $c_1 \wedge \ldots \wedge c_m$ be a SAT instance with variables $x_1, \ldots, x_n$. We define the sequent Σ as in figure 7 (from [8]).*

The space of matchings is analyzed via two partial matchings M_t and N_t based on a truth assignment t (where truth assignments are sequences of booleans). First, we prove that the partial matching M_t is always integral and then we prove that the partial matching N_t is integral if and only if the SAT instance is satisfiable. Finally, in proposition 9, we prove that any L^*-integral matching must partition into two such partial matchings.

The atoms of Σ are p_i^j, q_i^j, a_i^j, b_i^j, c_i^j and d_i^j for $1 \leq i \leq n$ and $1 \leq j \leq m$.

Lemma 6. *X has $4m+2$ atoms for $X \in \{A_i, B_i, C_i, D_i, E_i(t), G, H_i\}$.*

Proof. By induction.

Definition 6. *For $X \in \{A_i, B_i, C_i, D_i, E_i(t),\ G, H_i\}$, X^+ is the leftmost $2m+1$ atoms and X^- is the rightmost $2m+1$ atoms. We refer to these as* hills.

$$
\begin{array}{ll}
A_i^0 = a_i^0 \backslash p_i^0 & C_i^0 = c_i^0 \backslash p_i^0 \\
A_i^j = (q_i^j / ((b_i^j \backslash a_i^j) \backslash A_i^{j-1})) \backslash p_i^j & C_i^j = (q_i^j / ((d_i^j \backslash c_i^j) \backslash C_i^{j-1})) \backslash p_i^j \\
A_i = A_i^m & C_i = C_i^m \\
B_i^0 = a_i^0 & D_i^0 = c_i^0 \\
B_i^j = q_{i-1}^j / (((b_i^j / B_i^{j-1}) \backslash a_i^j) \backslash p_{i-1}^{j-1}) & D_i^j = q_{i-1}^j / (((d_i^j / D_i^{j-1}) \backslash c_i^j) \backslash p_{i-1}^{j-1}) \\
B_i = B_i^m \backslash p_{i-1}^m & D_i = D_i^m \backslash p_{i-1}^m \\
G^0 = p_0^0 \backslash p_n^0 & H_i^0 = p_{i-1}^0 \backslash p_i^0 \\
G^j = (q_n^j / ((q_0^j \backslash p_0^j) \backslash G^{j-1}) \backslash p_n^j & H_i^j = ((q_{i-1}^j / (q_i^j / H_i^{j-1})) \backslash p_{i-1}^j) \backslash p_i^j \\
G = G^m & H_i = H_i^m
\end{array}
$$

$$
\begin{aligned}
E_i^0(t) &= p_{i-1}^0 \\
E_i^j(t) &= \begin{cases} q_i^j / (((q_{i-1}^j / E_i^{j-1}(t)) \backslash p_{i-1}^j) \backslash p_i^{j-1}) & \text{if } \neg_t x_i \text{ appears in } c_j \\ (q_{i-1}^j / (q_i^j / (E_i^{j-1} \backslash p_i^{j-1}))) \backslash p_{i-1}^j & \text{if } \neg_t x_i \text{ does not appear in } c_j \end{cases} \\
E_i(t) &= E_i^m(t) \backslash p_i^m
\end{aligned}
$$

$$
\begin{aligned}
\Pi_i &= E_i(0) / (B_i \backslash A_i^m), H_i^m, (D_i \backslash C_i^m) \backslash E_i(1) \\
\Sigma &= \Pi_1, \ldots, \Pi_n \vdash G^m
\end{aligned}
$$

Fig. 7. The sequent Σ for the SAT instance $c_1 \wedge \ldots \wedge c_n$. Note that $\neg_0 x = \neg x$ and $\neg_1 x = x$.

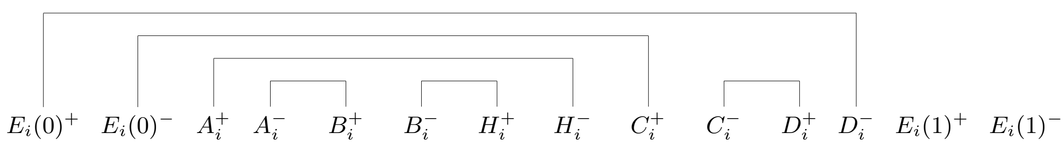

Fig. 8. M_t for Π_i and $t = \langle t_1, \ldots, t_{i-1}, 1, t_{i+1}, \ldots, t_n \rangle$

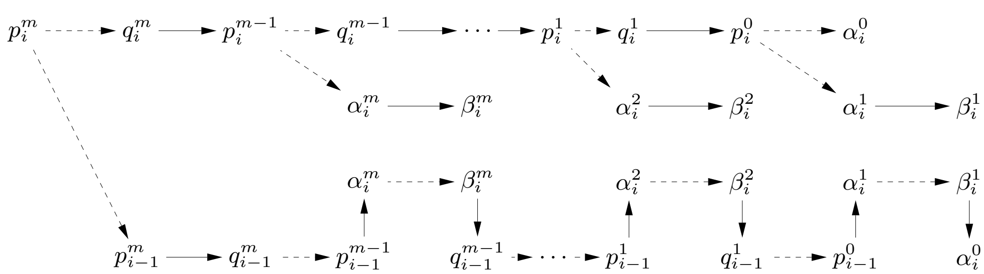

Fig. 9. Term graph for $A_i \backslash B_i$ (for $\alpha = a$ and $\beta = b$) and $C_i \backslash D_i$ (for $\alpha = c$ and $\beta = d$)

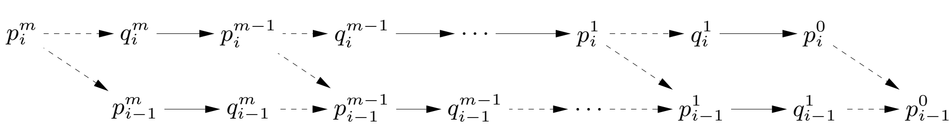

Fig. 10. Term graph for $(A_i \backslash B_i)^+$ where B_i^- has been matched to A_i^+ (or $(C_i \backslash D_i)^+$ where D_i^- has been matched to C_i^+) and all paths have been contracted. The angled edges other than the first are abstracted edges similar to those in [3].

Proposition 7. *Let $t = \langle t_1, \ldots, t_n\rangle$ be a truth assignment and let M_t be the following partial matching (depicted in figure 8). For $1 \leq i \leq n$, $M_t(B_i^+, A_i^-)$ and $M_t(D_i^+, C_i^-)$. If $t_i = 0$ then $M_t(E_i(1)^+, B_i^-)$, $M_t(A_i^+, E_i(1)^-)$, $M_t(H_i^+, D_i^-)$ and $M_t(C_i^+, H_i^-)$. If $t_i = 1$ then $M_t(E_i(0)^+, D_i^-)$, $M_t(C_i^+, E_i(0)^-)$, $M_t(H_i^+, B_i^-)$ and $M_t(A_i^+, H_i^-)$.*

Then, M_t is integral.

Proof. Figure 9 shows the partial term graph consisting of the vertices from $B_i\backslash A_i$ (or equivalently from $D_i\backslash C_i$ which are identical under renaming of α and β). However, by introducing the edges from M_t for A_i, B_i, C_i and D_i, contracting paths regular path longer than 1 and removing Lambek edges $\langle s, t\rangle$ which have a regular path from s to t, we get the abstraction of a term graph shown in figure 10. This abstraction over a term graph is essentially identical to a term graph except that the Lambek in-degree of negative vertices is now unbounded due to path contraction.

Then, regardless of whether $t_i = 0$ or $t_i = 1$, this partial abstract term graph is combined with the partial term graph for H_i (shown in figure 11) by inserting edges between identical atoms from the positive vertices to negative vertices. It can be seen that the combined term graph is L^*-integral by observing that each Lambek edge is in fact overlaid by a regular path.

In a parallel process, the abstract term graph for either $B_i\backslash A_i$ or $D_i\backslash C_i$ is combined with the term graph for $E_i(t_i)$, which is constructed out of components shown in figure 12. However, like the combination with H_i, the result is a term graph where all Lambek edges are forward edges despite the variation of $E_i(t_i)$ as exemplified in figure 13.

T(CT) is straightforward to check because for each Lambek edge $\langle s, t\rangle$ in these partial term graphs, the vertex s has a regular edge to a vertex x with a Lambek in-neighbour which does not have a regular path from s.

Therefore, M_t is integral.

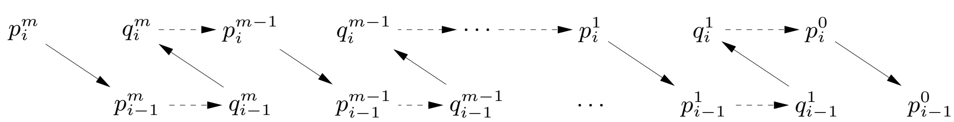

Fig. 11. Term graph for H_i

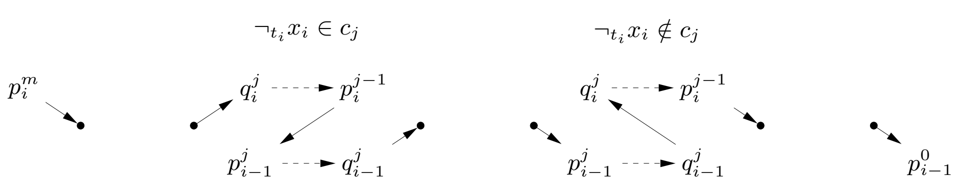

Fig. 12. Term graph components for $E_i(t_i)$. The round nodes are place holders used to indicate the source and target of some edges. They can be ignored once the term graph for $E_i(t_i)$ is complete.

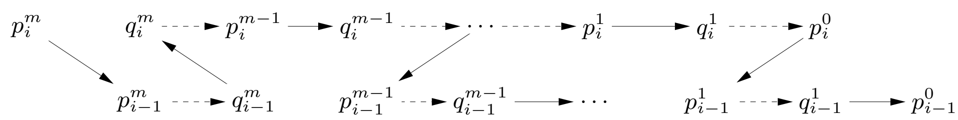

Fig. 13. Term graph for $E_2(0)$ for $(x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_1) \wedge (x_1) \wedge \ldots \wedge (x_2 \vee \neg x_1) \wedge (\neg x_2 \vee x_2))$.

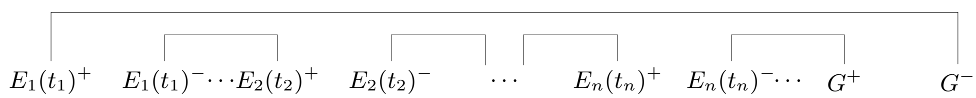

Fig. 14. N_t where $t = \langle t_1, t_2, \ldots, t_n \rangle$

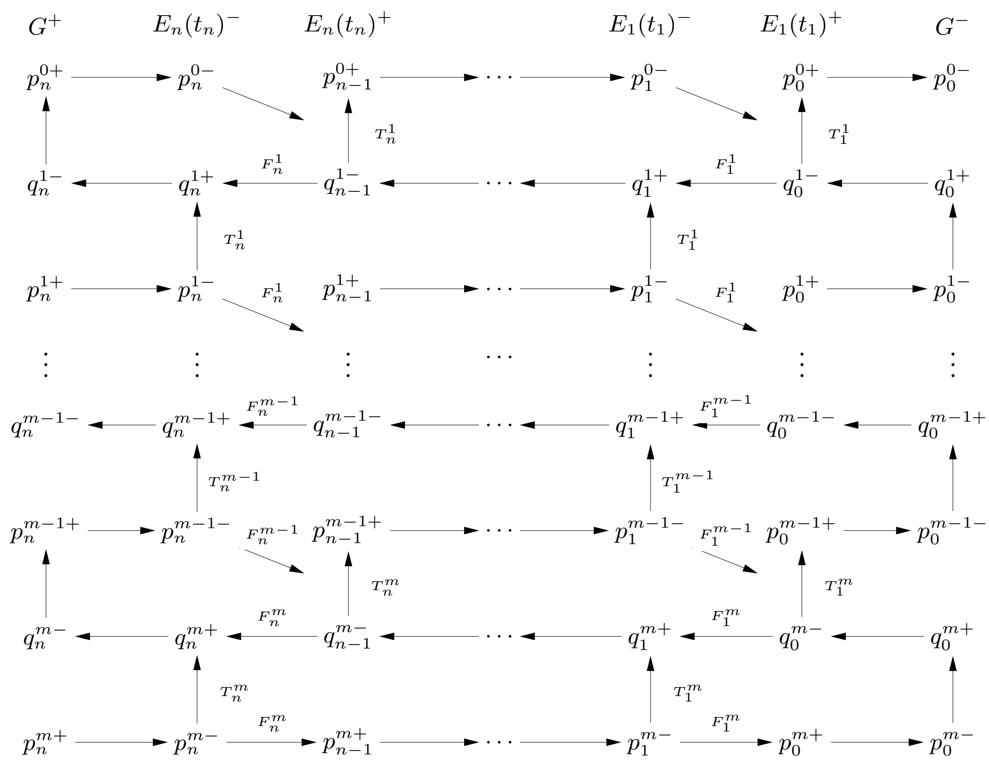

Fig. 15. The term graph N_t (with Lambek edges omitted). Each vertex in $\{p_i^{j-}, q_{i-1}^{j-}\}$ has two out-edges labelled T_i^j and F_i^j for $1 \leq j \leq m, 1 \leq i \leq n$. If $\neg_{t_i} x_i$ appears in c_j then we keep only the T_i^j edges and otherwise, we keep only the F_i^j edges. The angled edges, which appear to have no target, have a target which is dependant on whether $\neg_{t_k} x_k$ appears in c_l for $l < j$. Their target is always the lowest positive vertex which does not have another in-edge.

Proposition 8. *Let $t = \langle t_1, \ldots, t_n \rangle$ be a truth assignment and let N_t be the following partial matching (depicted in figure 14). For $1 < i \leq n$, $N_t(E_i(t_i)^+, E_{i-1}(t_{i-1})^-)$, $N_t(E_1(t_1)^+, G^-)$ and $N_t(G^+, E_n(t_n)^-)$. Then, N_t is integral iff N_t is L^*-integral iff t is a satisfying truth assignment for $c_1 \wedge \ldots \wedge c_m$.*

Proof. The relevant subgraph of the term graph for N_t is shown in figure 15 for the general case and in figure 16 for a specific example.

There are several types of Lambek pairs. Consider $\langle u, v \rangle$ such that $u, v \in E_i(t_i)$ for some i. Then, if $u = p^j_{i-1}$ and $v = q^j_{i-1}$ for $j \geq 1$, the path leaves u to the right but must eventually return to v since any rightward movement due to an F edge is mirrored by a leftward movement due its paired edge resulting in no **T** violation. If $u = q^j_i$ and $v = p^{j-1}_i$ for $j \geq 1$, either the link is completed immediately or it is completed via an angled edge whose target is v. The case where $u, v \in G$ where $u = p^j_n$ and $v = q^j_n$ is similar to the first case.

Next, consider $\langle u, v \rangle$ such that $u = p^{j-1}_n$ and $v = p^j_0$ for some $j \geq 1$. v is reached before u iff no edge T^j_i is present for $1 \leq i \leq n$ iff c_m does not contain $\neg_{t_i} x_i$ for any $1 \leq i \leq n$ iff $c_1 \wedge \ldots \wedge c_m$ is unsatisfiable.

Finally, consider the rooted Lambek edges $\langle u, v \rangle$ where $u = p^m_n$ and $v = p^m_i$ for some i. u is the leftmost atom in the bottom row and v appears somewhere to its right in the bottom row. Then, there must be a regular path from u to v since the angled edges target the lowest vertex without an in-edge.

Checking that **T**(CT) is never violated is straightforward via the same cases as above.

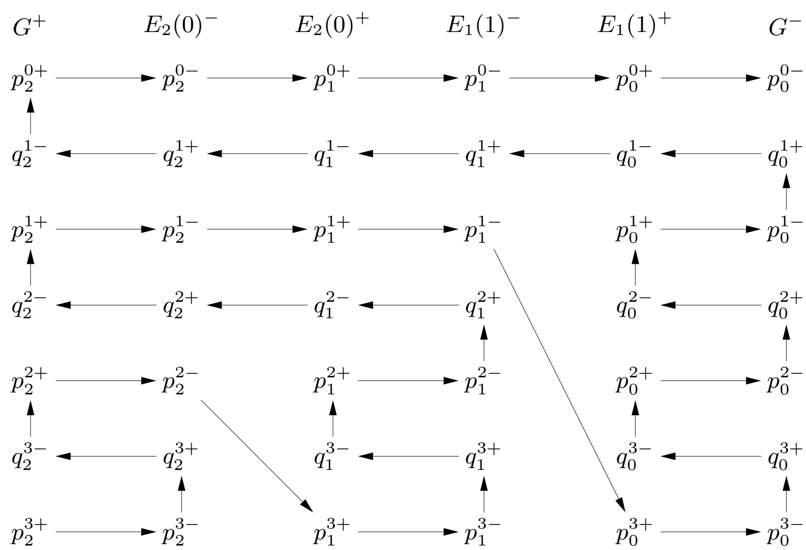

Fig. 16. The term graph N_t for the SAT instance $I = (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)$ and the truth assignment $t =< 0, 1 >$. The fact that t is not a satisfying assignment for I corresponds to the fact that p^{1-}_0 appears before p^{0+}_2 in N_t

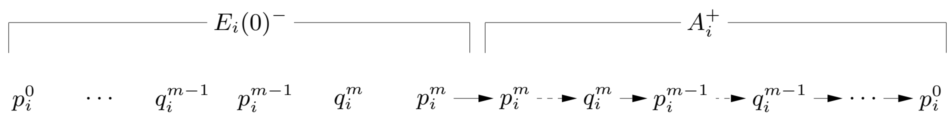

Fig. 17. A close-up of the boundary between $E_i(0)^-$ and A_i^+. The edges between the atoms of $E_i(0)^-$ have been omitted because they vary according to the SAT instance.

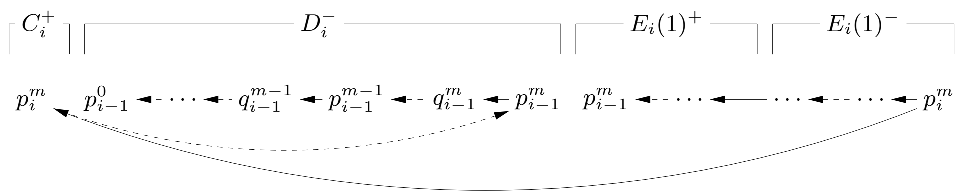

Fig. 18. A close-up of the boundary between D_i^- and $E_i(1)^+$. The edges in $E_i(1)$ vary depending on the SAT instance, but alternate regular and Lambek edges from p_i^m to p_{i-1}^m.

Definition 7. *Given a matching M, we denote the match of the occurrence of p_i^j in X^p to p_i^j in Y^q as $M(p_i^j, X^p, Y^q)$.*

Lemma 7. *$M(p_i^m, E_i(0)^-, A_i^+)$ cannot belong to an L^*-integral matching.*

Proof. Because there is a regular edge from p_i^m in $E_i(0)^-$ to p_i^m in A_i^+ the matched edge would violate $T(C)$, as seen in figure 17.

Lemma 8. *$M(p_{i-1}^m, D_i^-, E_i(1)^+)$ cannot belong to an L^*-integral matching.*

Proof. Consider figure 18. If such an edge were to exist, then the only way for a regular path from p_i^m in C_i^+ to p_{i-1}^m in D_i^- to exist is for their to be a regular path between p_i^m in C_i^+ and some vertex in $E_i(1)$. However, the first vertex of that path among the vertices of $E_i(1)$ is the target of a Lambek edge $\langle s, t\rangle$ and a regular path from s to t would need to include p_i^m in $E_i(1)^-$. However, that would violate $T(C)$.

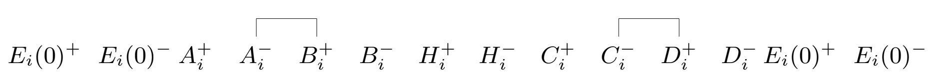

Fig. 19. The 14 hills of Π_i. The matches shown are obligatory because the atoms in those hills occur exactly twice in the term graph.

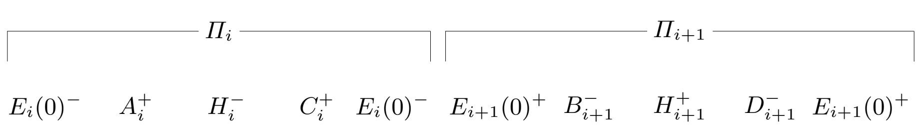

Fig. 20. The 10 hills containing p_i^j for $i < n$.

Proposition 9. *Any matching which does not extend M_t for some truth assignment t is not L^*-integral.*

Proof. Let M be an L^*-integral matching and let $1 \leq i \leq n$ be maximal such that neither of the following matchings are submatchings:

$$E_i(0)^+ \quad E_i(0)^- \quad A_i^+ \quad B_i^- \quad H_i^+ \quad H_i^- \quad C_i^+ \quad D_i^- \quad E_i(0)^+ \quad E_i(0)^-$$

$$E_i(0)^+ \quad E_i(0)^- \quad A_i^+ \quad B_i^- \quad H_i^+ \quad H_i^- \quad C_i^+ \quad D_i^- \quad E_i(0)^+ \quad E_i(0)^-$$

Let $0 \leq j \leq m$. Then, for $i = n$, p_i^j appears in 6 hills (from left to right):

$$E_i(0)^-, A_i^+, H_i^-, C_i^+, E_i(1)^-, G^+$$

For $i < n$, p_i^j appears in 10 hills:

$$E_n(0)^-, A_i^+, H_i^-, C_i^+, E_i(1)^-, E_{i+1}(0)^+, B_{i+1}^-, H_{i+1}^+, D_{i+1}^-, E_{i+1}(1)^+$$

However, because of our maximality assumption and the planarity requirement, we know that of the rightmost 5, only $E_{i+1}(t)^+$ does not match another of the rightmost 5. If $i = n$, let $E^+ = G^+$ and otherwise, let $E^+ = E_{i+1}(t)^+$. In either case, E^+ represents the rightmost unconstrained hill.

We now wish to consider the possible matchings of $E_i(0)^-, A_i^+, H_i^-, C_i^+, E_i(1)^-$ and E^+, but only for the occurrences of the atom p_i^j. For this section only, we will denote these matches by matching the hills they belong to, but we must remember that these matches are only for the p_i^j atoms and not the whole hill.

- Case 1: $\quad E_i(0)^- \quad A_i^+$
  Due to planarity, $M(p_i^j, E_i(0)^-, A_i^+)$ forces $M(p_i^m, E_i(0)^-, A_i^+)$ and by lemma 7 this matching is not L^*-integral.

- Case 2: $\quad C_i^+ \quad E_i(1)^-$
  Due to planarity, $M(p_i^j, C_i^+, E_i(1)^-)$ forces $M(p_{i-1}^m, D_i^-, E_i(1)^+)$ and by lemma 8 this matching is not L^*-integral.

- Case 3: $\quad H_i^- \quad E^+$
  Due to planarity, $M(p_i^j, H_i^-, E^+)$ forces $M(p_i^j, C_i^+, E_i(1)^-)$ which cannot be L^*-integral according to case 2.

This leaves us with only two possible matchings for p_i^j:

– Case 1: $E_i(0)^-$ A_i^+ H_i^- C_i^+ $E_i(1)^-$ E^+

We will now shift from analyzing the matches for a general p_i^j and focus on one important atom. Consider the possible matches for the atom p_{i-1}^m in D_i^-, the rightmost atom in D_i^-. There are five, as can be seen in figure 20 (from left to right) which we will rule out:

1. p_{i-1}^m in A_{i-1}^+
   Such a match is between the rightmost atom in D_i^- and the leftmost atom in A_{i-1}^+ as can be seen in figures 17 and 18. Then, we can see that no atom between these two has a regular out-edge to any atom not between these two. But, the regular in-neighbour of p_{i-1}^m in D_i^- is p_{i-1}^m in A_{i-1}^+ because of the match and the regular in-neighbour of p_{i-1}^m in A_{i-1}^+ is p_i^m in $E_{i-1}(0)^-$ because of the regular rewrite edge in figure 17. But, there is a Lambek edge whose target is p_{i-1}^m in D_i^- and whose source is p_i^m in C_i^+ as seen in figure 18. Then, there cannot possibly be a regular path from p_i^m in C_i^+ to p_{i-1}^m in D_i^-, resulting in a **T** violation.
2. p_{i-1}^m in C_{i-1}^+
   Because of planarity, the atoms in D_{i-1}^- would need to match the atoms in $E_{i-1}(1)^+$. However, by lemma 8, M would not be L^*-integral.
3. p_{i-1}^m in $E_i(0)^+$
   Contradicts our assumption that M does not have this submatching.
4. p_{i-1}^m in H_i^+
   Such a match would violate planarity since p_i^j in $E_i(0)^-$ matches p_i^j in C_i^+.
5. p_{i-1}^m in $E_i(1)^+$
   Then, M would not be L^*-integral by lemma 8.

– Case 2: $E_i(0)^-$ A_i^+ H_i^- C_i^+ $E_i(1)^-$ E^+

As in the previous case, we will focus on one important atom. This time, that atom will be p_{i+1}^m in A_i^+, the leftmost atom in A_i^+. Again, there are five possible matches:

1. p_{i+1}^m in $E_i(0)^-$
   Then, M would not be L^*-integral by lemma 7.
2. p_{i+1}^m in H_i^-
   Such a match would violate planarity, since p_i^j in A_i^+ matches p_i^j in $E_i(1)^-$.
3. p_{i+1}^m in $E_i(1)^-$
   Contradicts our assumption that M does not have this submatching.
4. p_{i+1}^m in B_{i+1}^-
   Contradicts the maximality assumption (because B_{i+1}^- is part of Π_{i+1}).
5. p_{i+1}^m in D_i^-
   Contradicts the maximality assumption (because D_{i+1}^- is part of Π_{i+1}).

Therefore, M must extend M_t for some truth assignment t.

Theorem 3. *$c_1 \wedge \ldots \wedge c_m$ is satisfiable iff Σ is derivable in L iff Σ is derivable in L^*.*

Proof. Propositions 7, 8 and 9 prove that any matching must partition into M_t and N_t for some truth assignment t. We need only consider the Lambek edges with a source in M_t and a target in N_t or vice versa but the only such edges are the rooted Lambek edges. It is tedious, but not difficult, to check that each such Lambek edge has an accompanying regular path and that **T**(CT) is not violated.

4 Conclusion

We have introduced a graphical representation of proof nets that is closely linked to both the structures of [8] and the abstract term graphs of [3]. Together these two results describe the boundary between tractability and intractability.

Our representation is very simple, requiring just two conditions (other than the matching conditions) to characterize correctness in the Lambek calculus. Furthermore, term graphs avoid the introduction of unneeded complexity such as the lambda terms of [5] and the algebraic terms of [1]. This has allowed us to provide a more intuitive proof of the NP-completeness result of [8], which allows us to more clearly see the boundary of tractability for the product-free Lambek calculus.

Acknowledgements

I must thank Gerald Penn for guiding this research as well as the useful comments of the three anonymous reviewers.

References

1. P. De Groote. An algebraic correctness criterion for intuitionistic multiplicative proof-nets. *Theoretical Computer Science*, 224(1-2):115–134, 1999.
2. T.A.D. Fowler. Efficient Parsing with the Product-Free Lambek Calculus. In *Proceedings of COLING '08*, 2008.
3. T.A.D Fowler. A Polynomial Time Algorithm for Parsing with the Bounded Order Lambek Calculus. In *Proceedings of Mathematics of Language 11*, 2009.
4. Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958.
5. Gerald Penn. A Graph-Theoretic Approach to Sequent Derivability in the Lambek Calculus. *Electronic Notes in Theoretical Computer Science*, 53:274–295, 2004.
6. Mati Pentus. Product-Free Lambek Calculus and Context-Free Grammars. *The Journal of Symbolic Logic*, 62(2):648–660, 1997.
7. Dirk Roorda. *Resource Logics: Proof-theoretical Investigations*. PhD thesis, Universiteit van Amsterdam, 1991.
8. Y. Savateev. Product-Free Lambek Calculus Is NP-Complete. In *Proceedings of the 2009 International Symposium on Logical Foundations of Computer Science*, pages 380–394. Springer-Verlag Berlin, Heidelberg, 2009.
9. H.J. Tiede. *Deductive Systems and Grammars: Proofs as Grammatical Structures*. PhD thesis, Indiana University, 1999.