ÉCOLE POLYTECHNIQUE
IP PARIS

MODULAR DECOMPOSITION FOR LOGIC

An interactive tool for the modular decomposition of graphs

Rémy Seassau

November 27, 2022

INTRODUCTION

- There exists a well know correspondence between formulae and *cographs*

An example:

$$P = (a \otimes b) \parr (c \otimes d)$$

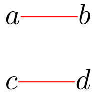
INTRODUCTION

- There exists a well know correspondence between formulae and *cographs*

An example:

$$P = (a \otimes b) \parr (c \otimes d)$$

G_P :



THE QUESTION

This correspondence has been extended into a proof system on *arbitrary undirected graphs* called GS (graphic proof system) [1].

Current work is being done on more logic systems based on graphs.

This motivates the question, how does one study a graph's structure?

THE ANSWER

Such a graph theoretical tool exists, and is called the **modular decomposition** of a graph.

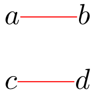
Modules are to a graph what subformulas are to a formula.

THE ANSWER

Such a graph theoretical tool exists, and is called the **modular decomposition** of a graph.

Modules are to a graph what subformulas are to a formula.

Back to our example:



PROBLEM DEFINITION

Given a graph, can we provide a nice way for logicians (or anyone else) to obtain it's modular decomposition?

WHAT DO WE MEAN BY GRAPH?

- Directed or undirected?

WHAT DO WE MEAN BY GRAPH?

- Directed or undirected? Both!

WHAT DO WE MEAN BY GRAPH?

- Directed or undirected? Both!
- Do we allow self loops?

WHAT DO WE MEAN BY GRAPH?

- Directed or undirected? Both!
- Do we allow self loops? No

WHAT DO WE MEAN BY GRAPH?

- Directed or undirected? Both!
- Do we allow self loops? No
- Are we restricted to cographs?

WHAT DO WE MEAN BY GRAPH?

- Directed or undirected? Both!
- Do we allow self loops? No
- Are we restricted to cographs? No

WHAT DO WE MEAN BY GRAPH?

- Directed or undirected? Both!
- Do we allow self loops? No
- Are we restricted to cographs? No
- What kind of vertices?

WHAT DO WE MEAN BY GRAPH?

- Directed or undirected? Both!
- Do we allow self loops? No
- Are we restricted to cographs? No
- What kind of vertices? Labelled

GRAPH DEFINITION

A graph $G = (V_G, E_G)$ is an ordered pair where V_G is a set of vertices and E_G is a set of pairs of elements of V_G.

GRAPH DEFINITION

A graph $G = (V_G, E_G)$ is an ordered pair where V_G is a set of vertices and E_G is a set of pairs of elements of V_G.

If G is undirected (resp. directed) then the pairs in E_G are unordered (resp. ordered).

GRAPH DEFINITION

A graph $G = (V_G, E_G)$ is an ordered pair where V_G is a set of vertices and E_G is a set of pairs of elements of V_G.

If G is undirected (resp. directed) then the pairs in E_G are unordered (resp. ordered).

We say that G is L-labelled if there exists an injection $l_G\colon V_G \to L$.

MODULE DEFINITION

A module M of a graph G is a subgraph of G, i.e. $(V_M, E_M) \subseteq (V_G, E_G)$ such that

MODULE DEFINITION

A module M of a graph G is a subgraph of G, i.e. $(V_M, E_M) \subseteq (V_G, E_G)$ such that

$$\forall x, y \in V_M, \forall z \in V_G \setminus V_M :$$
$$(x, z) \in E_G \iff (y, z) \in E_G \quad \text{and} \quad (z, x) \in E_G \iff (z, y) \in E_G$$

The edge set of M is the largest subset of E_G that contains all vertices in V_M.

MODULE DEFINITION

A module M of a graph G is a subgraph of G, i.e. $(V_M, E_M) \subseteq (V_G, E_G)$ such that

$$\forall x, y \in V_M, \forall z \in V_G \setminus V_M :$$
$$(x, z) \in E_G \iff (y, z) \in E_G \quad \text{and} \quad (z, x) \in E_G \iff (z, y) \in E_G$$

The edge set of M is the largest subset of E_G that contains all vertices in V_M.

The trivial modules of G are the empty graph $(\varnothing, \varnothing)$, the graphs where V_M are singletons and G itself. A module M of G is maximal if the only module M' such that $V_M \subseteq V'_M$ is G itself.

PRIME GRAPHS

We say that a graph G is prime if $|V_G| \geq 2$ and all of the modules of G are trivial.

PRIME GRAPHS

We say that a graph G is prime if $|V_G| \geq 2$ and all of the modules of G are trivial.

Here are all the directed prime graphs with $|V_G| = 2$:

COMPOSITION OF GRAPHS

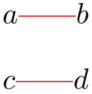
Let G be a graph with n vertices $V_G = \{v_1, \ldots, v_n\}$ and let $H_1, \ldots, H_n$ be n graphs such that $\forall i, j < n, i \neq j \implies V_{H_i} \cap V_{H_j} = \varnothing$.
The **composition of $\mathbf{H_1}, \ldots, \mathbf{H_n}$ via G** is the graph $G(|H_1, \ldots, H_n|)$ where each vertex v_i of G has been replaced by the graph H_i.

COMPOSITION OF GRAPHS

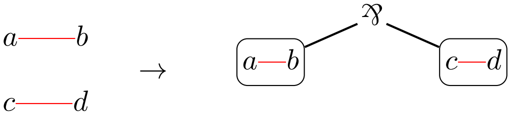
Let G be a graph with n vertices $V_G = \{v_1, \ldots, v_n\}$ and let $H_1, \ldots, H_n$ be n graphs such that $\forall i, j < n, i \neq j \implies V_{H_i} \cap V_{H_j} = \varnothing$.
The **composition of $\mathbf{H_1}, \ldots, \mathbf{H_n}$ via G** is the graph $G(|H_1, \ldots, H_n|)$ where each vertex v_i of G has been replaced by the graph H_i.

Theorem [2]: Let G be a graph such that $|V_G| = n \geq 2$. Then, there are non-empty graphs $H_1, \ldots, H_n$ and a prime graph P such that $G = P(|H_1, \ldots, H_n|)$.

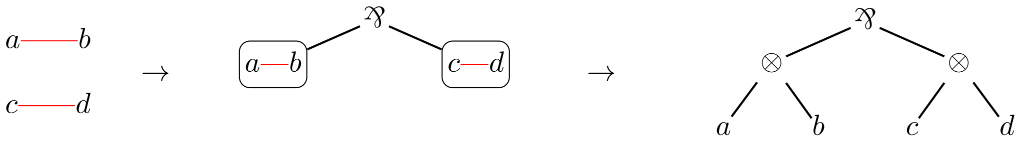
CONSTRUCTING A MODULAR DECOMPOSITION TREE



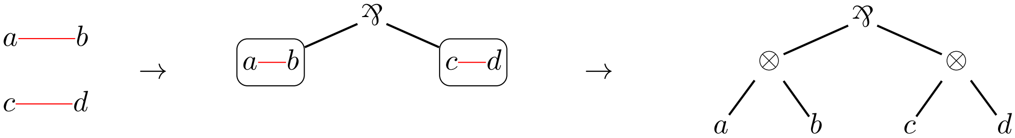
CONSTRUCTING A MODULAR DECOMPOSITION TREE



CONSTRUCTING A MODULAR DECOMPOSITION TREE



CONSTRUCTING A MODULAR DECOMPOSITION TREE



Which is the same as

$$G_P \quad \to \quad ⅋(\boxed{a \text{—} b}, \boxed{c \text{—} d}) \quad \to \quad ⅋(\otimes(a, b), \otimes(a, b))$$

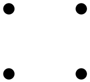
CONNECTIVES

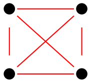
Recall the prime graphs of size 2:

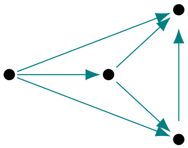
We introduce a family of graphs we call the connectives. These generalize the prime graphs of size 2 to graphs of size n.

- $⅋_n$: The graph of n vertices with no edges
- $\otimes_n$: The graph of n vertices where all edges are connected
- $\triangleleft_n$: The graph of n vertices where the edges form a transitive chain going through all of the vertices

EXAMPLES

$\parr_4$: 

$\otimes_4$: 

$\lhd_4$: 

COMPUTING THE MODULAR DECOMPOSITION

```
condensing = true;
while condensing do
    do
        prevGraph = graph;
        graph = compressConnectives(graph);
    while prevGraph ≠ graph;
    graph = compressSmallestMaximalModules(graph);
    if length(V_graph) == 1 then
        condensing = false;
    end
end
return graph;
```

Algorithm 1: Modular Decomposition Overview [3]

FINDING THE CONNECTIVES

Finding the connective modules is quite easy, we just iterate over all possible pair of vertices of the graph v_i, v_j and check that the following holds:

$$\mathrm{suc}(v_i) \setminus \{v_j\} = \mathrm{suc}(v_j) \setminus \{v_i\} \text{ and } \mathrm{pred}(v_i) \setminus \{v_j\} = \mathrm{pred}(v_j) \setminus \{v_i\}$$

If so, we just need to check if the edges (v_i, v_j) and/or (v_j, v_i) exists to determine the type of connective ($\parr$, $\otimes$ or $\triangleleft$) the vertices are composed by.

FINDING THE SMALLEST MAXIMAL MODULES

In order to find the smallest maximal modules of a graph, we find all of the maximal modules with at least 2 vertices and take the smallest non-intersecting ones.

We thus iterate over all sets of vertices for which there exists an edge and follow these steps:

1. We look at the vertices that are connected to one but not all of the vertices in our set
2. If there are none then we are done, otherwise add them to our set and repeat step 1

REFERENCES

[1] M. Acclavio, R. Horne, and L. Straßburger. Logic beyond formulas: a proof system on graphs. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 38–52. ACM, July 8, 2020.

[2] A. Ehrenfeucht, T. Harju, and G. Rozenberg. *The Theory of 2-Structures: A Framework for Decomposition and Transformation of Graphs*. WORLD SCIENTIFIC, Aug. 1999.

[3] L. James, R. Stanton, and D. Cowan. Graph decomposition for undirected graphs. *Utilitas Mathematica*, Jan. 1, 1972.