A GRAPH THEORETIC EXTENSION OF BOOLEAN LOGIC

CAMERON CALK

Context. This internship was supervised by Anupam Das and Olivier Laurent and took place at Laboratoire de l'Informatique du Parallelisme (LIP), ENS de Lyon between 06/06/2016 and 29/07/2016, with a break from 27/06/2016 to 08/07/2016.

1. Overview

In this project we explore a graph theoretic extension of Boolean logic, using notions developed in previous work concerning graph representations of propositional terms (*i.e.* formulae). These allow us to consider graphs as synctactic logical objects in their own right, and we are thus able to define notions of evaluation and entailment for arbitrary graphs. The motivation for this is to establish a model of Boolean logic on graphs in order to see how logical phenomena scale to this richer setting.

This work draws on the theory of *relation webs* [4] [6] [3], which are graphs representing the equivalence classes of terms modulo associativity and commutativity of conjunction and disjunction. A structural correspondence was established between the Boolean function computed by a term, and its relation web [3], effectively translating the semantic notions of Boolean logic into a graph theoretic setting. More specifically, the maximal cliques and maximal stable sets in the relation web of a term correspond exactly to the conjunctions (resp. disjunctions) in its minimal disjunctive (resp. conjunctive) normal form. The relation webs of terms, however, correspond to a fairly restricted class of graphs, known as *co-graphs*. It is natural, therefore, to apply these translated notions to larger classes of graphs, and study the consequences.

In this report we formalise an extension of Boolean logic on arbitrary graphs, and illustrate the effects of this generalisation in terms of logical behaviour. This leads to an attempt to determine the graphs for which our notion of evaluation is total or deterministic. We characterise the class of graphs for which evaluation is deterministic and obtain partial results concerning totality. Next we examine the interaction between these generalised forms of entailment and evaluation, notably proving *soundness* and *completeness*.

We then turn to the problem of evaluating arbitrary graphs under Boolean assignments. Unlike in the term setting, this is a computationally difficult task, but we are nonetheless able to give a form of inductive algorithm by appealing to the *modular decomposition* of a graph [5]. This algorithm was also useful in reducing some of our questions concerning evaluation to the so-called *prime* graphs.

We conclude this report with a brief explanation of other work that we did during the internship. Throughout, we present useful and interesting examples for future research in this area. Proofs not presented in this report can be found at anupamdas.com/ProofsInternship.pdf.

2. Preliminaries

Here we present basic elements of graph theory and propositional logic, and introduce the previous work which we have drawn from.

2.1. Terms.

We consider Boolean logic on the language $\mathcal{L} = \{\wedge, \vee\}$ and a set of variables $Var = \{x_1, x_2, \dots\}$. The set Ter of *terms* is built freely in the usual way;

$$t \quad ::= \quad x \in Var \quad | \quad t \vee t \quad | \quad t \wedge t$$

We write $Var(t)$ to denote the set of variables occurring in t. We do not consider negation or constant symbols for technical reasons; we refer the reader to [3] pp. 3-4.

We say that a term t is *linear* if every $x \in Var(t)$ has only one occurrence in t. For all $x, y \in Var(t)$ we can define the *first common connective* of x and y in t; it is the connective (either $\vee$ or $\wedge$) at the root of the smallest sub-tree of the term tree of t containing both x and y.

Example 2.1. Let $t = ((x \vee w) \wedge y) \vee (z \wedge v)$. Then the first common connective of x and y in t is $\wedge$, and that of x and z is $\vee$.

An *assignment* is a map $Var \to \{0, 1\}$, which we identify with a subset of Var, namely the set of variables mapped to 1 by the assignment. So the assignment $Y \subseteq Var$ is that in which the elements of Y are assigned 1, and the elements of its complement are assigned 0. Thus, the assigment $\overline{Y} = Var \setminus Y$ is that in which the elements of Y are assigned 0, and the elements of its complement are assigned 1.

A *Boolean function* is a map which associates assignments with Booleans; $f : \mathcal{P}(Var) \to \{0, 1\}$. We say that such a function is *monotone* if for all $X, Y \subseteq Var$, $Y \subseteq X \Rightarrow f(Y) \leq f(X)$.

Definition 2.2 (Minterms and maxterms)**.** Let f be a monotone Boolean function. A set $Y \subseteq Var$ is a *minterm* (resp. *maxterm*) for f if it is a minimal set such that $f(Y) = 1$ (resp. $f(\overline{Y}) = 0$). The set of all minterms (resp. maxterms) of f is denoted $MIN(f)$ (resp. $MAX(f)$).

A term t computes a Boolean function in the usual way and, in our case, these are monotone due to the fact that there is no negation symbol in our language. We will abuse notation by denoting both the term and the function by t. Since $Var(t)$ is finite, we typically consider assignments of finite support [3] pp. 5-7.

Minterms (resp. maxterms) characterise a monotone Boolean function. They correspond to the minimal disjunctive (resp. conjunctive) normal forms of the function. We refer the reader to [2] pp. 3-49 for a comprehensive introduction to their theory.

Proposition 2.3. *Let t be a term. Then for all $X \subseteq Var$, we have*

$$t(X) = 1 \iff \exists S \in MIN(t), S \subseteq X$$
$$t(X) = 0 \iff \exists T \in MAX(t), T \cap X = \emptyset$$

For terms s and t, we write $s \leq t$ if $\forall X \subseteq Var$, $s(X) \leq t(X)$. This relation, the fact that s *entails* t, can again be characterised by minterms and maxterms;

Proposition 2.4. *For s, t terms with $Var(s) = Var(t)$, the following are equivalent:*

(1) $s \leq t$.

(2) $\forall S \in MIN(s), \forall T \in MAX(t),\ S \cap T \neq \emptyset$.
(3) $\forall S \in MIN(s),\ \exists S' \in MIN(t),\ S' \subseteq S$.
(4) $\forall T \in MAX(t),\ \exists T' \in MAX(s),\ T' \subseteq T$.

Thus, the semantics of terms can be characterised by a structural property of Boolean functions, rather than an inductive definition.

2.2. **Graphs.** A (simple, loopless, undirected, finite) graph G is a couple (V, E) where V is a finite set, called the set of *vertices* or *nodes*, and $E \subseteq \binom{V}{2} := \{\{x, y\} \mid x, y \in V, \quad x \neq y\}$ is a set of unordered pairs of distinct vertices, called the set of *edges*. A few classic families of such graphs are given below;

Definition 2.5. Let $V = \{v_1, \ldots, v_n\}$.
An n-path, written P_n, is the graph $(V, \{\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}\})$.
An n-cycle, written C_n, is the graph $(V, \{\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}\})$.
A complete graph on n vertices, written K_n, is the graph $(V, \binom{V}{2})$

For a graph $G = (V, E)$, we say that x and y are neighbours if $\{x, y\} \in E$ and we define its *dual*, $\overline{G} := (V, \binom{V}{2} \setminus E)$. We say that two graphs G and H are isomorphic, and we write $G \equiv H$, if there exists a bijection $\psi : V(G) \to V(H)$ which preserves edges.

We say that H is a subgraph of G and we write $H \leq G$ if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. The subgraph of G *induced* by a subset $U \subseteq V(G)$ is the graph $G_{|U} = \left(U, E(G) \cap \binom{U}{2}\right)$. If H is a graph, we say that a graph G is H-free if $\forall U \subseteq V(G),\ G_{|U} \not\equiv H$.

A graph G is *connected* (resp. *co-connected*) if $\forall x, y \in V(G)$, $\exists H \leq G$ such that $x, y \in V(H)$ and $H \equiv P_k$ (resp. $H \equiv \overline{P}_k$) for some k. If not, we say that G is *disconnected* (resp. *co-disconnected*). A connected (resp. co-connected) *component* of G is a maximal connected (resp. co-connected) induced subgraph of G. When G is both connected and co-connected, we say that G is *bi-connected*.

A subset $U \subseteq V(G)$ is a *clique* (resp. *stable set*) if $G_{|U} \equiv K_k$ (resp. $G_{|U} \equiv \overline{K}_k$) for some k.

Definition 2.6 (Co-graphs)**.** A graph G is a *co-graph* if every induced subgraph of G is disconnected or co-disconnected.

We have the following characterisation of co-graphs [2] pp. 464:

Proposition 2.7. *G is a co-graph if and only if G is P_4-free.*

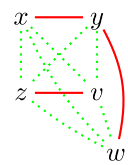
2.3. **Relation webs and correspondence.** A useful representation of a linear term, which links Boolean logic with graphs, is its web [3] pp. 7-9.

Recall the definition of first common connective, given above in Subsection 2.1.

Definition 2.8 (Relation Web)**.** The *(relation) web* $\mathcal{W}(t)$ of a linear term t is the graph whose vertex set is $Var(t)$, and in which $\{x, y\}$ is an edge if and only if the first common connective of x and y in t is $\wedge$.

For ease of representation and to underline the duality of 'conjunction' and 'disjunction' in a graph, we will write, for $x, y \in Var(t)$, x —— y (resp. x ⋯⋯ y) to denote that x is a neighbour (resp. non-neighbour) of y in G; we say that that x and y are in 'conjunction (resp. disjunction) in G'. Additionally, we will refer to the cliques (resp. stable sets) in G as conjunctive cliques (resp. disjunctive cliques).

Example 2.9. The web of the term $((x \vee w) \wedge y) \vee (z \wedge v)$ is:



Notation 2.10. For a graph G, we denote the set of its maximal conjunctive (resp. disjunctive) cliques by $MC_\wedge(G)$ (resp. $MC_\vee(G)$). For $U \subseteq V(G)$, we will abuse notation by writing $MC_\star(U)$ instead of $MC_\star(G_{|U})$ for $\star \in \{\vee, \wedge\}$.

These graph representations have been explored in previous works [3] [4] [6]. For example, the web of a term represents its equivalence class modulo associativity and commutativity of conjunction and disjunction (AC). Most importantly, there is a correspondence between the minterms (resp. maxterms) of a term, and the maximal conjunctive (resp. disjunctive) cliques of its web.

Proposition 2.11. *Let t be a linear term. Then $MIN(t) = MC_\wedge(\mathcal{W}(t))$ and $MAX(t) = MC_\vee(\mathcal{W}(t))$.*

Using this correspondence and Proposition 2.3 (resp. Proposition 2.4), we can define evaluation (resp. entailment) of a term using the maximal conjunctive and disjunctive cliques of its web. Thus, we have entirely graph theoretic formulations of evaluation and entailment.

Interestingly, the webs of terms correspond to a specific class of graphs:

Proposition 2.12. *If t is a linear term, then $\mathcal{W}(t)$ is a co-graph. Conversely, if G is a co-graph, then there exists some linear term t such that $G = \mathcal{W}(t)$.*

As previously mentioned, the point of this work is to study the graph theoretic formulations of entailment and evaluation *outside* the class of co-graphs.

3. Evaluation and entailment for arbitrary graphs

We will consider graphs G such that $V(G) \subseteq Var$. The notions developed in the previous section provide natural extensions for evaluation and entailment in the context of arbitrary graphs.

Definition 3.1 (Evaluation)**.** We define a binary relation $eval_G$ called an *evaluation* at a graph G on $\mathcal{P}(Var) \times \{0, 1\}$, by:

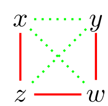
- $eval_G(X, 1)$ provided that $\exists S \in MC_\wedge(G)$ such that $S \subseteq X$.
- $eval_G(X, 0)$ provided that $\exists T \in MC_\vee(G)$ such that $T \cap X = \emptyset$.

Note the similarity between this definition and the characterisation of the evaluation of terms in Proposition 2.3; the essential difference is that here our evaluation is a relation, not a function. This is due to the fact that the totality and determinism of the relation are not immediate in the general setting. We remind the reader:

Definition 3.2 (Total and Deterministic)**.** For a binary relation $\mathcal{R}$ on $A \times B$, we say that

1. $\mathcal{R}$ is total provided that $\forall a \in A$, $\exists b \in B$ such that $\mathcal{R}(a, b)$.
2. $\mathcal{R}$ is deterministic provided that $\forall a \in A$, $\forall b \in B$, $\forall c \in B$, if $\mathcal{R}(a, b)$ and $\mathcal{R}(a, c)$, then $b = c$.

When working with co-graphs, the evaluation relation is (by default) a function (*i.e.* total and deterministic), since a graph of that class corresponds exactly to the web of some term, and the notion of evaluating to 1 or 0 is inherited from the usual theory of Boolean terms and functions. In the case of arbitrary graphs, this characteristic of the evaluation relation does not come for free. The problem can be seen, not surprisingly (*cf.* Proposition 2.7), with the P_4 configuration:



Note that there is a maximal conjunctive clique $\{w, z\}$ and a maximal disjunctive clique $\{x, y\}$ which are disjoint, so we can construct an assignment showing non-determinism of evaluation at P_4: if we take $X = \{w, z\}$, we have $eval_{P_4}(X, 1)$ since $\{w, z\}$ is a maximal conjunctive clique, and $eval_{P_4}(X, 0)$ since $\{x, y\}$ is a maximal disjuntive clique which is disjoint from X. The P_4 also has an assignment witnessing its non-totality; if we take $X = \{x, y\}$, by inspection we have that neither $eval_{P_4}(X, 1)$ nor $eval_{P_4}(X, 0)$. Thus the P_4 defines an evaluation relation which is neither total nor deterministic.

We also get a richer notion of entailment in this general setting:

Definition 3.3 (Entailment)**.** Let G and H be graphs, with $V(G) = V(H)$. We define *conjunctive entailment* $\underset{\wedge}{\rightarrow}$ and *disjunctive entailment* $\underset{\vee}{\rightarrow}$ as follows:

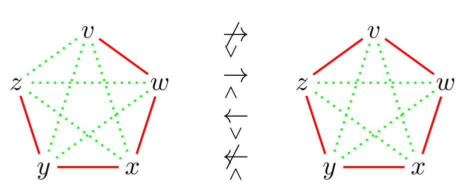
$$G \underset{\wedge}{\rightarrow} H \text{ provided that } \forall S \in MC_\wedge(G), \exists S' \in MC_\wedge(H), S' \subseteq S.$$

$$G \underset{\vee}{\rightarrow} H \text{ provided that } \forall T \in MC_\vee(H), \exists T' \in MC_\vee(G), T' \subseteq T.$$

The similarity between this definition and the characterisation in Proposition 2.4 is again quite clear, but notice that a graph theoretic formulation of characterisation (2) has been omitted. This is due to the fact that this notion is neither reflexive nor transitive in the general case. However, the relations defined above behave well in this regard:

Lemma 3.4. *The relations* $\underset{\wedge}{\rightarrow}$ *and* $\underset{\vee}{\rightarrow}$ *are reflexive and transitive.*

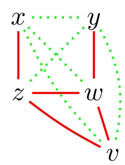
Whereas in the context of co-graphs these characterisations of entailment coincide, we distinguish them when dealing with arbitrary graphs because in this setting they do not. Indeed, consider the example:



These relations can be generalised to P_k and C_k for every $k \geq 5$, and many more examples of non-coincidence of these relations can be found. Again, we see that the notion has become richer in this setting.

3.1. **Determinism.** Recall the problem for determinism of the P_4, namely that there is a maximal conjunctive clique disjoint from a maximal disjunctive clique. This lead us to look at the notion of a 'settling node' [1] pp. 7, which 'settles' the indeterminacy between the disjoint maximal conjunctive and disjunctive cliques:

(1)



The added node forces the intersection of the disjoint maximal cliques from the P_4, thus solving the problem of determinism of evaluation. We have found that it is exactly this consideration of intersection which allows us to characterise the graphs for which evaluation is deterministic.

Definition 3.5 (CIS graphs [1] pp. 2)**.** A graph G is *CIS* if $\forall S \in MC_\wedge(G)$, $\forall T \in MC_\vee(G)$, $S \cap T \neq \emptyset$.

Proposition 3.6. *A graph G is CIS if and only if $eval_G$ is deterministic.*

Proof. We prove the contrapositive, which by Definition 3.2, can be written:

$$G \text{ is not CIS} \iff \exists X \subseteq Var, eval_G(X,1) \text{ and } eval_G(X,0).$$

Suppose that G is not a CIS graph. Then $\exists S \in MC_\wedge(G), \exists T \in MC_\vee(G)$ such that $S \cap T = \emptyset$. Let $X := S$. Then $S \cap X = S$ so $eval_G(X,1)$, and $T \cap X = T \cap S = \emptyset$ so $eval_G(X,0)$.

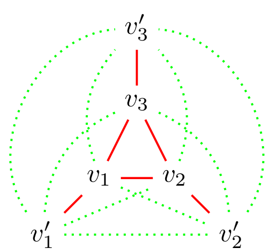
Conversely, assume that $\exists X \subseteq Var$ such that $eval_G(X,1)$ and $eval_G(X,0)$. Since $eval_G(X,1)$, $\exists S \in MC_\wedge(G)$ such that $S \subseteq X$, and since $eval_G(X,0)$, $\exists T \in MC_\vee(G)$ such that $T \cap X = \emptyset$. Then

$$S \cap T = (S \cap X) \cap T = S \cap (T \cap X) = S \cap \emptyset = \emptyset$$

so G is not a CIS graph. ☐

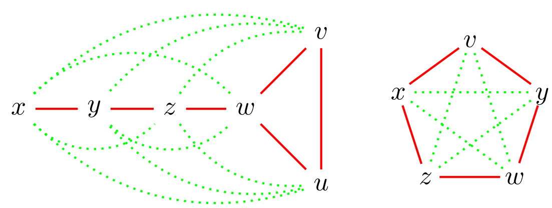
Observe that this result states that the CIS graphs correspond to the *partial* Boolean functions, *i.e.* the relations that are determined when defined.

Adding a 'settling node' was a way to force the CIS property for the P_4. However, every P_4 configuration in a graph being 'settled' does not suffice to conclude that the graph is CIS. Indeed, consider the following example from [1]:



which contains three P_4 configurations, all of which are 'settled', but $\{v_1, v_2, v_3\}$ is a maximal conjunctive clique which is disjoint from the maximal disjunctive clique $\{v_1', v_2', v_3'\}$. Indeed, characterising CIS graphs is not easy; it is an open problem whether they can be recognised in polynomial time [1] pp. 2. Therefore, deciding whether a graph yields a deterministic evaluation is *a priori* computationally difficult.

3.2. **Totality.** Characterising the totality of evaluation has proven more difficult. Firstly, notice that evaluation of CIS graphs is not always total; the 'settled' P_4 from (1) is a CIS graph, but if we take $X = \{x, y\}$, we have neither $eval_G(X, 1)$ nor $eval_G(X, 0)$. On the other hand, the evaluation relation being total is not exclusive to co-graphs:



These are examples of graphs with several induced P_4 configurations, but which compute total evaluations.

We have however been able to give a necessary condition for totality:

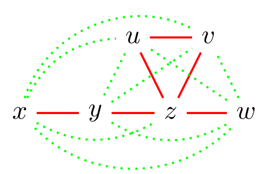
Definition 3.7 (CUS graphs)**.** A graph G is *CUS* if $\exists S \in MC_\vee(G)$, $\exists T \in MC_\wedge(G)$, $S \cup T = V(G)$.

Proposition 3.8. *Let G be a bi-connected graph. If $eval_G$ is total, G is not CUS.*

Notice the condition of bi-connectedness; it is appropriate because, as we show in the following section, the question of totality of evaluation can be reduced to the 'bi-connected components' of a graph.

Unfortunately, this condition is not sufficient. Consider the following:

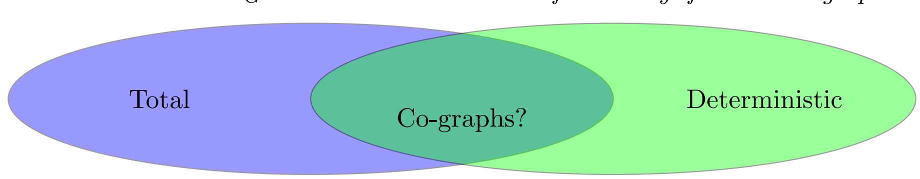
(2)



This graph does not have the CUS property, but its evaluation relation is not defined at the assignment $\{x, w\}$.

3.3. **Which graphs compute functions?** We have obtained the results presented above concerning totality and determinism, but we have been unable to characterise the graphs for which they coincide. A failure to find a single example of a non-co-graph for which the evaluation relation is both total and deterministic leads us to the following question;

Question 3.9. *Is $eval_G$ total and deterministic if and only if G is a co-graph?*



If this were the case, the only graphs corresponding to Boolean functions semantically would be those already corresponding to equivalence classes of linear terms syntactically. However, the class of Boolean functions computed by linear terms is not the whole class of Boolean functions, and we suspect that some of the 'extra' ones might be extractable in the richer arbitrary graph setting. For now, the question remains open.

3.4. **Entailment.** Here we illustrate an important link between entailment and evaluation, namely the *soundness* and *completeness* of the former with respect to the latter. This shows that, while the notions of entailment and evaluation are seemingly less 'well-behaved' in the setting of arbitrary graphs, they are still related to each other in such a way as to yield a coherent logical structure.

Theorem 3.10 (Soundness and completeness)**.** *Let G, H be graphs, with $V(G) = V(H)$. We have:*

$$G \underset{\wedge}{\rightarrow} H \iff \forall X \subseteq \mathit{Var}, \mathit{eval}_G(X,1) \Rightarrow \mathit{eval}_H(X,1)$$
$$G \underset{\vee}{\rightarrow} H \iff \forall X \subseteq \mathit{Var}, \mathit{eval}_H(X,0) \Rightarrow \mathit{eval}_G(X,0)$$

Proof. We will prove the equivalence for the first case, the other being similar.

Let $X \subseteq V$ and suppose $\mathit{eval}_G(X,1)$, *i.e.* $\exists S \in MC_\vee(G)$ such that $S \subseteq X$. Since $G \underset{\wedge}{\rightarrow} H$, we have $\exists S' \in MC_\vee(H)$ such that $S' \subseteq S$. By transitivity of inclusion, $S' \subseteq X$, so $\mathit{eval}_H(X,1)$.

Let $S \in MC_\vee(G)$. Then we have $\mathit{eval}_G(S,1)$, so by hypothesis, $\mathit{eval}_H(S,1)$. By definition, this means that $\exists S' \in MC_\vee(H)$ such that $S' \subseteq S$. Since S was selected arbitrarily, we have that $\forall S \in MC_\wedge(G), \exists S' \in MC_\wedge(H), S' \subseteq S$, *i.e.* $G \underset{\wedge}{\rightarrow} H$. □

We also prove that the totality and determinism of the evaluation relation have an impact on the coincidence of the two forms of entailment:

Proposition 3.11. *Let G,H be graphs such that eval_G is total and eval_H is deterministic. Then:*

$$G \underset{\wedge}{\rightarrow} H \Rightarrow G \underset{\vee}{\rightarrow} H$$
$$H \underset{\vee}{\rightarrow} G \Rightarrow H \underset{\wedge}{\rightarrow} G$$

Corollary 3.12. *Let G,H be graphs such that eval_G and eval_H are both total and deterministic. Then*

$$G \underset{\wedge}{\rightarrow} H \iff G \underset{\vee}{\rightarrow} H$$

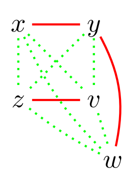
This last corollary allows us to link, in the context of Question 3.9, the coincidence of conjunctive and disjunctive entailment with totality and determinism of the evaluation, rather than with P_4-freeness.

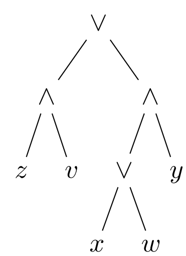
4. Algorithm for evaluation

The goal of this section is to find a decomposition for arbitrary graphs which generates a recursive algorithm for evaluation. During the course of this project we found that the problem of evaluating an arbitrary graph is **NP**-complete, so this algorithm can only get us so far. Intuitively, this can be seen by considering that in order to check the evaluation of a graph 'by hand', one must check (at worst) every one of its maximal cliques, of which there can be many. It is still generally useful to have an algorithm for evaluating arbitrary graphs, and it reduces the size and number of the cliques we need to consider. Furthermore, as we will see, this algorithm also helps us reduce questions about totality and determinism to simpler cases.

Throughout this section we will write $x \overset{\wedge}{\text{——}} y$ for $x \text{ —— } y$ and $x \overset{\vee}{\text{——}} y$ for $x \cdots\cdots y$. Now, if $\star \in \{\vee, \wedge\}$ and G, H are graphs with disjoint vertex sets, let $G \overset{\star}{\text{——}} H$ denote the graph with vertex set $V(G) \sqcup V(H)$ and edges $E(G) \sqcup E(H)$, where we also add an edge for each pair $x \in V(G)$ and $y \in V(H)$ when $\star = \wedge$. We will refer to this graph as the conjunctive (resp. disjunctive) composition of G and H, if $\star = \wedge$ (resp. $\vee$).

Let us momentarily return to the setting of terms and their webs, which we recall are co-graphs. When evaluating terms, we can devise an algorithm based on their inductive definition. This can easily be implemented at the level of co-graphs themselves. We remind the reader that in the case of a co-graph G, we know that G is either conjunctively or disjunctively disconnected. Therefore, we can partition G into $G_1 \text{ —— } G_2$ or $G_1 \cdots\cdots G_2$. Then, exactly as in the case of terms, we use the inductive rules about the evaluation of a conjunction or disjunction of terms, *e.g.* $G_1 \text{ —— } G_2$ evaluates to 1 if and only if G_1 and G_2 evaluate to 1. This recursive process of 'splitting' is represented by the graph's *co-tree*, which then gives an algorithm for evaluation of the co-graph. For example, below is a co-graph, on the left, and its co-tree, on the right:





Notice that this is simply the term tree of $t = ((x \vee w) \wedge y) \vee (z \wedge v)$, of which the above graph is the web. We work our way from the top downward, using the label at each node to decide how to 'combine' the evaluations of its children. The leaves of this tree are always simply variables since all induced subgraphs of a co-graph are splittable in the way described above. This process does not give a unique tree, since in general there are several choices for splitting the graph. However, they are all equivalent modulo AC [2] pp. 463-466.

4.1. **Reducing evaluation to bi-connected components.** We will use a similar decomposition to the above in the case of arbitrary graphs; when we can, we will split the graph as above, and when we cannot, we will stop. This defines a *decomposition tree*, which is similar to the co-tree of a co-graph, but the leaves of this tree are *a priori* not single variables, but are the *bi-connected components* of G.

Definition 4.1 (Bi-connected components)**.** A *bi-connected component* of G is a maximal bi-connected induced subgraph of G. We denote the set of bi-connected components of G by $BiCC(G)$.

Lemma 4.2. *We have the following:*

- *If G is bi-connected, $BiCC(G) = \{G\}$*
- *If there exist G_1, G_2 such that $G = G_1 \overset{\star}{\text{——}} G_2$, then $BiCC(G) = BiCC(G_1) \cup BiCC(G_2)$, where $\star \in \{\vee, \wedge\}$.*

Note that if $|BiCC(G)| \geq 2$, then G can be decomposed into $G = G_1 \overset{\star}{\text{——}} G_2$, $\star \in \{\vee, \wedge\}$, because otherwise G would be bi-connected.

We will now show that decomposing an arbitrary graph as above yields a similar strategy for its evaluation to that employed in the term setting.

Lemma 4.3. *For G_1 and G_2 with $V(G_1) \cap V(G_2) = \emptyset$, and $\star \in \{\vee, \wedge\}$, we have*

- $MC_\star(G_1 \overset{\star}{\text{——}} G_2) = \{S_1 \cup S_2, S_1 \in MC_\star(G_1), S_2 \in MC_\star(G_2)\}$
- $MC_{\bar\star}(G_1 \overset{\star}{\text{——}} G_2) = MC_{\bar\star}(G_1) \cup MC_{\bar\star}(G_2)$

Where $\bar\star \in \{\vee, \wedge\} \setminus \{\star\}$.

Proposition 4.4. *Let $X \subseteq Var$. If $G = G_1 \cdots\cdots G_2$ then:*

$$\begin{array}{lcl} eval_G(X,0) & \iff & eval_{G_1}(X,0) \text{ and } eval_{G_2}(X,0) \\ eval_G(X,1) & \iff & eval_{G_1}(X,1) \text{ or } eval_{G_2}(X,1) \end{array}$$

and if $G = G_1 \text{——} G_2$ then:

$$\begin{array}{lcl} eval_G(X,1) & \iff & eval_{G_1}(X,1) \text{ and } eval_{G_2}(X,1) \\ eval_G(X,0) & \iff & eval_{G_1}(X,0) \text{ or } eval_{G_2}(X,0) \end{array}$$

Using this proposition, we can clearly evaluate some node in the decomposition tree by using the evaluation of its two children. This can easily be used to devise a recursive algorithm for evaluating graphs, and effectively reduces the task of evaluating an arbitrary graph to evaluating its bi-connected components.

4.2. **Going further; reducing to modules.** We can even take this decomposition a step further, using what is called the *modular decomposition* of a bi-connected graph. First, some preliminaries:

Definition 4.5 (Module)**.** A *module* in a graph G is a subset $M \subseteq V(G)$ such that for every $y \in V(G) \setminus M$, we have one of the following:

- $\forall m \in M$, $m \text{——} y$, in which case we say that y is *complete* to M.
- $\forall m \in M$, $m \cdots\cdots y$, in which case we say that y is *anti-complete* to M.

i.e. all elements of M have the same neighbourhood outside of M. The sets $V(G)$, $\emptyset$, and $\{x\}$, for $x \in V(G)$, are always modules and are known as *trivial modules*. If every module in G is trivial, we say that G is a *prime* graph. M is a *proper* module when $M \neq V(G)$.

The following are basic properties of modules which will allow us to establish modular decomposition [5] pp. 4-9.

Lemma 4.6. *The intersection of two modules is a module, and if two modules have a non-empty intersection, their union is a module.*

Theorem 4.7. *Let G be a bi-connected graph. Then any two maximal (for set inclusion) proper modules of G are disjoint.*

We can now prove the main modular decomposition result:

Corollary 4.8. *If G is a bi-connected graph, G admits a unique partition*

$$\mathcal{P}_G = \{M_1, M_2 \ldots, M_n\}$$

into maximal proper modules.

Another important intermediate result is the following:

Lemma 4.9. *Any two disjoint modules are either complete or anti-complete to each other,* i.e. *for any two disjoint modules M and N, we have M —— N or $M \cdots\cdots N$.*

This allows us to see that, given a bi-connected graph, we can 'zoom out' and look at its modules as if they were the nodes of a smaller graph.

Definition 4.10 (Quotient graph)**.** Let G be a bi-connected graph, and $\mathcal{P} = \{M_1, M_2, \ldots, M_n\}$ its modular partition. We define the *quotient graph* of G by:

$$G/\mathcal{P} \quad := \quad (\mathcal{P}, \{\{M_i, M_j\} \mid M_i \text{——} M_j \text{ in } G\})$$

Lemma 4.11. *The quotient graph of a bi-connected graph is a prime graph.*

Remark 4.12. Note that by Corollary 4.8 and Lemma 4.9 , the modular partition $\mathcal{P}$ on a bi-connected graph G defines a graph homomorphism:

$$\begin{aligned} \varphi : V(G) &\to \mathcal{P} \\ x &\mapsto M \text{ such that } x \in M \end{aligned}$$

since if x —— y in G, either they are mapped to the same module (in which case we obtain a loop), or they are not, in which case, by Lemma 4.9 , $\varphi(x)$—$\varphi(y)$ in $G/\mathcal{P}$. The quotient graph, as the name indicates, is the image of G by this homomorphism, where we have removed the loops. Notice, however, that allowing or disallowing loops in a graph does not affect the notions of cliques, stables sets, or bi-connected components. Therefore, we can use properties of graph homomorphisms in our arguments.

We will now develop a method of evaluating a bi-connected graph by evaluating the quotient graph in a specific way.

Lemma 4.13. *Let G be a bi-connected graph, $\mathcal{P} = \{M_1, M_2, \ldots, M_n\}$ its modular partition, φ the graph homomorphism induced by $\mathcal{P}$. We have, for $\star \in \{\wedge, \vee\}$,*

(1) $S \in MC_\star(G) \Rightarrow \varphi(S) \in MC_\star(G/\mathcal{P})$.
(2) $S \in MC_\star(G/\mathcal{P}) \Rightarrow \forall M_i \in S, \forall S_i \in MC_\star(M_i)$, *we have that* $\bigcup_i S_i \in MC_\star(G)$.

Definition 4.14 (Positive/negative quotient assignments)**.** Given an assignment $X \subseteq Var$ and a bi-connected graph G, we define

- The *positive quotient assignment* given by X on G;

$$Y_{X,G} := \{M \in \mathcal{P}_G | eval_M(X, 1)\}$$

- The *negative quotient assignment* given by X on G;

$$Z_{X,G} := \{M \in \mathcal{P}_G | \neg eval_M(X, 0)\}$$

Note that we abuse notation in writing $eval_M$ to mean $eval_{G_{|M}}$.

Now we can determine the evaluation of the entire bi-connected graph in terms of the evaluation of its modules and the subsequent evaluation of the quotient graph:

Theorem 4.15. *For G a bi-connected graph with modular partition $\mathcal{P}$, we have, for all assigments $X \subseteq Var$,*

(1) $eval_G(X,1) \iff eval_{G/\mathcal{P}}(Y_{X,G},1)$.

(2) $eval_G(X,0) \iff eval_{G/\mathcal{P}}(Z_{X,G},0)$.

Proof. We prove (1), (2) being similar. Let $X \subseteq Var$.

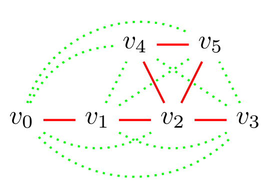
Suppose $eval_G(X,1)$. Then there exists a maximal conjunctive clique S of G such that $S \subseteq X$. By Lemma 4.13, we just need to show that $\varphi(S) \subseteq Y_{X,G}$. Let $M \in \varphi(S)$. We have that $S \cap M$ is a clique in M, and is maximal in M because otherwise S could be extended in G. So $S \cap M \in MC_\wedge(M)$, and by hypothesis $M \cap S \subseteq X$, which gives $eval_M(X,1)$, and therefore $M \in Y_{X,G}$, proving the direct implication.

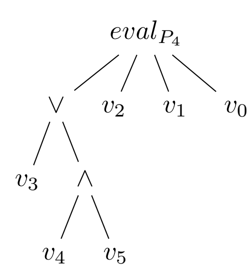
Suppose that $eval_{G/\mathcal{P}}(Y_{X,G},1)$. Then there exists some maximal conjunctive clique S of $G/\mathcal{P}$ such that $S \subseteq Y_{X,G}$. This means that for each $M_i \in S$, $eval_{M_i}(X,1)$, so $\exists S_i \in MC_\wedge(M_i)$ with $S_i \subseteq X$. Therefore $\bigcup_i S_i \subseteq X$, and by Lemma 4.13, $\bigcup_i S_i \in MC_\wedge(G)$. Therefore $eval_G(X,1)$. □

4.3. **Putting it all together.** We will now briefly illustrate how the preceding results can be used to devise an algorithm for the evaluation of an arbitrary graph G. First we will construct the *modular decomposition tree*: as described above, if G is 'splittable', *i.e.* not bi-connected, then we partition G into $G_1 \overset{\star}{—} G_2$, where $\star \in \{\vee, \wedge\}$, label the root with $\star$, and add two child nodes labelled G_1 and G_2. If G is bi-connected, we find its modular partition, label the root with $eval_{G/\mathcal{P}_G}$, and add a child node for each of its maximal proper modules. This tree provides the intuition for the construction of the algorithm; given some node in the tree, we can determine its evaluation by 'combining' the evaluations of its children according to its label, using Propositions 4.4 and Theorem 4.15. This clearly defines a recursive algorithm.

Notice that the only case in which we have to evaluate a graph 'by hand' is when G is bi-connected and we evaluate the quotient graph. Since every quotient graph is prime, we have reduced the problem of evaluating an arbitrary graph to evaluating prime graphs.

Example 4.16. Below is a bi-connected graph (*cf.* (2)) and its modular decomposition tree:





4.4. **Reducing totality and determinism to prime graphs.** The problem of determining whether some graph is deterministic or total is *a priori* a computationally difficult task. However, the preceding results allow us to reduce these problems to simpler settings.

Corollary 4.17 (Propagation of totality/determinism)**.** *Let G be a graph. If G is not bi-connected,*

1. *$eval_G$ is total $\iff \forall G' \in BiCC(G)$, $eval_{G'}$ is total*
2. *$eval_G$ is deterministic $\iff \forall G' \in BiCC(G)$, $eval_{G'}$ is deterministic.*

If G is bi-connected, let $\mathcal{P} = \{M_1, M_2, \ldots, M_n\}$ be its partition into maximal proper modules. We have:

1. *$eval_G$ is total $\iff$ $eval_{G/\mathcal{P}}$ and $eval_{M_i}$, for $i = 1, \ldots, n$, are total.*
2. *$eval_G$ is deterministic $\iff$ $eval_{G/\mathcal{P}}$ and $eval_{M_i}$, for $i = 1, \ldots, n$, are deterministic.*

Proof. We prove the results for non-bi-connected graphs by strong induction on $|BiCC(G)|$. For the base case, $|BiCC(G)| = 1$, the statements are vacuously true. We consider G with $|BiCC(G)| \geq 2$ and suppose that for all G' such that $|BiCC(G')| < |BiCC(G)|$, the statements are true. We decompose $G = G_1 \overset{\star}{\text{---}} G_2$, where $\star \in \{\vee, \wedge\}$.

Suppose that $eval_G$ is deterministic and that $\star = \vee$. We prove that $eval_{G_1}$ is total. Observe that if X is some subset of Var, and we define $X' = X \setminus V(G_2)$, we have $eval_{G_2}(X', 0)$. Consequently, by Proposition 4.4, we have

$$\begin{cases} eval_{G_1}(X', 0) \iff eval_G(X', 0) \\ eval_{G_1}(X', 1) \iff eval_G(X', 1) \end{cases}$$

And by definition of evaluation, we have that

$$\begin{cases} eval_{G_1}(X, 0) \iff eval_{G_1}(X', 0) \\ eval_{G_1}(X, 1) \iff eval_{G_1}(X', 1) \end{cases}$$

since the vertex sets of G_1 and G_2 are disjoint. Therefore by totality of $eval_G$ at X' we have totality of $eval_{G_1}$ at X. The argument is similar when $\star = \wedge$, but we take $X' = X \cup V(G)$. Similarly for the totality of $eval_{G_2}$.

Now we can apply the inductive hypothesis to G_1 and G_2; this gives that evaluation is total at each of the bi-connected components of G_1 and at each of the bi-connected components of G_2. Since $BiCC(G) = BiCC(G_1) \cup BiCC(G_2)$, the result follows.

Conversely, suppose that evaluation is total at each of the bi-connected components of G. We then have, by inductive hypothesis, that evaluation is total at G_1 and at G_2. We use Proposition 4.4 to conclude by exhaustion of cases.

We now prove the result for determinism. Note that by Proposition 3.6, it is equivalent to prove that G is CIS if and only if its bi-connected components are CIS.

Suppose that G is CIS, let $\bar{\star} \in \{\vee, \wedge\}$, and let $S \in MC_\star(G_1), T \in MC_{\bar{\star}}(G_1)$. By Lemma 4.3, we have that $T \in MC_{\bar{\star}}(G)$, and that for any $S' \in MC_\star(G_2)$, $S' \cup S \in MC_\star(G)$. So, for any such S' we have by hypothesis that $T \cap (S' \cup S) \neq \emptyset$. However, since $T \cap V(G_2) = \emptyset$, we have that $T \cap S \neq \emptyset$, and so G_1 is CIS. Similarly, G_2 is CIS, and so we can apply the inductive hypothesis to G_1 and G_2 and conclude as before with determinism.

Conversely, if every bi-connected component of G is CIS, we have by inductive hypothesis that G_1 and G_2 are CIS, and we use Lemma 4.3 to conclude by exhaustion of cases.

Now we take G a bi-connected graph, $\mathcal{P}$ its modular decomposition.

We prove the result for totality by contraposition:

Suppose that $eval_{G/\mathcal{P}}$ is not total. This means there exists $W \subseteq \mathcal{P}$ such that $\neg eval_{G/\mathcal{P}}(W, 1)$ and $\neg eval_{G/\mathcal{P}}(W, 0)$. Let $X = \varphi^{-1}(W)$. We now prove that $W = Y_{X,G} = Z_{X,G}$. Let $M \in W$. Then $M \subseteq X$, so $eval_M(X, 1)$ and $\neg eval_M(X, 0)$. Conversely, let $M \in Y_{X,G}$ (resp. $M \in Z_{X,G}$). Then $eval_M(X, 1)$, so $\exists S$ a maximal $\wedge$-clique of M with $S \subseteq X$ (resp. then $\neg eval_M(X, 0)$, so $\forall T$ maximal $\vee$-clique of M, $T \cap X \neq \emptyset$) and therefore there exists at least one element of M in X, so $M \in \varphi(X) \subseteq W$. So we have $W = Y_{X,G} = Z_{X,G}$, and to conclude that $eval_G$ is not total, we apply the contrapositive of the statements in Proposition 4.15.

Suppose there exists some $M_i \in \mathcal{P}$ and $X \subseteq \mathit{Var}$ such that $\neg eval_{M_i}(X, 1)$ and $\neg eval_{M_i}(X, 0)$. Let

$$\mathcal{C} = \{M_j \in \mathcal{P} | M_i \text{---} M_j\}; P_1 = \bigcup_{M \in \mathcal{C}} M$$
$$\mathcal{D} = \{M_j \in \mathcal{P} | M_i \cdots M_j\}; P_2 = \bigcup_{N \in \mathcal{D}} N$$
$$X' = X \cup P_1 \setminus P_2$$

Note that since the maximal proper modules are disjoint, we have

$$\neg eval_{M_i}(X', 1) \iff \neg eval_{M_i}(X, 1)$$
$$\neg eval_{M_i}(X', 0) \iff \neg eval_{M_i}(X, 0).$$

Suppose for contradiction that $eval_G(X', 0)$. Then there exists a maximal $\vee$-clique T of G such that $T \cap X' = \emptyset$. Then $T \cap P_1 = \emptyset$, so $T \subseteq M_i \cup P_2$, *i.e.* $T \in MC_\vee(M_i \cdots P_2)$. By 4.3, $T = T_1 \cup T_2$, $T_1 \in MC_\vee(M_i)$, $T_2 \in MC_\vee(P_2)$. This means that $eval_{M_i}(X', 0)$ since $T_1 \cap X = \emptyset$. By the earlier remark, we therefore have $eval_{M_i}(X, 0)$, which contradicts the original assumption. By a similar argument, we obtain a contradiction if $eval_G(X', 1)$. Therefore $eval_G$ is not total.

Now suppose that evaluation is total at every module and at the quotient graph, and let $X \subseteq \mathit{Var}$. If $eval_{G/\mathcal{P}}(Y_{X,G}, 1)$, by Proposition 4.15, $eval_G(X, 1)$. If not, then by totality at the quotient graph, we have $eval_{G/\mathcal{P}}(Y_{X,G}, 0)$, *i.e.* $\exists T \in MC_\vee(G/\mathcal{P}), T \cap Y_{X,G} = \emptyset$. Therefore, $\forall M \in T$, $\neg eval_M(X, 1)$, and so $eval_M(X, 0)$ by totality at the modules. This means that for all $M \in T$, $M \notin Z_{X,G}$, *i.e.* $T \cap Z_{X,G} = \emptyset$, so $eval_{G/\mathcal{P}}(Z_{X,G}, 0)$, and Proposition 4.15 allows us to conclude that $eval_G(X, 1)$. So either way, $eval_G$ is defined at X.

Notice that by Proposition 3.6, the following is an equivalent statement:

$$G \text{ is CIS} \iff \begin{cases} \forall i, M_i \text{ is CIS} \\ \text{and} \\ G/\mathcal{P} \text{ is CIS} \end{cases}$$

We assume that G has the CIS property, so we immediately get that $G/\mathcal{P}$ is CIS since the homomorphism induced by $\mathcal{P}$ preserves maximal $\vee$- and $\wedge$-cliques. Let $M \in \mathcal{P}$, and $S \in MC_\wedge(M), T \in MC_\vee(M)$. Let $S' \supseteq S$, resp. $T' \supseteq T$, be a maximal $\wedge$- (resp. $\vee$-) clique of G which extends S (resp. T). We know that $S' \cap T' \neq \emptyset$. The single point in that intersection must be in M, for if not, that point has a $\wedge$-neighbour and a $\vee$-neighbour in M, contradicting that M is a module. So $S \cap T \neq \emptyset$, and we can conclude that M is CIS.

Conversely, let $S \in MC_\wedge(G), T \in MC_\vee(G)$. By hypothesis and by Lemma 4.13, we know that $\varphi(S) \cap \varphi(T)$ is non-empty. Let $M \in \varphi(S) \cap \varphi(T)$; we have that $S \cap M \in MC_\wedge(M)$, and $T \cap M \in MC_\vee(M)$. By hypothesis we get $(S \cap M) \cap (T \cap M) \neq \emptyset$ so $S \cap T \neq \emptyset$. So G is CIS.

□

Thus, the evaluation algorithm we found reduces not only the problem of evaluating a graph, but also the problem of proving the totality or determinism of its evaluation relation, to prime graphs, namely the ones occurring in its modular decomposition tree.

5. Conclusions

5.1. **Summary.** Drawing from previous work on graph representations of linear terms [3], we have formalised notions from Boolean logic in a purely graph theoretical setting. We have examined properties of evaluation in this setting, and proved a coherency between evaluation and entailment. Furthermore, we have taken advantage of the modular decomposition of bi-connected graphs to provide a correct algorithm for evaluation. This also reduced questions of totality and determinism of evaluation to simpler contexts, namely the prime graphs. Finally, we briefly discuss some other work done during the internship.

5.2. **Other work.** The work presented above is an extension of the correspondence between *linear* terms and co-graphs. We also developed a theory of *non-linear* graphs, in which variables can have several occurences. This was acheived by associating a vertex-labelling map to the graph, which associates the vertices with variables. The linear setting can be seen as a special case of this, in which the labelling map is injective. However, in the non-linear setting, the elements of cliques are vertices, because we can no longer identify them with variables. We therefore generalised the notions of evaluation and entailment to use the images of maximal cliques by the labelling map, rather than the maximal cliques themselves. As in the linear case, we were able to prove soundness and completeness for these generalisations of entailment and evaluation.

While we were able to prove a generalisation of Proposition 4.4, we were not able to extend Corollary 4.17. This is due to the fact that the variable sets of disjoint subgraphs need not be disjoint in the non-linear setting (*cf.* Proof of Corollary 4.17). This being said, our main motivation for studying the non-linear case was to develop a *complete proof system* for graphs. It turns out we can easily duplicate variables in a graph to obtain a normal form which is analogous to the disjunctive normal form of a term. The difficulty in using this to construct proofs is in the formalisation of a 'merging' of variables, a rule akin to $x \vee x \to x$ in the term setting, since the occurences of x may have distinct neighbourhoods. While there are certain conditions that allow us to soundly implement such a merging rule, a deep study of this phenomenon is yet to be carried out, and could form the basis of future work.

References

[1] D Andrade, Endre Boros, and Vladimir Gurvich. On graphs whose maximal cliques and stable sets intersect. Technical report, 2006.

[2] Yves Crama and Peter L Hammer. *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press, 2011.

[3] Anupam Das and Lutz Straßburger. No complete linear term rewriting system for propositional logic. In Maribel Fernández, editor, *26th International Conference on Rewriting Techniques and Applications (RTA 2015)*, volume 36 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 127–142, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[4] Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007.

[5] Michel Habib and Christophe Paul. A survey on algorithmic aspects of modular decomposition. *CoRR*, abs/0912.1457, 2009.

[6] Lutz Straßburger. A characterisation of medial as rewriting rule. In Franz Baader, editor, *RTA 2007*, volume 4533 of *LNCS*, pages 344–358. Springer-Verlag, 2007.