HAL open science

The atomic decomposition of strongly connected graphs

Bruno Courcelle

To cite this version:

Bruno Courcelle. The atomic decomposition of strongly connected graphs. 2013. hal-00875661

HAL Id: hal-00875661

https://hal.science/hal-00875661v1

Preprint submitted on 22 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The atomic decomposition of strongly connected graphs

Bruno Courcelle
LaBRI, Bordeaux University and CNRS
Email : courcell@labri.fr

July 2, 2013

Abstract

We define and study a new canonical decomposition of strongly connected graphs that we call the *atomic decomposition*. We can construct it in linear time from the decomposition in 3-connected components of the considered graph. In a companion article, we use it to characterize, up to homeomorphism, all closed curves in the plane having a given Gauss word.

1 Introduction

There are many types of hierarchical graph decompositions. They are useful for studying the structure and properties of certain graphs and for building efficient graph algorithms. These algorithms exploit the tree structure of the considered decomposition of the input graph. Here are some examples. The *modular decomposition* [MoRad] (also called *substitution decomposition*) has been introduced by Gallai [Gal] for studying comparability graphs and their transitive orientations. The *split decomposition* introduced by Cunnigham [Cun] (its definition is in Section 4) helps to understand and recognize circle graphs [Cou08, GPTCb]. *Tree-decompositions* are essential for the proof by Robertson and Seymour of the Graph Minor Theorem (see [Die] for an overview). They yield the notion of *tree-width*, an integer graph invariant that can be used as parameter in many fixed parameter tractable (FPT) algorithms (see the books [CouEng, DF, FG]). *Clique-width* is another integer graph invariant based on hierchical decompositions, that can also be used as parameter in FPT algorithms [CouEng, CMR]. Each type of decomposition is based on a set of elementary graphs and of operations on graphs. A decomposition of a graph is an expression of this graph in terms of these elementary graphs and operations (hence a labelled tree). Here are examples of graph operations used in decompositions : substitution to a vertex, disjoint union, concatenation of two graphs at distinguished vertices.

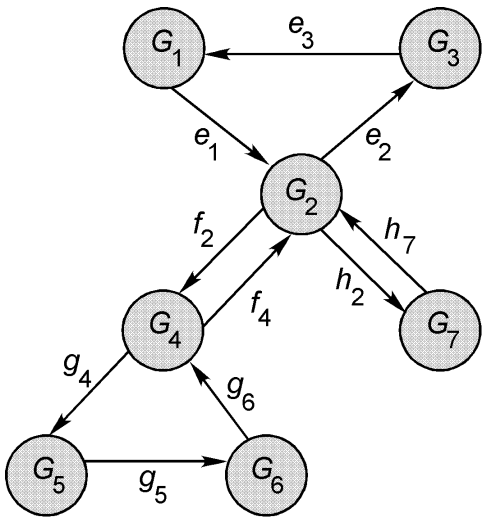

Figure 1: An atomic decomposition.

A decomposition is *canonical* if it is unique for each graph. The modular decomposition, the split decomposition and the decomposition of a graph in 3-connected components ([Tut], we call it the *Tutte decomposition,* its exact definition is recalled in Section 2.1) are canonical. They exhibit a certain intrinsic structure of the considered graphs. The decompositions related to tree-width and clique-width are not, and they are also more difficult to construct than the three mentioned canonical ones as they raise NP-complete problems (see [CouEng] for references).

We define and study a new canonical decomposition of strongly connected (directed) graphs and of 2-edge connected undirected graphs (they are the undirected graphs having a strongly connected orientation). We call it the *atomic decomposition.* The underlying composition operation is as follows : if G and H are disjoint directed graphs, $e : x \to y$ is an edge of G and $f : u \to v$ is one of H, then $K := G \boxplus_{e,f} H$ is the union of G and H where e is redirected towards v (hence $e : x \to v$ in K) and f is redirected towards y. A typical example of atomic decomposition is shown in Figure 1. The graphs $G_1, ..., G_7$ are atoms, i.e., they cannot be decomposed.

The underlying structure is not a tree. It is a strongly connected graph whose biconnected components are directed cycles (hence, "almost" a tree). We call it a *cactus.* Using a cactus instead of a tree is important to obtain a canonical expression, because the composition operation that we use satisfies the following *circular associativity*:

$$(G_1 \boxplus_{e_1,e_2} G_2) \boxplus_{e_2,e_3} G_3 = (G_3 \boxplus_{e_3,e_1} G_1) \boxplus_{e_1,e_2} G_2.$$

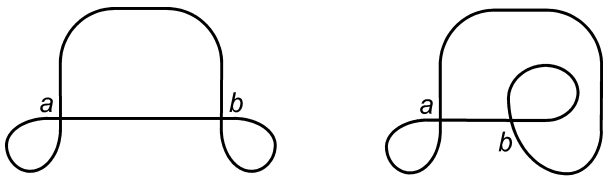

Figure 2: Curves with Gauss word *aabb*.

A cactus represents a class of the congruence relation on expressions generated by circular associativity. In a similar way, in presence of an associative binary operation, we can get canonical terms by considering this operation as of variable arity. The corresponding trees (the syntactic trees of terms) are unranked and ordered (no fixed outdegree and a linear order on sons of a node).

The atomic decomposition is based on a set of bipartitions of the vertex set of a strongly connected graph, as is the split decomposition of the same graphs [Cun]. It is different but can be seen as a preliminary step in the construction of the split decomposition. The Tutte decomposition is also related to the atomic decomposition, and this fact yields a linear time algorithm for constructing the latter.

The definition of atomic decompositions is motivated by the study of *Gauss words*: they are finite words that encode the *self-intersections of closed curves in the plane* (with no triple intersection). Each crossing is named by a letter, and a word with two occurrences of each letter is obtained by following the curve and writing the letter seen at each crossing. These words have been characterized in several ways (See [FOM, LM, Ros] and the book [GodRoy]). However, the following question does not seem to have been considered: What is the common structure of all curves having a same associated word ? A word is *unambiguous* if it characterizes a unique curve up to homeomorphism. Otherwise, it is *ambiguous*. Figure 2 shows two curves that are not related by any homeomorphism of the plane or even of the sphere but have the same ambiguous Gauss word *aabb*. The word *abcabc* is unambiguous and Figure 3 shows the corresponding curve.

For studying these questions, we first observe that self-intersecting closed curves without triple intersections are plane 4-regular graphs, that we will describe combinatorially (up to homeomorphisms) by maps. We recall that a *map* is a connected graph equipped with a circular order of edges around each vertex called a *rotation* (see the book by Mohar and Thomassen [MT], Chapter 3). It represents an embedding of the graph in a surface. Hence, intersecting closed curves can be described up to homeomorphism by 4-regular planar maps. The atomic decomposition works for graphs, for maps and for 4-regular graphs and maps. Roughly speaking, a strongly connected 4-regular graph can be associated with a Gauss word w, and from the atomic decomposition of this graph,

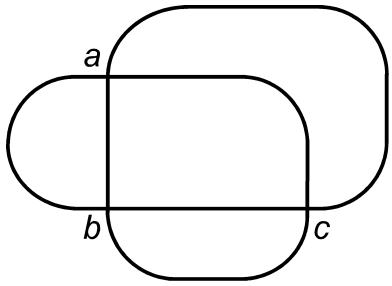

Figure 3: Curve with Gauss word *abcabc*

one can describe combinatorially all closed curves whose Gauss word is w. (We do this in [Cou13]).

The article is organized as follows. Section 2 reviews definitions about graphs and maps. Section 3 defines atomic decompositions of graphs and related maps. Section 4 is a conclusion where we review related decompositions of graphs (in particular, one defined by Knuth [Knu]), knot diagrams and matroids. We also list some open questions. For the reader's convenience, an appendix reviews the various equivalence and isomorphism notions used in this article.

2 Definitions

All graphs, trees and related objects will be finite. By saying that $(e_1, ..., e_k)$ is a *circular sequence*, we mean that it can also be specified as $(e_{i+1}, ..., e_k, e_1, ..., e_{i-1})$ and that its properties and associated constructions do not depend on the initial element e_1.

A class of an equivalence relation is *nontrivial* if it has at least two elements. Otherwise, we call it a *singleton* class.

2.1 Graphs

A directed graph G is defined as a triple $(V_G, E_G, vert_G)$ consisting the set of vertices V_G, the set of edges E_G (with $V_G \cap E_G = \emptyset$) and a mapping $vert_G : E_G \to V_G \times V_G$ that defines incidences. If $vert_G(e) = (x, y)$, we say that x is the *tail* of e, denoted by $\alpha(e)$, that y is its *head*, denoted by $\beta(e)$, we also write $e : x \to y$ and we say that x and y are the *ends* of e. If G is undirected, then $vert_G(e)$ is a set $\{x, y\}$ of one or two vertices, called the *ends* of e and we write $e : x - y$. In both cases, e is a *loop* if $x = y$. We say that an edge *links* two disjoint sets of vertices if it has one end in each of them. Two directed or undirected edges are *parallel* if they have the same ends. A graph without

parallel edges is *simple*. A *bond* is a loop-free graph with 2 vertices and at least 2 parallel edges. (Some books, e.g. [Die], use this term for a different notion). A *k-bond* is a bond with k edges. A graph, directed or not, is *k-regular* if each vertex has degree k, where an incident loop is counted for 2.

We denote by $Und(G)$ the undirected graph obtained from a directed one by taking as incidence function $vert_{Und(G)}(e) = \{x, y\}$ whenever $vert_G(e) = (x, y)$. Note that we *do not identify* an edge with the pair or the set of its ends.

Subdividing an edge $e : x \to y$ (or $e : x - y$) consists in replacing it by a directed path of length 2 from x to y (or by an undirected path of length 2 linking x and y). The opposite operation is the *fusion of two consecutive edges*. It is nothing but the contraction of one of two such edges.

We call *nodes* the vertices of a tree, and we adapt notation accordingly. This convention is useful when we discuss simultaneously a graph and a tree representing a decomposition of this graph.

Walks and paths.

Let G be a graph and x,$y \in V_G$. A *walk* from x to y is a sequence $(x_0, e_1, x_1, e_2, ..., e_n, x_n)$ such that $x_0, x_1, ..., x_n \in V_G$, $x_0 = x$, $x_n = y$, $e_1, ..., e_n \in E_G$, $e_i : x_{i-1} \to x_i$ ($e_i : x_{i-1} - x_i$ if G is undirected) for each $i = 1, ..., n$, and $e_i \neq e_j$ if $1 \leq i < j \leq n$. It is a *path* if we (also) have $x_i \neq x_j$ for $0 \leq i < j \leq n$, except possibly if $i = 0$ and $j = n$. A walk is *closed* if $x_0 = x_n$. A *circuit* is a closed path in a directed graph. A *cycle* is similar in an undirected graph (a cycle with two vertices consists of two parallel edges). An edge never occurs twice in a walk. A vertex never occurs twice in a path except if $x_0 = x_n$. A walk $(x_0, e_1, x_1, e_2, ..., e_n, x_n)$ of a directed graph can be described without ambiguity by the sequence $(e_1, e_2, ..., e_n)$. A walk, a path or a circuit in a directed graph is said to be *undirected* if its edges can be traversed in any direction (i.e., with $e_i : x_{i-1} \to x_i$ or $e_i : x_i \to x_{i-1}$ in the above definition).

4-regularity.

A 4-*regular graph* is a graph whose vertices have all degree 4. A (2,2)-*regular graph* is a directed, 4-regular graph, each vertex of which has 2 incoming edges and 2 outgoing edges. For convenience, we include connectedness in the notation: $\mathcal{G}_4$ denotes the class of connected and undirected 4-regular graphs and $\mathcal{G}_{2,2}$ denotes the class of connected (2,2)-regular graphs. Each graph in $\mathcal{G}_4 \cup \mathcal{G}_{2,2}$ has an *Eulerian tour, i.e.*, is covered by a closed walk (*covered* means that the walk goes through all edges); the walk is directed if the graph is. (See [Die], Section 1.8, where the proof given for undirected graphs extends easily to directed ones). Every graph in $\mathcal{G}_{2,2}$ is strongly connected. Every graph in $\mathcal{G}_4$ is $Und(G)$ for some $G \in \mathcal{G}_{2,2}$.

Subgraphs

We write $G \subseteq H$ (resp. $G \subseteq_i H$) if G is a subgraph (resp. an induced subgraph) of H. If $F \subseteq V_G \cup E_G$, then $G - F$ is the subgraph of G obtained by deleting the edges and vertices in F and the edges incident with a vertex in F. We write it $G - x$ if $F = \{x\}$. If $X \subseteq V_G$, we denote by $G[X]$ the graph $G - (V_G - X)$: it is the induced subgraph of G with vertex set X.

Union and intersection

Let H_1 and H_2 be two subgraphs of a graph G. Their *intersection* $H_1 \cap H_2$ is the subgraph K of G (and of H_1 and H_2) such that $V_K = V_{H_1} \cap V_{H_2}$ and $E_K = E_{H_1} \cap E_{H_2}$. Their *union* $H_1 \cup H_2$ is the subgraph L of G such that $V_L = V_{H_1} \cup V_{H_2}$ and $E_L = E_{H_1} \cup E_{H_2}$. We have $H_1 \subseteq L$ and $H_2 \subseteq L$. Two graphs H_1 and H_2 are *disjoint* if $V_{H_1} \cap V_{H_2} = \emptyset$ and $E_{H_1} \cap E_{H_2} = \emptyset$.

Graph gluings

Let $G = H_1 \cup H_2$ with $E_{H_1} \cap E_{H_2} = \emptyset$. We write $G = H_1//_u H_2$ if $V_{H_1} \cap V_{H_2} = \{u\}$. If neither H_1 nor H_2 is reduced to u, we say that u is a *separating vertex* of G. Similarly, we write $G = H_1//_{u,v} H_2$ if $G = H_1 \cup H_2$, $E_{H_1} \cap E_{H_2} = \emptyset$, $V_{H_1} \cap V_{H_2} = \{u, v\}$ and $u \neq v$. If H_1 and H_2 are connected and have at least 3 vertices, we say that $\{u, v\}$ is a *separating pair*. In both cases, we say that G is a *gluing* of H_1 and H_2.

Let G be the union of pairwise edge disjoint graphs H_1,...,H_k, $k \geq 3$. We write $G = \bigodot_{u_1,...,u_k}(H_1, ..., H_k)$ if:

$$V_{H_i} \cap V_{H_{i+1}} = \{u_{i+1}\} \text{ for each } i = 1, ..., k-1,$$
$$V_{H_1} \cap V_{H_k} = \{u_1\},$$
$$V_{H_i} \cap V_{H_j} = \emptyset \text{ if } \mid i - j \mid \geq 2 \text{ and } \{i, j\} \neq \{1, k\}.$$

We will say in this case that G is a *circular gluing* of H_1,...,H_k.

Connectivity

Following [Die], we say that a graph G is k-connected if $|V_G| \geq k + 1$ and $G - X$ is connected whenever X is a set of at most $k - 1$ vertices. A connected graph with at least 3 vertices is 2-connected if and only if it is not of the form $H_1//_u H_2$ where H_1 and H_2 have at least 2 vertices. A graph with a loop may be 2-connected and be of the form $H_1//_u H_2$ where H_1 is a loop on u. A 2-connected graph with at least 4 vertices is 3-connected if and only if it has no separating pair. A graph G is *k-edge connected* if $G - F$ is connected for every set F of at most $k - 1$ edges.

A graph is *strongly connected* if it is directed and every two vertices belong to a closed walk. The corresponding undirected graphs are the 2-edge connected graphs, equivalently, the connected graphs without bridges (by classical results due to Robbins and Menger; see [BanGut], Theorem 1.6.2 and [Die], Theorem 3.3.6; a *bridge* in G is an edge e such that $G - e$ has one more connected component than G). We denote respectively by $\mathcal{SC}$ and $2\mathcal{EC}$ the classes of strongly connected graphs and of undirected 2-edge connected graphs having

at least one edge. The graphs with one vertex and at least one loop belong to these classes.

The tree of 2-connected components of a connected graph

The *2-connected components* of a loop-free graph, i.e., its maximal 2-connected subgraphs form a tree whose nodes are these components and the separating vertices.

A *cactus* is a loop-free connected graph whose 2-connected components are cycles if the graph is undirected, and circuits if it is directed. (Some authors allow a cactus to have bridges, but we do not). A graph reduced to one vertex is a cactus, and a cactus with two vertices consists of two parallel edges. A loop-free directed graph G is a cactus if and only if, for any two distinct vertices x and y, there is a unique (directed) path from x to y. (Similarly, a loop-free undirected graph is a tree if and only if any two distinct vertices are linked by a unique path). The condition is clearly necessary. For proving the other direction, observe that G is strongly connected, hence has an *ear decomposition* (Theorem 7.2.2 of [BanGut]). This means that G can be constructed from a circuit by adding successively directed paths (new vertices and new edges) linking two existing vertices, and circuits containing exactly one existing vertex. We use an induction on the number of such addition steps. Each intermediate graph is strongly connected and the last step cannot be the addition of a path from u to $v \neq u$ because there would exist two directed paths from u to v. Hence it is the addition of a circuit to a graph G'. This graph satisfies also the condition that there is a unique directed path from x to y for any two distinct vertices x and y. Hence G' is a cactus by induction and so is G.

Canonical decomposition of a 2-connected loop-free graph

We review the definition of a decomposition defined by Tutte [Tut]. As in [Cou99] (and in [Die], Chap.12, Exercise 20), we define it in terms of tree decompositions. (A tree-decomposition of a graph G is a pair (T, f) where T is an undirected tree and f is a mapping from N_T the set of nodes of T, to $\mathcal{P}(V_G)$ satisfying the three well-known conditions). Let G be loop-free and 2-connected. Let (T, f) be a tree-decomposition of this graph satisfying the following conditions :

T1: $N_T = N_{2T} \cup N_{3T}$ where N_{2T} is the set of nodes x such that $|f(x)| = 2$ and N_{3T} is the set of those such that $|f(x)| \geq 3$.

T2 : the nodes in N_{2T} have degree at least 2; no two nodes, both in N_{2T} or both in N_{3T}, are adjacent.

T3 : The *boxes* $f(x)$ (for $x \in N_T$) are pairwise different and $f(x) \subset f(y)$ if $x \in N_{2T}$ is adjacent to y (thus $y \in N_{3T}$).

For every $x \in N_{3T}$, we denote by $B(x)$ the simple undirected graph with vertex set $f(x)$ such that u and v in $f(x)$ are adjacent if and only if they are adjacent in G or $f(y) = \{u, v\}$ for some node y adjacent to x. Then, we require the following additional conditions:

T4 : Each graph $B(x)$ is 3-connected or is a cycle with at least 3 vertices.

T5 : If $x \in N_{2T}$ has exactly two neighbours y and z (both in N_{3T} by T2), if u and v such that $f(x) = \{u, v\}$ are not adjacent in G, then $B(y)$ and $B(z)$ are not both cycles.

The graph G must have at least 3 vertices. No decomposition is needed for graphs with one or two vertices. Note that $B(x)$ is not always a subgraph of G because edge directions are omitted and $B(x)$ may have edges $u - v$ such that u and v are not adjacent in G. Such an edge of $B(x)$ is called a *virtual edge*, as it does not "come from" any edge of G. A theorem of [Tut] (Chap. IV, see also [CunEdm]) establishes that every loop-free 2-connected graph has a unique tree-decomposition (T, f) satisfying these conditions (up to isomorphism, i.e., up to the denotation of the nodes of T). We call it the *Tutte decomposition* of G. The graphs $B(x)$ are called the 3-*blocks* of G. They are minors of $Und(G)$. The graphs $B(x)$ that are cycles are the *cycle-blocks*. This decomposition can be constructed in linear time ([HopTar2]).

Figure 4 shows a 4-regular graph with vertex set $\{a, b, ..., n\}$ and Figure 5 shows its Tutte decomposition. The edges $a - b$ and $b - k$ in Figure 5 are virtual edges, shown by dotted lines. The edges $e - f$ and $m - l$ are not virtual, but they are shown by dotted lines in boxes C and G because the corresponding edges of the graphs are located in boxes D and F. In Figure 5, each edge of G is put in a unique box of the tree-decomposition; we represent it also with dotted lines in the neighbour boxes. The cycle-blocks correspond (after fusion of parallel edges) to the boxes B, C, F and G. The 3-connected blocks correspond similarly to D and E. To make the figure simpler, the nodes of N_{2T} of degree 2 are omitted (these nodes should be between C and D, E and F and F and G).

It is easy to transform a tree-decomposition satisfying Conditions T1-T4 into one satisfying also T5: if $x \in N_{2T}$ has exactly two neighbours y and z such that $B(y)$ and $B(z)$ are cycles and the two vertices in $f(x)$ are not adjacent in G, then one can fuse x, y and z into a single node and merge $B(y)$ and $B(z)$ into a single cycle. This transformation preserves the validity of T1-T4 and can be repeated until T5 holds.

Let (T, f) be a tree decomposition of a graph G, and x, y be adjacent nodes of T. We denote by $T(x; y)$ the subtree of $T - x$ that contains y, and by $V(x; y) \subseteq V_G$ the union of the sets $f(z)$ such that z is a node of $T(x; y)$. Let us now assume that (T, f) is the Tutte decomposition of G. For each $x \in N_{3T}$ such that $B(x)$ is a k-cycle, we get an expression of G of the form $\bigodot_{u_1,...,u_k}(H_1, ..., H_k)$. In this expression, $H_i = G[V(x; y)]$ where y is the neighbour of x such that $f(y) = \{u_i, u_{i+1}\}$ (is $\{u_1, u_k\}$ if $i = k$). The graph H_i is connected and, unless it is a single edge, its vertices u_i and u_{i+1} have degree at least 2. Similarily, for each $x \in N_{2T}$, we get an expression of G of the form $H_1//_{u_1,u_2}H_2//_{u_1,u_2}\cdots //_{u_1,u_2}H_k$, where $k \geq 2$ is the degree of x and $f(x) = \{u_1, u_2\}$. The graphs H_i are connected but u_i and u_{i+1} may have degree 1 in H_i.

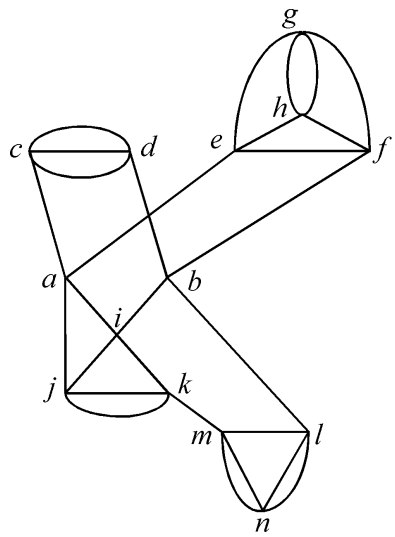

Figure 4: A 4-regular graph G

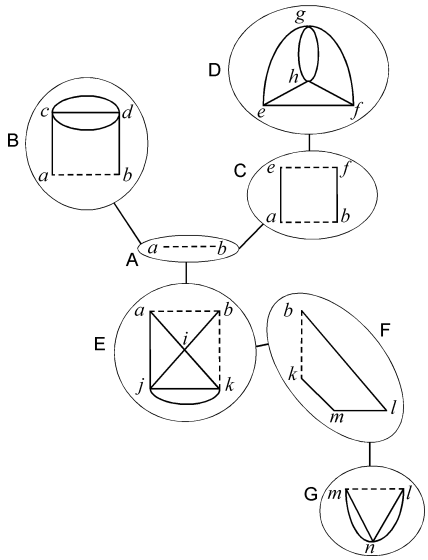

Figure 5: The Tutte decomposition of G of Figure 5.

Sizes

As we will state algorithmic results, we must specify the size measures of inputs and outputs : graphs and their decompositions.

We define the size of a graph G as $\|G\| = |V_G| + |E_G|$, i.e., as the total number of vertices and edges. For graphs of bounded degree and for trees, we take only the number of vertices or nodes, which is possible without modifying the notion of a linear-time algorithm since, for such graphs, $\|G\| = O(|V_G|)$.

We define the size of a tree-decomposition (T, f) of a graph G as:

$$\|(T, f)\| = |N_T| + \sum_{x \in N_T} |f(x)| + |E_G|$$

where. Since we allow graphs with parallel edges, the number of edges is not bounded in terms of the sum of sizes of boxes. Clearly, $\|G\| \leq \|(T, f)\|$. Every tree-decomposition can be transformed into one of the same graph having the same width and whose tree has at most $|V_G|$ nodes (see, e.g., Chapter 2 of [CouEng]). For such a decomposition of width k, we have :

$$\|(T, f)\| \leq \|G\| + (k+1).\,|V_G| = O(\|G\|) \text{ for fixed } k.$$

The case of square grids shows that we do not have $\|(T, f)\| = O(\|G\|)$ for graphs of unbounded tree-width.

For the Tutte decomposition (T, f) of a 2-connected loop-free graph G, we define:

$$\|(T, f)\|_1 = \|(T, f)\| + \sum_{x \in N_{3T}} \left|E_{B(x)}\right|$$

which is larger than $\|(T, f)\|$ because we also count the edges of the blocks $B(x)$. Some algorithms use the virtual edges, hence we count them in all boxes where they occur. So, we have $\|G\| \leq \|(T, f)\| \leq \|(T, f)\|_1$. We will prove that :

$$\|(T, f)\|_1 \leq 2.\,\|G\| + 9.\,|V_G| = O(\|G\|),$$

so that $\|(T, f)\|_1$ can be replaced by $\|G\|$ in upper-bounds to computation times of algorithms. (This is so because Tutte decompositions are tree-decompositions with strong constraints.)

It is clear that $|N_{2T}| < |N_{3T}| \leq |V_G|$. Hence, $|N_T| < 2.\,|V_G|$.

A vertex of G is *shared* if it belongs to several boxes $f(x)$, which implies that it belongs to at least one box $f(x)$ with $x \in N_{2T}$. An edge e of a graph $B(x)$ is *shared* if its two ends u and v belong to several boxes $f(x)$, and this implies that $\{u, v\} = f(x)$ for a unique node $x \in N_{2T}$. We let $f'(x)$ be the set of shared vertices of $f(x)$ and $E'_{B(x)}$ be the set of shared edges of $B(x)$.

We must bound the sum $\sum_{x \in N_T} |f(x)| + \sum_{x \in N_{3T}} \left|E_{B(x)}\right|$. The total number of the vertices and edges of the sets $f(x)$ and $B(x)$ that are not shared is

$$\sum_{x\in N_T} |f(x)-f'(x)| + \sum_{x\in N_{3T}} \left|E_{B(x)} - E'_{B(x)}\right|,$$

clearly at most $\|G\|$. Every shared vertex that belongs to $f(x)$ for $x \in N_{3T}$ is the end of a shared edge of the corresponding graph $B(x)$. Hence,

$$\sum_{x\in N_T} |f'(x)| = \sum_{x\in N_{2T}} |f'(x)| + \sum_{x\in N_{3T}} |f'(x)|$$
$$\leq 2.\,|N_{2T}| + 2.\sum_{x\in N_{3T}} \left|E'_{B(x)}\right|.$$

Finally, $\sum_{x\in N_{3T}} \left|E'_{B(x)}\right|$ is the number of pairs (e,x) such that e is a shared edge that belongs to $E_{B(x)}$. But these pairs are in bijection with the edges of T, which gives $\sum_{x\in N_{3T}} \left|E'_{B(x)}\right| = \left|E'_{B(x)}\right| - 1$. So we get:

$$\|(T,f)\|_1 = |N_T| + |E_G| + \sum_{x\in N_T} |f(x)| + \sum_{x\in N_{3T}} \left|E_{B(x)}\right|$$
$$\leq 2.\,|V_G| + |E_G| + \|G\| + 2.\,|N_{2T}| + 2.(|N_T|-1) + |N_T| - 1.$$
$$\leq 2.\,\|G\| + |V_G| + 2.\,|V_G| + 4.\,|V_G| + 2.\,|V_G| = 2.\,\|G\| + 9.\,|V_G|\,.$$

Isomorphisms

An *isomorphism* of $G = (V_G, E_G, vert_G)$ to $G' = (V_{G'}, E_{G'}, vert_{G'})$ is a bijection $h : V_G \cup E_G \to V_{G'} \cup E_{G'}$ that maps vertices to vertices, edges to edges and preserves incidences, that is : $vert_{G'}(h(e)) = (h(x), h(y))$ (resp. $\{h(x), h(y)\}$) if $vert_G(e) = (x,y)$ (resp. $vert_G(e) = \{x,y\}$). It is a *v-isomorphism* if $V_G = V_{G'}$. In this case, one can consider E_G and $E_{G'}$ as different set of names used to designate the edges of a graph with vertex set V_G. If $G' = G$, we get the notions of *automorphism* and of *v-automorphism*. The graph of Figure 2 has several v-automorphisms. We denote by $G \cong G'$ the existence of an isomorphism between G and G'.

An *abstract graph* is the equivalence class, denoted by $[G]_{iso}$, of a graph G. In this article, unless otherwise specified, graphs (and related relational structures) will be concrete, *i.e.*, will be defined with precise sets of vertices and edges, and not up to isomorphism. (See [CouEng] for more details on abstract graphs and relational structures).

An appendix reviews the different notions of isomorphism and equivalence relations used in this article.

2.2 Maps

Maps are combinatorial objects that represent embeddings of connected graphs in oriented surfaces, up to orientation preserving homeomorphisms. We review

the classical definitions (cf. [MT], Chapter 3 for detailed definitions), and we introduce some new notions.

Embeddings

If $\mathcal{E}$ is an embedding of a graph G in a surface, we denote by $\mathcal{E}(u)$ the point representing a vertex u, by $\mathcal{E}(e)$ the curve segment representing an edge e, by $\mathcal{E}(W)$ the union of the curve segments representing the edges of a walk W.

Darts.

To formalize planar embeddings, it is useful to split each directed edge e into two *darts* e^-, e^+ with incidences defined by a function γ such that $\gamma(e^-) = x$ and $\gamma(e^+) = y$ if $e : x \to y$. We denote by D_G^+ the set of darts e^+, by D_G^- the set of darts e^- and by D_G the set $D_G^+ \cup D_G^-$. It is clear that $D_G^+ \subseteq D_{G'}^+$ and $D_G^- \subseteq D_{G'}^-$ if $G \subseteq G'$.For an undirected graph H, we fix an arbitrary orientation of its edges and we specify from it the darts e^- and e^+. (The "half-edges" designated by e^+ and e^- depend thus on the chosen orientation.)

Maps

A *map* is a pair $M = (G, \rho)$ consisting of a connected and directed graph G and a bijection $\rho : D_G \to D_G$ such that, for every d in D_G, the set $\{\rho^i(d) \mid i \geq 0\}$ is the set of darts incident with $\gamma(d)$. This bijection is called the *rotation* of M.

From an embedding $\mathcal{E}$ of a connected and directed graph G in an orientable surface, we get a map (G, ρ) by letting $\rho(d)$ be the dart following d in the circular order, "around the vertex $\gamma(d)$" (and according to the orientation of the surface) of the darts incident with $\gamma(d)$. For any two embeddings of G in the sphere with same associated map, there is an orientation preserving homeomorphism of the sphere that maps $\mathcal{E}$ to $\mathcal{E}'$. (See [MT], Theorem 3.2.4 for the proof, and a more general statement concerning orientable surfaces.).

If $\mathcal{E}$ is an embedding of a connected and directed graph G in the plane with map $M = (G, \rho)$ and $\mathcal{E}'$ is the embedding of G that is the image of $\mathcal{E}$ by a symmetry relative to a straight line, then the map of $\mathcal{E}'$ is the *symmetric map* of M, defined as $M^{-1} = (G, \rho^{-1})$.

We say that two maps M and M' are *equivalent* if $M' = M$ or $M' = M^{-1}$. They represent homeomorphic embeddings of a same graph.

We denote respectively by $\mathcal{M}$, $\mathcal{MSC}$ and $\mathcal{M}_{2,2}$ the classes of all maps, of those of graphs in $\mathcal{SC}$ and of graphs in $\mathcal{G}_{2,2}$.

Submaps

A map $M = (G, \rho)$ is a *submap* of $M' = (G', \rho')$ (denoted by $M \subseteq M'$) if $G \subseteq G'$ and, for every dart d of G, $\rho(d) = \rho'^i(d)$ where i is the smallest positive integer such that $\rho'^i(d) \in D_M$. By using $\subseteq_i$ instead of $\subseteq$, we get the notion of an *induced submap*. Since the union and the intersection of two graphs are defined in terms of subgraphs, we get the corresponding notions for maps.

Undirected maps

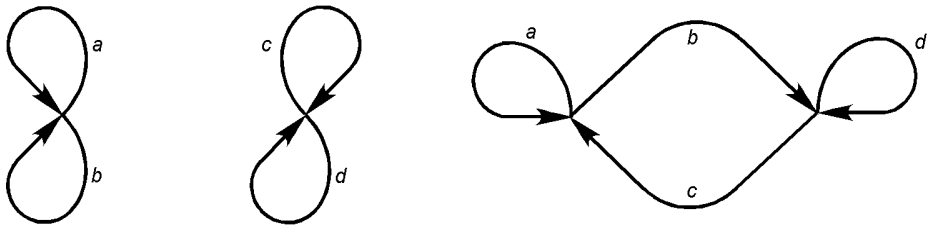

Figure 6: Three planar maps $M, N, P \in \mathcal{M}_{2,2}$

A map of an undirected graph H is a map (G, ρ) such that $H = Und(G)$. Two maps $M = (G, \rho)$ and $M' = (G', \rho')$ of an undirected graph H are *Und-equivalent*, if there exists a bijection $h : D_G \to D_{G'}$ such that $\{h(e^+), h(e^-)\} = \{e^+, e^-\}$ for every $e \in E_G$ and $\rho'(h(d)) = h(\rho(d))$ for every $d \in D_G$. Formally, a map of an undirected connected graph is an equivalence class of maps with respect to Und-equivalence. We call it an *undirected map.*

We denote respectively by $\mathcal{UM}$, $\mathcal{M}2\mathcal{EC}$ and $\mathcal{M}_4$ the class of all undirected maps, of those of graphs in $2\mathcal{EC}$ and of graphs in $\mathcal{G}_4$. Since every graph H in $\mathcal{G}_4$ (resp. in $2\mathcal{EC}$) is $Und(G)$ for some graph G in $\mathcal{G}_{2,2}$ (resp. in $\mathcal{SC}$), the maps of $\mathcal{M}_4$ (resp. $\mathcal{M}2\mathcal{EC}$) are equivalence classes of maps in $\mathcal{M}_{2,2}$ (resp. in $\mathcal{MSC}$).

We denote by $\mathcal{P}2\mathcal{EC}$ and $\mathcal{PG}_4$ the classes of planar undirected graphs in $2\mathcal{EC}$ and $\mathcal{G}_4$ respectively, and by $\mathcal{PM}2\mathcal{EC}$ and $\mathcal{PM}_4$ the corresponding classes of undirected maps.

3 Atomic decompositions of graphs and maps

We define a decomposition of strongly connected graphs and prove that it is canonical. Then we extend the definition to maps either planar or not.

3.1 Circular composition of directed graphs and maps

We define an operation that composes graphs and maps. We will actually use it mainly to *decompose* these objects. We denote respectively by $\mathcal{D}$ and $\mathcal{U}$ the classes of directed and undirected graphs.

Definition 1 : *Circular composition.*

We let $G_1, ..., G_k$ be paiwise disjoint graphs in $\mathcal{D}$ and e_i be an edge of G_i for each i. We define $\boxplus_{e_1,...,e_k}(G_1, ..., G_k)$ as the graph H such that $V_H = V_{G_1} \cup ... \cup V_{G_k}$, $E_H = E_{G_1} \cup ... \cup E_{G_k}$ and the incidence function $vert_H$ is defined as follows (α and β are defined in Section 2.1):

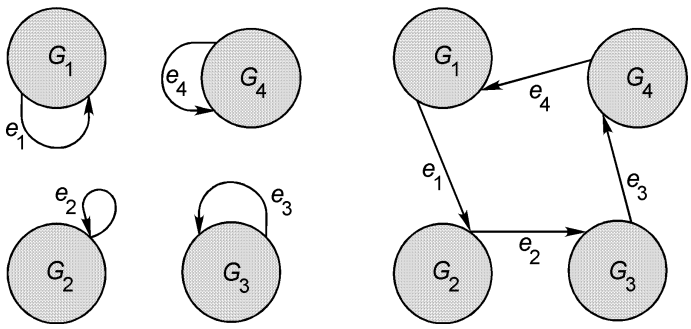

Figure 7: G (to the right) is the circular composition of $G_1, ..., G_4$.

$vert_H(e_i) = (\alpha(e_i), \beta(e_{i+1}))$ if $1 \leq i \leq k-1$,
$vert_H(e_k) = (\alpha(e_k), \beta(e_1))$,
$vert_H(e) = vert_{G_i}(e)$ if $e \in E_{G_i} - \{e_i\}, 1 \leq i \leq k$.

We call H a *circular composition of* $G_1, ..., G_k$. If $k = 2$, we will also write $G_1 \boxplus_{e_1,e_2} G_2$ instead of $\boxplus_{e_1,e_2}(G_1, G_2)$.

Here are examples. We denote respectively by G, H, K the graphs (of the maps M, N, P) of Figure 6. We have $K = G \boxplus_{b,c} H$. The left part of Figure 7 shows graphs $G_1, ..., G_4$ with distinguished edges $e_1, ..., e_4$ (note that e_2 is a loop). The right part part shows $\boxplus_{e_1,...,e_4}(G_1, G_2, G_3, G_4)$.

In this article, we will only apply this operation to strongly connected graphs, leaving for future work its application to other types of graphs. It is clear that H is strongly connected if $G_1, ..., G_k$ are so. We now consider how circular composition acts on certain subclasses of $\mathcal{SC}$ and on related classes of maps.

We recall that $\mathcal{G}_{2,2} \subseteq \mathcal{SC}$. If $G_1, ..., G_k$ are connected and (2,2)-regular (i.e., if they belong to $\mathcal{G}_{2,2}$), then so is their circular composition. For pairwise disjoint maps $M_1, ..., M_k$, we define a map $N = \boxplus_{e_1,...,e_k}(M_1, ..., M_k)$ with underlying graph $H = \boxplus_{e_1,...,e_k}(Graph(M_1), ..., Graph(M_k))$ and rotation ρ_N defined as follows:

$\rho_N(e_i^+) = \rho_{M_{i+1}}(e_{i+1}^+)$ if $1 \leq i \leq k-1$,
$\rho_N(e_k^+) = \rho_{M_1}(e_1^+)$,
$\rho_N(d) = e_{i-1}^+$ if $1 \leq i \leq k, d \in D_{M_i} - \{e_i^+\}$ and $\rho_{M_i}(d) = e_i^+$,
$\rho_N(d) = e_k^+$ if $d \in D_{M_1} - \{e_1^+\}$ and $\rho_{M_1}(d) = e_1^+$,
$\rho_N(d) = \rho_{M_i}(d)$ if $1 \leq i \leq k$, $d \in D_{G_i}$ and the above cases do not apply.

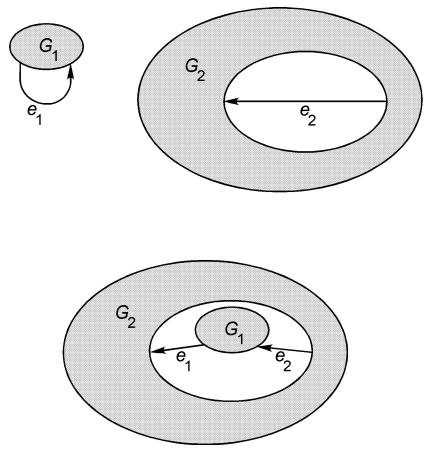

Figure 8: Composition of planar maps

It is clear that N is a map. If $M_1, ..., M_k$ belong to $\mathcal{M}_{2,2}$, then N belongs to the same class. For the maps of Figure 6, we have the equality $P = M \boxplus_{b,c} N$. Figure 7 shows the circular composition of four planar maps represented in the plane, with distinguished edges drawn "on the outer face". Figure 8 shows the composition $G_1 \boxplus_{e_1,e_2} G_2$ of two planar maps G_1 and G_2 , represented similarly, but with e_2 not on the outer face.

In all these cases, circular composition is a partial operation because of the disjointness conditions on the arguments. It can be made total in the usual way: if graphs or maps are not disjoint, one replaces them by isomorphic copies that meet the requested disjointness conditions. So, the result is well-defined only up to isomorphism, hence on *abstract* graphs and maps, *i.e.*, on isomorphism classes of such objects. (See Chapter 2 of [CouEng] for more details on abstract graphs and related notions.) In the present article, we will use circular composition to *decompose given maps and graphs*. That is, for given H, we will try to find graphs (or maps) $G_1, ..., G_k$ such that $H = \boxplus_{e_1,...,e_k}(G_1, ..., G_k)$. Hence, we will not need to take isomorphic copies.

What about undirected graphs?

We could define $G = H \boxplus_{e,f} K$ if H and K are undirected as $Und(H' \boxplus_{e,f} K')$ where H' and K' are orientations of H and K. However, G is not well-defined because we get (in general) two different graphs. In order to get a unique graph, we could use an operation $\boxplus_{e,u,f,w}$ where u is the end of e in H that becomes in G an end of f, and w is the end of f in K that becomes an end of e. This is too complicated and we prefer to define only circular compositions of directed graphs (that can be orientations of undirected graphs). However, the two graphs of the initial definition of $H \boxplus_{e,f} K$ have the same *cycle matroid*. Circular composition is actually well-defined on matroids. See Section 4 on this point.

Proposition 2 :
(1) Let M be a map (resp. a map in $\mathcal{M}_{2,2}$) such that

$$Graph(M) = \boxplus_{e_1,...,e_k}(H_1, ..., H_k)$$

for some graphs $H_1, ..., H_k$. There are unique rotations $\rho_1, ..., \rho_k$ such that

$$M = \boxplus_{e_1,...,e_k}(N_1, ..., N_k)$$

where, for each i, $N_i = (H_i, \rho_i)$ is a map (resp. is in $\mathcal{M}_{2,2}$).
(2) If $G = \boxplus_{e_1,...,e_k}(H_1, ..., H_k)$ where $G, H_1, ..., H_k$ are graphs or maps, then G is planar if and only if $H_1, ..., H_k$ are.

Proof: Assertion (1) are easy to prove from the definitions.
(2) For proving the "if" direction, we observe that a planar embedding of the graph or map $\boxplus_{e_1,...,e_k}(H_1, ..., H_k)$ can be built by combining planar embeddings of $H_1, ..., H_k$ such that the edges $e_1, ..., e_k$ are on the outer face.
For the "only if" direction, we observe that if $G = \boxplus_{e_1,...,e_k}(H_1, ..., H_k)$ then each H_i can be obtained from G by a sequence of deletions of edges and of isolated vertices and of fusions of consecutive edges. These operations preserve planarity. (See [CouDus] for the general notion of a *map minor*, of which we use here a particular case). The result follows. $\square$

We now state some equational properties of circular composition that are valid for directed graphs and maps.

Proposition 3 : Let $G_1, ..., G_k$ be pairwise disjoint directed graphs or maps, and $e_i \in E_{G_i}$ for each i. We have the following equalities (in each case, both handsides are defined):

(1) $G_1 \boxplus_{e_1,e_2} G_2 = G_2 \boxplus_{e_2,e_1} G_1$,
(2) $(G_1 \boxplus_{e_1,e_2} G_2) \boxplus_{f,e_3} G_3 = G_1 \boxplus_{e_1,e_2} (G_2 \boxplus_{f,e_3} G_3)$ if f is an edge of G_2, $f \neq e_2$,
(3) $(G_1 \boxplus_{e_1,e_2} G_2) \boxplus_{e_2,e_3} G_3 = (G_3 \boxplus_{e_3,e_1} G_1) \boxplus_{e_1,e_2} G_2$.

Let $H = \boxplus_{e_1,...,e_k}(G_1, ..., G_k)$. Then, we have :
(4) $H = (...(G_1 \boxplus_{e_1,e_2} G_2) \boxplus_{e_2,e_3} G_3)... \boxplus_{e_{k-1},e_k} G_k)$,

Proof: Straightforward from Definition 1.$\square$

Property (3) is called: *circular associativity*. From these equalities, we get the following ones:

Corollary 4 : With the same hypotheses as in Proposition 3:

(1) $G_1 \boxplus_{e_1,e_2} (G_2 \boxplus_{e_2,e_3} G_3) = G_3 \boxplus_{e_3,e_1} (G_1 \boxplus_{e_1,e_2} G_2)$,

(2) $H = \boxplus_{e_i,...,e_k,e_1,...,e_{i-1}}(G_i, ..., G_k, G_1, ..., G_{i-1})$ for $2 \leq i \leq k$,

(3) $H = G_1 \boxplus_{e_1,e_k} (\boxplus_{e_2,...,e_k}(G_2, ..., G_k))$,

(4) $H = \boxplus_{e_1,...,e_i}(G_1, ..., G_i) \boxplus_{e_i,e_k} (\boxplus_{e_{i+1},...,e_k}(G_{i+1}, ..., G_k))$

for $2 \leq i \leq k$.

To illustrate Property (4) consider the graph H at the right of Figure 7: we have $H = \boxplus_{e_1,...,e_4}(G_1, G_2, G_3, G_4) = (G_1 \boxplus_{e_1,e_2} G_2) \boxplus_{e_2,e_4} (G_3 \boxplus_{e_3,e_4} G_4)$.

From now on, we will only consider strongly connected graphs, their maps, and the associated undirected graphs and maps.

Terms defining graphs and maps.

Definition 5: *Atoms and terms.*

An *atom* is a graph G in $\mathcal{SC}$ that cannot be decomposed as $G = G_1 \boxplus_{e_1,e_2} G_2$. An *atomic* map is a map whose underlying graph is an atom. By Proposition 2, it cannot be expressed as a composition (by $\boxplus$) of two maps. The graph $Graph(M) = Graph(N)$ where M, N are at the left of Figure 6 and the graph H to the left of Figure 12 (cf. Section 3.2) are atoms.

Let $\mathcal{C}$ be a set of pairwise disjoint graphs or maps. A *term over* $\mathcal{C}$ is a term t built with cicular composition and elements of $\mathcal{C}$ used as constants, each of them having at most one occurrence in the term. It defines a graph or a map, depending on the types of the elements of $\mathcal{C}$. We denote this object by $val(t)$: it is the *value* of t. We say that t *uses* $\mathcal{C}$ if each element of $\mathcal{C}$ has one (and only one) occurrence in t.

Every strongly connected graph or map G can be defined by a term over atoms (of the corresponding type): if it is defined by a term t using $G_1, ..., G_k$ such that G_i is not an atom, then $G_i = H \boxplus_{h,k} K$ and G_i can be replaced in t by $H \boxplus_{h,k} K$. By starting from the trivial term reduced to G, and by repeating this splitting step at most $|V_G|$ times, we obtain a term over atoms that defines G. We get in this way a polynomial-time algorithm for constructing t from G, but we will give below a linear-time one.

Proposition 3 shows that different terms (even over atoms) can have the same value. We will say that two such terms are *equivalent*. For example, the following two terms define the graph shown in Figure 9:

$$t = \boxplus_{e_1,e_2,e_3}(G_1, [(G_2 \boxplus_{h_2,h_7} G_7) \boxplus_{f_2,f_4} (\boxplus_{g_4,g_5,g_6}(G_4, G_5, G_6)], G_3))$$
$$t' = G_7 \boxplus_{h_7,h_2} [\boxplus_{e_1,e_2,e_3}(G_1, G_2, G_3) \boxplus_{f_2,f_4} (\boxplus_{g_4,g_5,g_6}(G_4, G_5, G_6))].$$

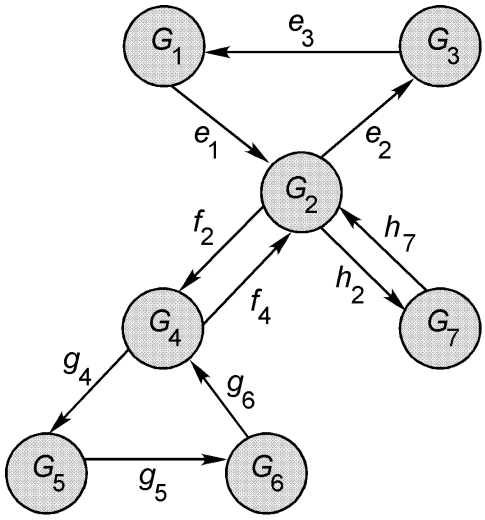

Figure 9: The graph definied by term t.

We will first prove that every graph G in $\mathcal{SC}$ can be expressed by a term using a unique set of atoms. Then, we will define a canonical representation of G, not by a term, but by a cactus (cf. Section 2.1), whose vertices are labelled by the atoms of G. These results extend to maps by Definition 1 and Proposition 2.

3.2 Atomic sets of vertices and atoms

We will prove that every two terms over atoms that define the same strongly connected graph G use the same set of atoms, and we will characterize these atoms directly from G, independently of any decomposition process.

Definition 6: *Splits, 2-cuts and atomic skeletons.*

(a) Let G be any graph, directed or not. A set of vertices X is a *split* if there are exactly two edges linking this set and its complement. (Hence $V_G - X$ is also a split.) The two edges linking X and $V_G - X$ form a 2-*cut*, and are called 2-*cut edges*. Two vertices x and y are *separated* by a split X if $x \in X$ and $y \notin X$ or vice versa. They are *equivalent*, which we denote by $x \sim_G y$, if no split separates them. This is an equivalence relation. (For transitivity, let $x \sim_G y$ and $y \sim_G z$. If a split X contains x and not z, then either $y \in X$ and $y \nsim_G z$ or $y \notin X$ and $x \nsim_G y$. Both cases are impossible, hence $x \sim_G z$.) Its equivalence classes form the *atomic partition* of V_G, denoted by $\mathcal{A}(G)$ and are the *atomic* sets of vertices. It is clear that an atomic set is the intersection of all splits that contain it.

More generally, a split X *separates* two nonempty sets U and V if $U \subseteq X$ and $V \subseteq V_G - X$ or vice versa. (The splits defined here are not those of Cunnigham [Cun]; in Section 4, we will compare the different notions.)

(b) We now use these notions for $G \in \mathcal{SC}$. Let X be a split. One of the two associated edges is directed from X to $V_G - X$ and the other from to $V_G - X$ to X. Furthermore the graphs $G[X]$ and $G[V_G - X]$ are connected. (They may be reduced to one vertex possibly with loops).

It is clear that if $G = H \boxplus_{e,f} K$, then V_H is a split with 2-cut $\{e, f\}$ such that e is directed from V_H to $V_K = V_G - V_H$ and f from V_K to $V_H = V_G - V_K$. Conversely, if X is a split of G, we define $G\{X\}$ as the graph $G[X]$ augmented with the edge e whose head is redefined as that of f. We call e the *handle of* $G\{X\}$. It is clear that $G = G\{X\} \boxplus_{e,f} G\{V_G - X\}$. Hence, the splits of a graph G are in bijection with its expressions of the form $H \boxplus_{e,f} K$.

Note also that $G \in \mathcal{SC}$ is an atom if and only if it has no split. Hence, it is an atom if and only if it has no 2-cut, if and only if it is 3-edge connected (by a theorem by Menger, see [Die]). We will investigate in more detail the structure of atoms in Section 3.5.2.

(c) Let $G \in \mathcal{SC}$, let E_G^{cut} be the set of its 2-cut edges, i.e., of those whose two ends are not equivalent.

The *atomic skeleton of* G is the graph $Sk(G) = (\mathcal{A}(G), E_G^{cut}, vert_{Sk(G)})$ whose vertices are the atomic sets of vertices, whose edges are the 2-cut edges and whose incidence function is defined as follows, for every e in E_G^{cut}:

> $vert_{Sk(G)}(e) = (X, Y)$ if and only if X and Y are the (distinct) atomic sets containg respectively the vertices u and v such that $vert_G(e) = (u, v)$.

If G' is any other strongly connected orientation of $Und(G)$, then $\mathcal{A}(G) = \mathcal{A}(G')$ because the definition of a split does not depend on the orientation of the considered graph, and so $\sim_G = \sim_{G'}$. It follows that $Und(Sk(G)) = Und(Sk(G'))$. Hence, the *atomic skeleton of* a graph $H \in 2\mathcal{EC}$ can be defined as $Sk(H) = Und(Sk(G))$ where G is any strongly connected orientation of H. $\square$

The main technical result on which is based the notion of atomic decomposition is the following one:

Theorem 7: The atomic skeleton of a strongly connected or 2-edge connected graph is a cactus.

Proof: We first consider the case of a graph G in $\mathcal{SC}$. We let X and Y be two distinct atomic sets of vertices.

First, we prove that there is a path in $Sk(G)$ from X to Y. Let $u \in X$ and $v \in Y$. There is in G a path from u to v. First we contract the edges of this path whose two ends are in a same atomic set. Then we replace every vertex by the atomic set containing it. We get a walk in $Sk(G)$ from X to Y. By deleting from it some closed subwalks, we get a path from X to Y.

If $Sk(G)$ is not a cactus, there are distinct atomic sets X and Y with several paths from X to Y. Let X and Y be two such sets with a path $P = (Z_0, e_1, Z_1, e_2, ..., e_p, Z_p)$ such that $Z_0 = X$, $Z_p = Y$ (we may have $p = 1$) and another path $P' = (W_0, f_1, W_1, f_2, ..., f_q, W_q)$ in $Sk(G)$ such that $W_0 = X$ and $W_q = Y$. Let us furthermore assume that P is of minimal length among all such paths, for all atomic sets X and Y. There is a split S such that $\alpha(e_1) \in S$ and $\beta(e_p) \in V_G - S$. Each atom $Z_0, Z_1, ..., Z_p, W_0, ..., W_q$ is contained either in S or in $V_G - S$. Hence some edge e_i goes from S to $V_G - S$. Similarly, some edge f_j goes from S to $V_G - S$. Hence we must have $e_i = f_j$ because S is a split, hence there are no two edges from S to $V_G - S$. Hence we have $Z_{i-1} = W_{j-1}$ and $Z_i = W_j$.We cannot have $p = 1$ because then $P = P'$. If $i = 1$, then P is not of minimal length because it can be replaced by $(Z_1, e_2, ..., e_p, Z_p)$. Similarly $i \neq p$, hence $1 < i < p$. We must have $(Z_0, e_1, ..., e_{i-1}, Z_{i-1}) = (W_0, f_1, ..., f_{j-1}, W_{j-1})$ otherwise P is not of minimal length. Similarly, $(Z_i, e_{i+1}, ..., e_p, Z_p) = (W_j, f_{j+1}, ..., f_q, W_q)$. But then, $P = P'$. We get again a contradiction, which shows that $Sk(G)$ is a cactus.

If $H \in 2\mathcal{EC}$, then $Sk(H)$ defined as $Und(Sk(G))$ where G is a (any) strongly connected orientation of H, is also a cactus.$\square$

A graph isomorphic to $Sk(G)$ is obtained by fusing any two $\sim_G$-equivalent vertices and by removing all loops (those resulting from the fusions and the ones of G).

In order to relate $Sk(H \boxplus_{e,f} K)$ to $Sk(H)$ and $Sk(K)$, we need some lemmas, and first an observation. If e, f are two distinct edges of $G \in \mathcal{SC}$ that are not loops, we define $W(e, f)$ as the set of vertices on the walks from $\beta(e)$ to $\alpha(f)$ that contain neither e nor f. Since e and f belong to a closed walk, $W(e, f)$ is not empty and it contains $\beta(e)$ and $\alpha(f)$. It is clear that $V_G = W(e, f) \cup W(f, e)$, and that $\{e, f\}$ is a 2-cut if and only if $W(e, f) \cap W(f, e) = \emptyset$.

Lemma 8 : Let $G \in \mathcal{SC}$ be of the form $H \boxplus_{e,f} K$. A set X is a split of H if and only if it is nonempty, not V_H and is $Y \cap V_H$ for some split Y of G.

Proof: Let X is a split of H. We have $H = H_1 \boxplus_{h_1,h_2} H_2$ where $X = V_{H_1}$. Hence $G = (H_1 \boxplus_{h_1,h_2} H_2) \boxplus_{e,f} K$. There are four cases.

If $e \in E_{H_2} - \{h_2\}$, then, by Proposition 3(2), $G = H_1 \boxplus_{h_1,h_2} (H_2 \boxplus_{e,f} K)$ and X is a split of G, the conclusion holds.

If $e = h_2 \in E_{H_2}$, then, by Proposition 3(3), $G = (K \boxplus_{f,h_1} H_1) \boxplus_{h_1,e} H_2$ and $X = V_H \cap (V_G - V_{H_2})$ where $V_G - V_{H_2}$ is a split of G, hence the conclusion holds.

If $e \in E_{H_1} - \{h_1\}$, then, by Proposition 3(1,2), $G = H_2 \boxplus_{h_2,h_1} (H_1 \boxplus_{e,f} K)$ and $X = V_H \cap (V_G - V_{H_2})$ where $V_G - V_{H_2}$ is a split of G and the conclusion holds.

If $e = h_1 \in E_{H_1}$, then, by Proposition 3(1,3), $G = (K \boxplus_{f,h_2} H_2) \boxplus_{h_2,e} H_1$ and X is a split of G, the conclusion holds.

For proving the other direction, we let $G = H \boxplus_{e,f} K$ and Y be a split of G such that $X = Y \cap V_H$ is not empty and properly included in V_H. We let $\{g, g'\}$ be the 2-cut corresponding to Y. We distinguish several cases.

Case 1 : $\{g, g'\} \cap \{e, f\} = \emptyset$ and $g \in E_H, g' \in E_K$.

Every closed walk containing e and f must contain g and g', and vice versa. The sets $W(f, g)$ and $W(g, e)$ form a partition of $W(f, e) = V_H$, the sets $W(e, g')$ and $V(g', f)$ form a partition of $W(e, f) = V_K$. The set Y is either $W(g', f) \cup W(f, g)$ or $W(g, e) \cup W(e, g')$. Then X is either $W(f, g)$ or $W(g, e)$ and, in both cases, it is a split of H.

Case 2 : $\{g, g'\} \cap \{e, f\} = \emptyset$ and $g, g' \in E_H$. There is a closed walk containing e and f and also g and g'. Without loss of generality, we assume that we have f, g, g' and e in this order on this walk. Then, Y is either $W(g, g')$ (included in V_H) or $W(g', g)$. In the first case, $X = Y$ and is a split of H. In the second one, $X = Y \cap V_H = W(g', g) \cap V_H$ is also a split of H.

Case 3 : $g' = e, g \neq f$. We must have $g \in E_H$ otherwise $X = V_H$, and then, X is either $W(f, g)$ or $W(g, e)$. In both cases, it is a split of H.

We cannot have $\{g, g'\} = \{e, f\}$ otherwise X is V_H or the empty set. All cases have been considered up to exchanging g and g and/or H and K. $\square$

Corollary 9 : Let G be strongly connected.

(1) If $G = H \boxplus_{e,f} K$, then $\mathcal{A}(G) = \mathcal{A}(H) \cup \mathcal{A}(K)$.

(2) Let G be defined by a term t, and let H occur in t as a constant. Then H is an atom if and only if V_H is atomic in G.

(3) If G is defined by a term t over atoms, then its atomic partition consists of the vertex sets of the atoms occurring in t.

Proof: (1) An atomic set of vertices of G must be included in V_H or in V_K. By Lemma 8, a subset of V_H is atomic in H if and only if it is atomic in G, and this gives the desired equality. (Hence, we have the equality $\sim_H = \sim_G \cap (V_H \times V_H)$).

(2) By induction on the structure of t with help of (1).

(3) Immediate consequence of (2).$\square$

The vertex sets of the atoms used in every two equivalent terms over atoms that define a graph G are the same as they form the atomic partition of G.

Algorithm 10 : *Constructing the atomic skeleton from a term over atoms.*

Given a term t over atoms that defines a graph G in $\mathcal{SC}$, we want to obtain $Sk(G)$. This construction can be done by induction on the structure of t. There is nothing to do if the term is a constant denoting an atom.

For the inductive step, we need only construct the atomic skeleton of $G = H \boxplus_{e,f} K$ from those of H and K. Hence, given $Sk(H) = (\mathcal{A}(H), E_H^{cut}, vert_{Sk(H)})$ and $Sk(K) = (\mathcal{A}(K), E_K^{cut}, vert_{Sk(K)})$ we will construct $Sk(G) = (\mathcal{A}(G), E_G^{cut}, vert_{Sk(G)})$.

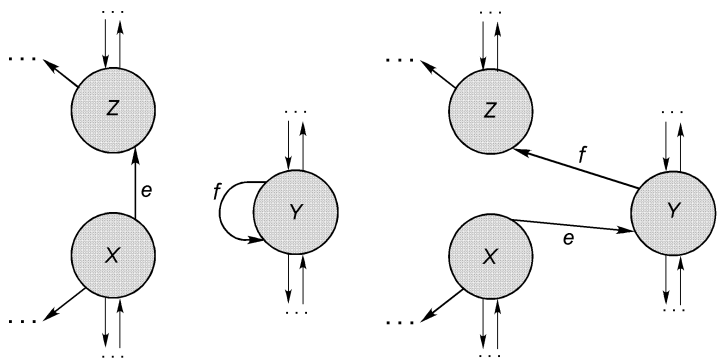

Figure 10: Case 2 of the proof of Algorithm 10

We know by Corollary 9 that $\mathcal{A}(G) = \mathcal{A}(H) \cup \mathcal{A}(K)$. For defining E_G^{cut} and $vert_{Sk(G)}$, we distinguish three cases.

Case 1: $e \notin E_H^{cut}, f \notin E_K^{cut}$.
Then e has its ends in some $X \in \mathcal{A}(H)$ and f has its ends in some $Y \in \mathcal{A}(K)$. So we have $E_G^{cut} = E_H^{cut} \cup E_K^{cut} \cup \{e, f\}$ and $vert_{Sk(G)}$ is defined by:

$$vert_{Sk(G)}(g) = vert_{Sk(H)}(g) \text{ if } g \in E_H^{cut},$$
$$vert_{Sk(G)}(g) = vert_{Sk(K)}(g) \text{ if } g \in E_K^{cut},$$
$$vert_{Sk(G)}(e) = (X, Y),$$
$$vert_{Sk(G)}(f) = (Y, X).$$

Case 2: $e \in E_H^{cut}, f \notin E_K^{cut}$.
We let Y be as above and e link X to Z such that $X, Z \in \mathcal{A}(H)$. Then we have $E_G^{cut} = E_H^{cut} \cup E_K^{cut} \cup \{f\}$ and $vert_{Sk(G)}$ is defined by:

$$vert_{Sk(G)}(g) = vert_{Sk(H)}(g) \text{ if } g \in E_H^{cut} - \{e\}$$
$$vert_{Sk(G)}(g) = vert_{Sk(K)}(g) \text{ if } g \in E_K^{cut},$$
$$vert_{Sk(G)}(e) = (X, Y),$$
$$vert_{Sk(G)}(f) = (Y, Z).$$

This case is illustrated in Figure 10. (The left part shows fragments of $Sk(H)$ and $Sk(K)$. The right part shows how they are combined to yield $Sk(G)$). The case where $e \notin E_H^{cut}, f \in E_K^{cut}$ is similar.

Case 3: $e \in E_H^{cut}, f \in E_K^{cut}$.
We let e link X to Z as in the previous case and e link Y to U such that $Y, U \in \mathcal{A}(K)$. Then we have $E_G^{cut} = E_H^{cut} \cup E_K^{cut}$ and $vert_{Sk(G)}$ is defined by:

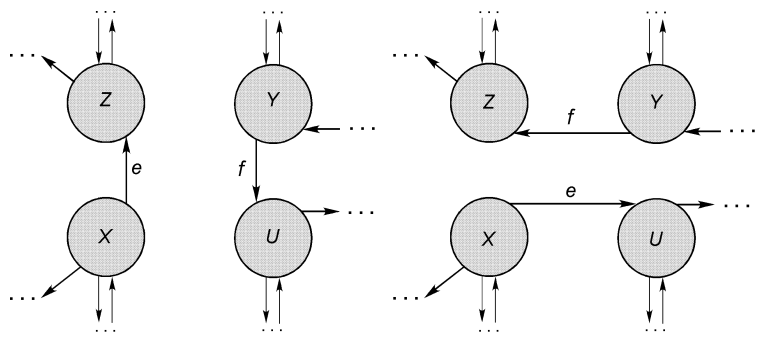

Figure 11: Case 3 of Algorithm 10.

$$vert_{Sk(G)}(y) = vert_{Sk(H)}(y) \text{ if } \ y \in E_H^{cut} - \{e\},$$
$$vert_{Sk(G)}(y) = vert_{Sk(K)}(y) \text{ if } \ y \in E_K^{cut} - \{f\},$$
$$vert_{Sk(G)}(e) = (X, U),$$
$$vert_{Sk(G)}(f) = (Y, Z).$$

This case is illustrated in Figure 11.

The construction of $Sk(H \boxplus_{e,f} K)$ depends only on $Sk(H), Sk(K)$ and the knowledge of the atomic sets of H and K where are the ends of e and f. Hence, we can write:

$$Sk(H \boxplus_{e,f} K) = Sk(H) \boxplus_{e,f} Sk(K)$$

where $Sk(H)$ and $Sk(K)$ are the concrete graphs defined above and not the corresponding abstract graphs (their isomorphism classes). We assume that e and f are given with their ends. The correctness of the construction is clear from Corollary 9(1) and the definitions. $\square$

Note that the construction of $Sk(H \boxplus_{e,f} K)$ is defined for H, K in $\mathcal{SC}$, not in $2\mathcal{EC}$.

The knowledge of $Sk(G)$ brings some structural information about G but does not allow to reconstruct it. We will equip $Sk(G)$ with additional information, and define the *atomic decomposition* of G, from which we will be able to build terms over atoms that define G.

Definition 11: *Equivalent edges.*
Let $G \in \mathcal{SC} \cup 2\mathcal{EC}$. Two edges are *equivalent* if they belong to the same closed walks (closed and directed if $G \in \mathcal{SC}$). This relation is an equivalence denoted

by $\approx_G$. By a *class of edges* of G, we mean an equivalence class of $\approx_G$. A class is *nontrivial* if it is not singleton. Each loop forms a singleton class of edges.

Proposition 12: Let $G \in \mathcal{SC} \cup 2\mathcal{EC}$. Two distinct edges are equivalent if and only if they form a 2-cut if and only if they belong to the same 2-connected component of the cactus $Sk(G)$.

Proof: Let $G \in \mathcal{SC}$ and e and f be distinct edges. If any of them is a loop, then they do not form a 2-cut, they are not equivalent and each of them is a 2-connected component of $Sk(G)$. We now exclude this case.

If they form a 2-cut, every closed walk going through e must go through f, hence $e \approx_G f$. Conversely, assume that e and f are equivalent. The sets $W(e, f)$ and $W(f, e)$ are disjoint, otherwise one can build a closed walk going through e that avoids f. Hence, e and f form a 2-cut (with split $W(e, f)$).

If e and f belong to a 2-connected component of $Sk(G)$, then they form a 2-cut. Conversely, let e and f form a 2-cut. There is a path in G from $\beta(e)$ to $\alpha(e)$ that goes through f and yields a path in $Sk(G)$ (cf. the proof of Theorem 7) from the atomic set containing $\beta(e)$ to the one containg $\alpha(e)$, hence e and f belong to a 2-connected component of $Sk(G)$.

The same equivalences hold for $Und(G)$ in $2\mathcal{EC}$. $\square$

The set E_G^{cut} of 2-cut edges of G is thus the union of the nontrivial classes of $\approx_G$.

Lemma 13: Let $G \in \mathcal{SC}$ be of the form $H \boxplus_{e,f} K$ and $g, g' \in E_G$. We have $g \approx_G g'$ if and only if:

$g \approx_H g'$ or $g \approx_K g'$ or

$g \approx_H e$ and $g' \approx_K f$ or vice versa (by exchanging g and g').

Proof: Assume that $g, g' \in E_H$ and $g \approx_H g'$. Consider a closed walk W in G containing g. If it does not contain e, then it is a closed walk in H, hence it must contain g'. If W contains e, it contains a subwalk of the form (e, A, f) with $A \subseteq E_K$. If one deletes (A, f) from W, one gets a closed walk W' of H (recall that the head of e in H is that of f in G). Hence it must contain g', and so does W. Hence, $g \approx_G g'$. All other parts of the proof use similar arguments.$\square$

Definition 14 : *The atoms of a strongly connected graph*

Let $G \in \mathcal{SC}$ and $X \in \mathcal{A}(G)$. If $X = V_G$ then G is an atom and is its unique atom. Otherwise, we make $G[X]$ into a graph $G\{X\} \in \mathcal{SC}$ by adding new edges. Let F be a nontrivial class containing an edge with an end in X. There exists one because otherwise G is not connected. By Proposition 12, F has exactly two edges, e and f, such that $\alpha(e) \notin X$, $\beta(e) \in X$, $\alpha(f) \in X$ and $\beta(f) \notin X$. We make f into a new edge of $G\{X\}$ by changing its head into $\beta(e)$. We call f the *F-handle of* $G\{X\}$. We construct $G\{X\}$ by adding to $G[X]$ the F-handles

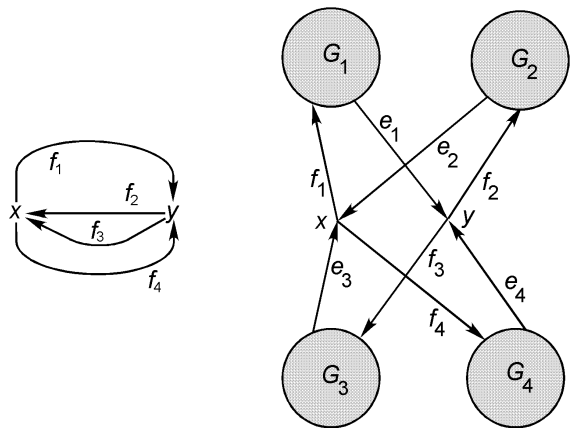

Figure 12: An atom H (to the left) of a graph G.

for all such nontrivial classes F. It is clear that $G\{X\}$ belongs to $\mathcal{SC}$ and is an atom. We say that it is *an atom of* G. It is (2,2)-regular if G is.

We will denote by $\mathcal{A}tom(G)$ the set of atoms of G.

For an example consider Figure 7. Let us assume that the vertex sets $V_{G_1}, ...,$ V_{G_4} of the graph G are atomic. The corresponding atoms are shown to the left. Another example is in Figure 12 which shows a graph $G \in \mathcal{SC}$ with atomic set $X = \{x, y\}$ such that $G[X]$ has no edge, hence is not in $\mathcal{SC}$. The corresponding atom $H = G\{X\} \in \mathcal{SC}$ is the 4-bond shown to the left.

A vertex with at least one loop (resp. exactly two loops) is an atom of $\mathcal{SC}$ (resp. of $\mathcal{G}_{2,2}$). As other atoms of $\mathcal{SC}$, we have the k-bonds with one pair of opposite edges and $k \geq 3$. The atoms of $\mathcal{G}_{2,2}$ that are bonds are the 4-bonds with two pairs of opposite edges (see the left part of Figure 12).

Proposition 15 : Let $G \in \mathcal{SC}$.

(1) If H is an atom of G and $Y_1, ..., Y_k$ are the vertex sets of the connected components of $G - V_H$, then, each set Y_i is a split and we have:

$$G = (..(H \boxplus_{f_1,e_1} G\{Y_1\}) \boxplus_{f_2,e_2} G\{Y_2\})...) \boxplus_{f_k,e_k} G\{Y_k\}$$

where $f_1, ..., f_k$ are the handles of H and each e_i is the unique handle of $G\{Y_i\}$.

(2) Conversely, if

$$G = (..(H \boxplus_{f_1,e_1} G_1) \boxplus_{f_2,e_2} G_2)...) \boxplus_{f_k,e_k} G_k,$$

if $f_1, ..., f_k$ are edges of H and V_H is atomic, then H is an atom with handles $f_1, ..., f_k$.

(3) If $G = H \boxplus_{e,f} K$ then, the atoms of H are those of G having a vertex set included in V_H.

Proof: (1) This is a consequence of Theorem 7. The set V_H is atomic and each set Y_i is a union of atomic sets. The edges of $Sk(G)$ incident to V_H form a set $\{e_1, ..., e_k, f_1, ..., f_k\}$ such that for each i, f_i has its tail in V_H, e_i has its head in V_H, e_i is equivalent to f_i and is not equivalent to e_j if $j \neq i$. They can be numbered in such a way that e_i and f_i have their other ends in Y_i. The equality follows then from the definitions, and clearly, $Y_1, ..., Y_k$ are splits (V_H is the intersection of their complements).

(2) Let $G = (..(H \boxplus_{f_1,e_1} G_1) \boxplus_{f_2,e_2} G_2)...) \boxplus_{f_k,e_k} G_k$ where $f_1, ..., f_k$ are edges of H and V_H be atomic. Each graph G_i is connected because G is connected and for each i, e_i and f_i belong to a same closed walk. Let $H' = G\{V_H\}$. We have an expression of G following from (1) and it is easy to see that $H = H'$. Hence, H is an atom of G.

(3) A consequence of Corollary 8(1) and the previous properties.□

Note that, in Assertion (1), the vertex sets $Y_1, ..., Y_k$ correspond to the nontrivial classes having edges with an end in V_H.

3.3 The atomic decomposition of a strongly connected graph.

There are many terms over atoms that define a given graph $G \in \mathcal{SC}$, and all these terms use the same set of atoms. We now define a *canonical* structure, called the *atomic decomposition* of G that expands $Sk(G)$ by indicating the atom that corresponds to each vertex of $Sk(G)$. The graph G can be reconstructed from it. Terms over atoms that define G can also be constructed from this structure.

Definition 16: *Atomic decomposition.*

(a) An *atomic decomposition* is a 4-tuple $\mathcal{D} = (V, E, vert, atom)$ such that:

(i) the triple $(V, E, vert)$ is a directed cactus C,

(ii) for each $u \in V$, $atom(u)$ is an atom in $\mathcal{SC}$ that is disjoint from $atom(v)$ for every $v \neq u$ and is such that $V \cap V_{atom(u)} = \emptyset$.

(iii) each edge $e \in E$ that links u to v in C is (also) an edge of $atom(u)$ (with different tail and head).

The last condition implies that, if $vert(e) = (u, v)$, then $vert_{atom(u)}(e)$ is a pair of vertices of $atom(u)$. If $w \neq u$, then $vert_{atom(w)}(e)$ is undefined because e is not an edge of $atom(w)$.

(b) The graph $G = G(\mathcal{D})$ that is *decomposed by* $\mathcal{D}$ is defined as follows:

$V_G = \bigcup\{V_{atom(u)} \mid u \in V\}$,
$E_G = \bigcup\{E_{atom(u)} \mid u \in V\}$,
$vert_G(e) = vert_{atom(u)}(e)$ if $e \in E_{atom(u)} - E$,
$vert_G(e) = (x, y)$ if $e \in E \cap E_{atom(u)}$, $vert(e) = (u, v)$, $vert_{atom(u)}(e) = (x, x')$, $vert_{atom(v)}(f) = (y', y)$ where f is the edge of C with tail v that follows e on the circuit that is the 2-connected component of C containing e.

Clearly, $G \in \mathcal{SC}$. For an example, consider the graph G of Figure 7 and assume that $G_1, ..., G_4$ are atoms. It has an atomic decomposition whose cactus is the circuit $(e_1, ..., e_4)$. For another example, consider the graph G of Figure 12 where, again, $G_1, ..., G_4$ are atoms. It has an atomic decomposition whose cactus has vertices $z, g_1, ..., g_4$ and edges $e_i : g_i \to z$ and $f_i : z \to g_i$ for $i = 1, ..., 4$. Furthermore, $atom(z)$ is the graph H shown at the left of Figure 12 and $atom(g_i) = G_i$ for $i = 1, ..., 4$.

It is clear that each set $V_{atom(u)}$ is atomic in G. It follows that $C = (V, E, vert)$ is isomorphic to the atomic skeleton of G (by the mapping h such that $h(u) = V_{atom(u)}$ and $h(e) = e$). Furthermore, the graphs $atom(u)$ are the atoms of G.

(c) Two atomic decompositions $\mathcal{D} = (V, E, vert, atom)$ and $\mathcal{D}' = (V', E', vert', atom')$ (with corresponding cactus C') are *isomorphic*, denoted by $\mathcal{D} \cong \mathcal{D}'$, if there is a mapping $h : V \cup V_{G(\mathcal{D})} \cup E_{G(\mathcal{D})} \to V' \cup V_{G(\mathcal{D}')} \cup E_{G(\mathcal{D}')}$ such that:

its restriction to $V \cup E$ is a graph isomorphism : $C \to C'$,

its restriction to $V_{atom(u)} \cup E_{atom(u)}$ is an isomorphism $atom(u) \to atom'(h(u))$ for each $u \in V$.

Note that for every $e \in E$, we have $vert'(h(e)) = (h(u), h(v))$ if $vert(e) = (u, v)$ and also, $vert'_{atom'(u)}(h(e)) = (h(x), h(y))$ if x, y are such that $vert_{atom(u)}(e) = (x, y)$. It is clear that h induces also an isomorphism: $G(\mathcal{D}) \to G(\mathcal{D}')$. In the special case where h is the identity on E and on each graph $atom(u)$, then we say that h is a *vertex renaming* of $\mathcal{D}$ and, clearly, $G(\mathcal{D}) = G(\mathcal{D}')$.

Theorem 17 : Every strongly connected graph has a unique atomic decomposition, where unicity is understood up to a vertex renaming.

Proof: Let $G \in \mathcal{SC}$. We construct a decomposition $\mathcal{D} = (V, E, vert, atom)$ from the atomic skeleton of G: we let $V = \mathcal{A}(G)$, $E = E_G^{cut}$ and, for each $e \in E$, we let $vert(e) = (X, Y)$ if $X, Y \in \mathcal{A}(G)$ and e links X to Y in G (necessarily, $X \neq Y$). For $X \in V$, we define $atom(X) = G\{X\}$. Conditions (i) and (ii) clearly hold, hence, $\mathcal{D} = (V, E, vert, atom)$ is an atomic decomposition. It is also clear from Definitions 14 and 16 that $G(\mathcal{D}) = G$.

Now we prove the unicity. Assume that $G = G(\mathcal{D}) = G(\mathcal{D}')$ where $\mathcal{D}$ and $\mathcal{D}'$ are two atomic decompositions. There is a unique bijection $h : V \to V'$ such that $atom(u) = atom'(h(u))$ for each $u \in V$. Theorem 7 shows that $E = E_G^{cut}$, hence $E = E'$. This shows that h is a vertex renaming : $\mathcal{D} \to \mathcal{D}'$. $\square$

All atomic decompositions of a given graph G are isomorphic. However, the atomic decomposition $\mathcal{AD}(G)$ defined as $(\mathcal{A}(G), E_G^{cut}, vert, atom)$ such that $vert(e) = (X, Y)$ if e links X to Y in G and $atom(X) = G\{X\}$ is constructed (in a unique way) from vertices and edges of G (This observation is useful in view of a formalization in logic, see Section 3.9).

Constructing terms from atomic decompositions

Algorithm 10 (in Section 3.2) constructs the atomic skeleton of a strongly connected graph from a term over atoms that defines it. From the atomic skeleton one gets the atomic decomposition $\mathcal{AD}(G)$. We now consider the converse construction.

Algorithm 18 : *Constructing a term from the atomic decomposition.*

Let be given an atomic decomposition $\mathcal{D}$ and let $G = G(\mathcal{D})$. We define a term t_G that defines G by using an induction on the size of the cactus C of $\mathcal{D}$.

If C has a unique vertex, then G is an atom and t_G is the term reduced to this atom as a constant.

Otherwise, we let $(u_1, e_1, u_2, e_2, ..., u_k, e_k, u_1)$ be the circuit of some (arbitrarily chosen) 2-connected component of C. (Here we list also the vertices $u_1, u_2, ..., u_k$ of the circuit). Let $X_i = V_{atom(u_i)}$ and Y_i be the split of G defined by the 2-cut $\{e_{i-1}, e_i\}$ that contains X_i (the 2-cut $\{e_1, e_k\}$ if $i = 1$). We let also $G_i = G\{Y_i\}$. Clearly, $G = \boxplus_{e_1,...,e_k}(G\{Y_1\}, ..., G\{Y_k\})$.

The connected component of $C - \{e_{i-1}, e_i\}$ that contains u_i is a cactus, denoted by C_i, and we let $\mathcal{D}_i$ be the restriction of $\mathcal{D}$ to it. It is an atomic decomposition of $G\{Y_i\}$. The induction gives a term t_{G_i} using the atoms of $G\{Y_i\}$ (they are the atoms of G whose vertex set is included in $G\{Y_i\}$). Hence, we can take $t_G = \boxplus_{e_1,...,e_k}(t_{G\{Y_1\}}, ..., t_{G\{Y_k\}})$.

This construction takes linear time in term of the size of $\mathcal{D} = (V, E, vert, atom)$ that we define in a natural way as $|V| + |E| + \sum_{u \in V} \left|V_{atom(u)}\right| + \left|E_{atom(u)}\right|$; this size is proportional to $\left|E_{G(\mathcal{D})}\right|$.$\square$

3.4 The atomic skeleton of a 2-edge connected undirected graph

We now extend to undirected graphs the previous constructions. We recall that a graph H is $Und(G)$ for some strongly connected graph G (*i.e.*, for G in $\mathcal{SC}$) if and only if it is 2-edge connected (*i.e.*, is in $2\mathcal{EC}$), if and only if it is connected and

has no bridge. Such a graph H has several strongly connected orientations but, fortunately, most of the notions defined in the previous sections are invariant under a change of strongly connected orientation.

We have observed that if $G, G' \in \mathcal{SC}$ and $Und(G) = Und(G')$, then $\sim_G = \sim_{G'}, \approx_G = \approx_{G'}$ and $\mathcal{A}(G) = \mathcal{A}(G')$, so that G and G' have the same atomic skeleton. However, the atoms need not be the same in the atomic decompositions of G and G'. We may have $Und(G) = Und(G')$ with $G = H \boxplus_{e,f} K$, $G' = H' \boxplus_{f,e} K'$, for atoms H, K, H' and K' such that $V_H = V_{H'}$, with e directed from H to K in G and from K' to H' in G'. In this case, V_H is an atomic set of G and G' but the corresponding atoms are respectively H and H', and they are not the same since e is an edge of H and not of H'. However, $Und(H)$ and $Und(H')$ are v-isomorphic (cf. Section 2.1).

Definition 19 : *The atoms of a graph in* $2\mathcal{EC}$

We define an *atom* of $H = Und(G)$ as $Und(K)$ where K is an atom of $G \in \mathcal{SC}$. This definition depends on G that is not unique. However, if G' belonging to $\mathcal{SC}$ is another orientation of H, then the atoms of H defined from G are v-isomorphic to those defined from G'; (a v-isomorphism is the identity on vertices, cf. Section 2.1). For checking that, we recall from Definition 14 that an atom K is obtained from $G[X]$ where X is an atomic set of vertices by adding new edges; for each nontrivial class of edges F having an edge f with tail in X, we add f to $G[X]$ as an edge of K and we change its head into the head of the unique edge e in F (hence, equivalent to f), with head in X. Let G' be as above and F be a nontrivial class of edges of G. Either the edges of F have all the same orientation in G' as in G, or they have all the opposite orientation. In the latter case, e instead of f is added to $G[X]$ to build an atom K' of G', but its two ends are the same as those of f in K. It follows that $Und(K)$ and $Und(K')$ are v-isomorphic.

We define $\mathcal{A}tom_{iso}(H)$ where that $H = Und(G)$, as the set of v-isomorphism classes of the graphs $Und(K)$ for K in $\mathcal{A}tom(G)$. The subscript *iso* is to recall that the atoms of H are defined up to v-isomorphisms.

Examples of atoms in $2\mathcal{EC}$ are the cliques K_n for $n \geq 4$, the (undirected) k-bonds for $k \geq 3$, the wheels with at least 4 vertices (see [Die]; K_4 is the one with 4 vertices), the single vertices with at least one loop.

What could or should be an atomic decomposition of a 2-edge connected undirected graph ?

The first possibility is to take the atomic decomposition of any strongly connected orientation of the considered graph. It is not unique, but its skeleton is, as we have seen.

Another possibility is to define, for every atomic decomposition $\mathcal{D} = (V, E, vert, atom)$ of a graph G in $\mathcal{SC}$, the tuple $Und(\mathcal{D})$ as $(V, E, \underline{vert}, \underline{atom})$ where,

for $e \in E$ and $u \in V$, $\underline{vert}(e) = \{x, y\}$ if $vert(e) = (x, y)$ and $\underline{atom}(u) = Und(atom(u))$ (hence is isomorphic to an atom of the graph $Und(G)$). We could call it an atomic decomposition of $Und(G)$. With this definition, two atomic decompositions $\mathcal{D} = (V, E, \underline{vert}, \underline{atom})$ and $\mathcal{D}' = (V', E', \underline{vert}', \underline{atom}')$ of a graph H in $2\mathcal{EC}$ are *isomorphic* in the following sense:

> there is a graph isomorphism $h : (V, E, \underline{vert}) \to (V', E', \underline{vert}')$, and
>
> for each $u \in V$, there is a graph isomorphism $h_u : \underline{atom}(u) \to \underline{atom}'(h(u))$.

Consider for an example the graph $G \in \mathcal{SC}$ of Figure 7 defined by the term $\boxplus_{e_1,...,e_4}(G_1, ..., G_4)$ where $G_1, ..., G_4$ are atoms. Let $G' \in \mathcal{SC}$ be obtained from G by reversing the directions of $e_1, ..., e_4$ and of some other edges of $G_1, ..., G_4$. We obtain two "atomic decompositions" for $Und(G)$. The directed cactus of G is the circuit $(u_1, e_1, u_2, e_2, u_3, e_3, u_4, e_4, u_1)$ and that of G' is $(u_1, e_4, u_4, e_3, u_3, e_2, u_2, e_1, u_1)$. The corresponding (undirected) cycles are thus equal.

However, this definition is not satisfactory because we cannot reconstruct $Und(G)$ in a unique way from $Und(\mathcal{D})$. Consider $G = H \boxplus_{e,f} K$, where H and K are atoms, $e : u \to u'$, $f : v' \to v$. The cactus of its (canonical) atomic decomposition $\mathcal{D}$ is the 2-bond consisting of $e : V_H \to V_K$ and $f : V_K \to V_H$. From $Und(\mathcal{D})$, we know the corresponding undirected cactus and we know that e links u and u' in $Und(H)$ and that f links v and v' in $Und(K)$, but nothing indicates that, in $Und(G)$, e links u and u' rather than u and v'. (We observed that already at the end of Definition 1). Hence, for graphs in $2\mathcal{EC}$, we will only use their atomic skeletons and the atomic decompositions of their strongly connected orientations.

3.5 From Tutte decompositions to atomic skeletons

In this section, we show that the atomic skeleton of a graph in $\mathcal{SC} \cup 2\mathcal{EC}$ can be obtained from its Tutte decomposition. This construction will yield a linear-time algorithm and help us to understand the structure of atoms. Relevant notation and definitions are reviewed in Section 2.1. We recall that the Tutte decomposition of a graph does not depend on edge directions. It is defined for graphs with at least three vertices. (We need not decompose small graphs).

We recall that if G is loop-free and 2-connected, if (T, f) is its Tutte decomposition and $B(x)$ is a cycle-block of the form $(e_1, ..., e_k)$ (with $k \geq 3$), then G is the circular gluing $\bigodot_{u_1,...,u_k}(H_1, ..., H_k)$ where each graph H_i is (as usual with circular sequences, we let u_{k+1} be another notation for u_1) :

(a) either the edge e_i of G linking u_i and u_{i+1},

(b) or a bond linking u_i and u_{i+1},

(c) or, if (a) and (b) do not hold, then $H_i = G[V(x;y)]$ where y is the unique neighbour of x such that $f(y) = \{u_i, u_{i+1}\}$.

In Cases (a) and (b), we have $H_i = G[\{u_i, u_{i+1}\}]$.

Definition 20 : *Potential 2-cut edges and splittable cycle-blocks.*
An edge e_i as in Case (a) is a *potential 2-cut edge.* A cycle-block is *splittable* if it has at least two potential 2-cut edges. In Case (b), we say that H_i is a *bond of* the block $B(x)$ (as well as of G). In Cases (b) and (c), u_i and u_{i+1} have degree at least 2 in H_i. Furthermore, if G is strongly connected, then all potential 2-cut edges are directed all from u_i to u_{i+1} or all from u_{i+1} to u_i.

It is clear that V_{H_i} and $V_{H_{i'}}$, such that $1 \leq i < i' \leq k$ are separated in G (which means that there is a split X such that $V_{H_i} \subseteq X$ and $V_{H_{i'}} \subseteq V_G - X$) if and only if there are j and j' such that e_j and $e_{j'}$ are potential 2-cut edges, $i \leq j \leq i' - 1$ and, either $1 \leq j' \leq i - 1$ or $i' \leq j' \leq k$.

Lemma 21 : Let $G \in \mathcal{SC} \cup 2\mathcal{EC}$ be loop-free and 2-connected with Tutte decomposition (T, f).

(1) Two edges form a 2-cut if and only if they are potential 2-cut edges in a same cycle-block.

(2) If this is the case, their common equivalence class is the set of potential 2-cut edges of this cycle-block.

Proof: (1) We first consider $G \in \mathcal{SC}$.

"Only if". Let e be a 2-cut edge of G. This means that G has a split X such that e links X to $V_G - X$ and that some other edge e' links $V_G - X$ to X. The edges e and e' are not parallel edges because, otherwise, G would not be 2-connected (it has at least three vertices). Hence, G can be expressed as a circular gluing of e,e' and one or two connected subgraphs. It follows that the Tutte decomposition of G has a cycle-block that contains e and e'. Hence e and e' are potential 2-cut edges in a same cycle-block.

"If". Conversely, let e be a potential 2-cut edge in a cycle-block B. Then B can be ordered as a sequence of consecutive edges $e_1, e_2, ..., e_k$ with $e = e_1$ and $k \geq 3$. By the initial remark, we have $G = \bigodot_{u_1,...,u_k}(H_1, ..., H_k)$ where H_1 is e_1. If H_i (for $i > 1$) is an edge, hence, is another potential 2-cut edge, then $\{e_1, e_i\}$ is a 2-cut of G. This completes the proof if $G \in \mathcal{SC}$. If $G \in 2\mathcal{EC}$, the proof is the same except that the edges e, e', $e_1, e_2, ..., e_k$ are undirected.

(2) is an immediate consequence of (1). $\square$

Figure 5 in Section 2 shows the Tutte decomposition of a 4-regular undirected graph that is shown in Figure 4. We can see that the 2-cut edges of G are $a - c$ and $b - d$ in cycle-block B, $a - e$ and $b - f$ in C, $b - l$ and $k - m$ in F, as stated by Lemma 21.

3.5.1 An algorithm for the atomic decomposition.

In this section, we describe a linear-time algorithm that constructs the atomic decomposition of a strongly connected graph. It yields a linear-time algorithm that constructs the atomic skeleton of a 2-edge connected undirected graph, because one can construct in linear time, by using a depth-first traversal, a strong orientation of it.

The case of a 2-connected loop-free graph.

We first consider the case of $G \in \mathcal{SC}$ that is loop-free and 2-connected. The first step consists in constructing its Tutte decomposition (T, f). This can be done in linear time by using the algorithm of [HopTra2]. By Lemma 21, the nontrivial classes of edges are obtained from the splittable cycle-blocks, which are easy to recognize. We now determine the atomic sets of vertices.

Definition 22 : Let $G \in \mathcal{SC}$ be loop-free and 2-connected with Tutte decomposition (T, f). Two sets of vertices U and V are *separated at a cycle-block* B if they are separated by a split relative to two potential 2-cut edges of B.

Note that two vertices in a box $f(x)$ for $x \in N_{2T}$, or in a 3-connected block or in a cycle-block that is not splittable, are not separated. Here is a complementary statement.

Lemma 23 : Let P be a path in T that links two nodes x and y in N_{3T}. (We may have $x = y$ and then P is reduced to x). Let $u, v \in V_G$ such that $u \in f(x)$ and $v \in f(y)$.

(1) If u and v are separated at a cycle-block, then this cycle-block is on P (i.e., it is $B(z)$ for some node z of P).

(2) Let z be a node of P such that $B(z)$ is a cycle-block whose corresponding expression of G is $\bigodot_{u_1,...,u_k}(H_1, ..., H_k)$. Then, u and v are separated at $B(z)$ if and only if, for some i, j (as usual, $u_{k+1} = u_1$) :

(2.1) $u = u_i$, $v = u_j$ and u and v are separated at $B(z)$, or

(2.2) $u \in V_{H_i} - \{u_i, u_{i+1}\}$, $v = u_j$ and $\{u_i, u_{i+1}\}$ and u_j are separated at $B(z)$, or

(2.3) the similar condition holds where u and v are exchanged, or

(2.4) $u \in V_{H_i} - \{u_i, u_{i+1}\}$, $v \in V_{H_j} - \{u_j, u_{j+1}\}$ and $\{u_i, u_{i+1}\}$ and $\{u_j, u_{j+1}\}$ are separated at $B(z)$.

The verification of Conditions (2.2), (2.3) and (2.4) uses the following observation:

> if $u \notin \{u_1, ..., u_k\}$, then $u \in V_{H_i} - \{u_i, u_{i+1}\}$ where $\{u_i, u_{i+1}\}$ is $f(x')$ and x' is the node of N_{2T} on the subpath of P between x and z that is adjacent to z.

Proof (1) Let u and v are separated at a cycle-block $B(z)$ whose corresponding expression of G is $\bigodot_{u_1,...,u_k}(H_1,...,H_k)$. Assume that z is not on P. Let t be the node of N_{2T} that is adjacent to z on the path in T between x and z and on the one between y and z. Then $f(t) = \{u_i, u_{i+1}\}$ for some i and thus, u and v belong to V_{H_i}. Hence they are not separated at $B(z)$. It follows that z must be on P.

(2) The "if" direction is clear. For proving the converse, consider u, v and z as in the statement, where u and v are separated by a split X relative to two potential 2-cut edges of $B(z)$. Then, in each of the four cases, it is trivial or easy to check that the same split separates u_i and u_j, or $\{u_i, u_{i+1}\}$ and u_j, or $\{u_i, u_{i+1}\}$ and $\{u_j, u_{j+1}\}$. $\square$

For an example, look at the Tutte decomposition of Figure 5. The cycle-blocks are B, C, F and G. The splittable ones are B, C and F. The atomic sets are $\{a, b, i, j, k\}$, $\{c, d\}$, $\{e, f, g, h\}$ and $\{l, m, n\}$.

Proposition 24 : Let $G \in \mathcal{SC}$ be loop-free and 2-connected with Tutte decomposition (T, f).

(1) Let $u \in f(x)$, for some $x \in N_{3T}$. The atomic set of vertices that contains u is the set of vertices v belonging to $f(y)$ for some $y \in N_{3T}$ such that u and v are not separated at any cycle-block on the path in T between x and y.

(2) The atomic partition of G can be computed from G and (T, f) in linear time.

Proof: (1) By Lemma 21(1), two vertices are separated in G if and only if they are so at some cycle-block. And this cycle-block, if it exists, must be on the path between x and y by Lemma 23(1). This proves the first assertion.

(2) The algorithm is based on a traversal of T that defines an increasing sequence of subtrees : $T_1 \subseteq T_2 \subseteq ... \subseteq T_p = T$ where T_1 is just one node in N_{3T} and, for each i, T_{i+1} is T_i with one more node x_{i+1} linked to some node of T_i. For each i, we let $V_i \subseteq V_G$ be the union of boxes $f(x)$ such that x is a node of T_i. It follows from Lemma 23 that two vertices u and v in V_i are separated in G if and only if there is in T_i a path with a cycle-block witnessing that. Hence the partition $\mathcal{A}_i$ of V_i consisting of the nonempty sets of the form $V_i \cap X$ for $X \in \mathcal{A}(G)$ can be determined from T_i. This can be done in the following inductive way.

Initial step: $\mathcal{A}_1 = \{V_1\}$ if the block $f(x_1)$ is 3-connected or not splittable. If this block is splittable, then $\mathcal{A}_1$ is the corresponding partition of V_1, according to Lemma 23(2).

Inductive step: We construct $\mathcal{A}_{i+1}$ from $\mathcal{A}_i$ as follows.

If $x_{i+1} \in N_{2T}$, then $\mathcal{A}_{i+1} = \mathcal{A}_i$ because the two vertices of $f(x_{i+1})$ are already in V_i (and they belong to a same class of $\mathcal{A}_i$ because they are not separated in G).

If $x_{i+1} \in N_{3T}$ and $B(x_{i+1})$ is 3-connected or not splittable, then, we add its vertices not already in V_i to the class of $\mathcal{A}_i$ that includes the set $f(y)$ where y is the node of T_i (necessarily in N_{2T}) adjacent to x_{i+1}.

If $x_{i+1} \in N_{3T}$ and $B(x_{i+1})$ is splittable, then, by using Lemma 23(2), we build $\mathcal{A}_{i+1}$ by adding new classes to $\mathcal{A}_i$, and if necessary, we add some vertices of $B(x_{i+1})$ to the class of $\mathcal{A}_i$ that includes the set $f(y)$ where y is as in the previous case.$\square$

With the same hypotheses:

Corollary 25 : One can construct in linear time the Tutte decompositions of all atoms of G that have at least three vertices.

Proof: First we observe that the atoms of a 2-connected graph are 2-connected (because if u is a separating vertex of a graph H, it is also one of $H \boxplus_{e,f} K$), hence, they also have Tutte decompositions.

Let $W \in \mathcal{A}(G)$ and (T, f_W) be the tree-decomposition of $G[W]$ such that $f_W(x) = f(x) \cap W$ for every $x \in N_T$. If W has one or two vertices, there is nothing to do.

Otherwise, Definitions 14 and 16 imply that each edge in $E_{G\{W\}} - E_{G[W]}$ has its two ends in some box of (T, f). Hence, (T, f_W) is a tree-decomposition of $G\{W\}$. It may not be its Tutte decomposition because some boxes may be empty. We remove the corresponding nodes from T, and we obtain a tree T'. But T' with the corresponding restriction of f_W is not always a Tutte decomposition because a splittable block $B(x)$ may become reduced to two vertices. (Such two vertices constitute the box $f(y)$ for some $y \in N_{2T}$ adjacent to x). In this case, we can delete x and y because $f(y) \subseteq f(z)$ where z is any other neighbour of y. By performing this deletion step as many times as necessary, one reaches a tree-decomposition that satisfies conditions T1-T4 but perhaps not T5. As observed in Section 2.1, it can be transformed into a Tutte decomposition.

We can of course repeat this construction for each set $W \in \mathcal{A}(G)$ (we know how to get them by Proposition 24(1)). However, we can do better. Consider again the inductive construction of Proposition 24(2). Together with $\mathcal{A}_i$ we can maintain, for each set W of $\mathcal{A}_i$, the subtree $T_{i,W}$ of T_i consisting of the union of all paths P between nodes x and y in $V_i \cap N_{3T}$ such that $f(x)$ and $f(y)$ contain respectively a vertex u and a vertex v, both in W. (Hence, u and v are not separated on P by a cycle-block). At the end, we obtain a subtree $T_{p,W}$ of T and thus, a tree-decomposition $(T_{p,W}, f_{p,W})$ of $G[W]$ by defining $f_{p,W}(z) = W \cap f(z)$ for every node z of $T_{p,W}$. As in the case of the initial observation, it is actually a tree-decomposition of $G\{W\}$. Its boxes are all non empty, however, we may have splittable blocks reduced to two vertices, and we can use the same transformations to get the desired Tutte decomposition.

The number of vertices (resp. of edges) of a graph is equal to the sum of the corresponding numbers for its atoms. It follows that these computations can be done in linear time in the size of the input graph. The size of the output is also linear in that of the input graph. $\square$

The general case.

We now extend this algorithm to all graphs in $\mathcal{SC}$ by using an induction on the number of separating vertices.

Lemma 26 : Let G in $\mathcal{SC}$ be $H//_uK$ where H and K are not reduced to u, so that u is a separating vertex.

(1) Let $x, y \in V_G$. Then $x \sim_G y$ if and only if either $x \sim_H y$ or $x \sim_K y$ or $x \sim_H u$ and $u \sim_K y$.

(2) Let $e, f \in E_G$. Then $e \approx_G f$ if and only if either $e \approx_H f$ or $e \approx_K f$.

Proof: (1) Let X be a split of G that contains u. Then, $V_G - X$ is a split of H or of K. The proof is as follows: if $V_G - X$ contains vertices of H and of K, then its 2-cut consists of a bridge of H and one of K so that G is not strongly connected. Hence, either $V_G - X \subseteq V_H$ and it is a split of H or $V_G - X \subseteq V_K$ and it is a split of K. Conversely, if X is a split of H that contains u, then, $V_G - X$ is a split of G, and similarly for K. The claimed assertion follows from this fact.

(2) Let us observe that if $e \in E_H$ and W is a closed walk of G including edges of K, then it goes at least twice through u and it has a subwalk that includes e but no edge of K. Hence, if $e, f \in E_G$ and $e \approx_G f$, we cannot have e in H and f in K. The claimed assertion follows easily. $\square$

We define a "gluing" of atomic decompositions. Let $G = H//_uK$ and let A and B be the atoms of H and K respectively that contain u. Then $A//_uB$ is an atom of G. We use Lemma 33 to build the atomic decomposition $\mathcal{D}$ of G from those, $\mathcal{D}' = (V', E', vert', atom')$ and $\mathcal{D}'' = (V'', E'', vert'', atom'')$ of H and K respectively. Without loss of generality, we assume that $V \cap V' = \{w\}$ where w is, in $\mathcal{D}$ and in $\mathcal{D}'$, the vertex corresponding to A and B respectively. Then, we define :

the cactus C of $\mathcal{D}$ as $C'//_wC''$, i.e., as the gluing at w of the cactuses C and C',

the mapping $atom$ by

$atom(w) = A//_uB$,

$atom(x) = atom'(x)$ if $x \in V' - \{w\}$,

$atom(x) = atom''(x)$ if $x \in V'' - \{w\}$.

It is clear that $G(\mathcal{D}) = G(\mathcal{D}')//_uG(\mathcal{D}'') = G$.

The decomposition of a connected graph in 2-connected components can be constructed in linear time ([HopTar1]). The gluing of two atomic decompositions can be done in constant time. Hence, we have proved the following theorem.

Theorem 27 : There exist linear-time algorithms that construct the atomic decomposition of any graph in $\mathcal{SC}$ and the atomic skeleton of any graph in $2\mathcal{EC}$.

Proof: The algorithm for graphs in $\mathcal{SC}$ has been described above. For a graph H in $2\mathcal{EC}$, we construct first a strongly connected orientation G of it by means of a depth-first traversal. This step takes linear time. Then we construct the atomic skeleton of G which gives the one of H.$\square$

Remark : Let us consider the particular case of $G = H//_u K \in \mathcal{G}_{2,2}$. The vertex u cannot have degree 1 in H or K and degree 3 in the other because the sum of the degrees of the vertices of H must be even and all vertices except u have degree 4. Hence, u has degree 2 in H and in K. Moreover, by a similar counting argument we can see that u has indegree and outdegree 1 in H and in K.

We first assume that H and K are not loops. We let $e_1 : x \to u$ and $f_1 : u \to y$ be the edges of H with end u and $e_2 : x' \to u$ and $f_2 : u \to y'$ be the similar edges of K. We have $G = G_1 \boxplus_{e_1,f_1} \mathbf{8}(u, f_1, f_2) \boxplus_{f_2,e_2} G_2$ where $\mathbf{8}(u, f_1, f_2)$ has vertex u and loops f_1 and f_2, $G_1 = G\{V_H - \{u\}\}$ and $G_2 = G\{V_K - \{u\}\}$. The graphs G_1 and G_2 have fewer separating vertices than G. Their atomic decompositions, obtained by way of induction, can be combined to yield the one of G by Algorithm 10.

If H and K are loops, then G is atomic and there is nothing to do. If K is a loop f_2 on u, then we have $G = G_1 \boxplus_{e_1,f_1} \mathbf{8}(u, f_1, f_2)$ and $G_1 = G\{V_H - \{u\}\}$. Hence, we can also use induction.

For this case, the atoms A and B of the general construction are loops on vertex u, and $A//_u B = \mathbf{8}(u, f_1, f_2)$. So, we get the same result.$\square$

3.5.2 The structure of atoms

The graph $Graph(M)$ where M is the first map of Figure 6 is the unique atom in $\mathcal{G}_{2,2}$ with a unique vertex. The graph H of Figure 12 is also an atom in $\mathcal{G}_{2,2}$. We know that a graph in $\mathcal{SC} \cup 2\mathcal{EC}$ is an atom if and only if it has no 2-cut, hence, if and only if it is 3-edge connected. For 4-regular graphs, we have a bit more.

Proposition 28: The loop-free atoms in $\mathcal{G}_4$ (hence also those in $\mathcal{G}_{2,2}$) are 4-edge connected.

Proof: Let u, v be two distinct vertices of an atom $G \in \mathcal{G}_4$. Since G has no split, they are linked by three edge-disjoint paths ([Die], Theorem 3.3.6). Let us remove the edges of two of these paths. Some connected component of the resulting graph contains u and v: its vertices have degree 2 or 4. Hence, it has an Eulerian tour from which we can extract two edge-disjoint paths between u and v by removing some closed subwalks if necessary. Hence, there are four edge-disjoint paths between u and v and G is 4-edge connected. $\square$

By Lemma 23, a graph in $\mathcal{SC} \cup 2\mathcal{EC}$ is an atom if and only its Tutte decomposition has no splittable cycle-block. We now examine what these conditions give for graphs in $\mathcal{G}_{2,2} \cup \mathcal{G}_4$.

Since a graph in $\mathcal{G}_4$ is an atom if and only if it has an orientation in $\mathcal{G}_{2,2}$ that is an atom (and if and only if it all its orientations in $\mathcal{G}_{2,2}$ are atoms), and since Tutte decompositions do not depend on edge directions, we need only consider atoms in $\mathcal{G}_4$.

Let G in $\mathcal{G}_4$ be of the form $\bigodot_{u_1,...,u_k}(H_1,...,H_k)$ where each graph H_i is connected and $k \geq 3$. Then, we have one of the following two cases:

(i) for each $i = 1,...,k$, the vertices u_i and u_{i+1} have degree 2 in H_i, (as usual $u_{k+1} = u_1$),

(ii) k is even, H_i is an edge for each even i, and u_i and u_{i+1} have degree 3 in H_i for each odd i, or similarly, H_i is an edge for each odd i, and u_i and u_{i+1} have degree 3 in H_i for each even i.

This fact follows from the observation that the sum of degrees of a graph is even and that all vertices of $V_{H_i} - \{u_i, u_{i+1}\}$ have degree 4 in H_i. If $G \in \mathcal{G}_{2,2}$, we have more constraints. For example in Case (i), either u_i has outdegree 2 and u_{i+1} has indegree 2 in H_i for all i, or u_i has indegree 2 and u_{i+1} has outdegree 2 in H_i for all i, or u_i and u_{i+1} have indegree and outdegree 1 in H_i for all i.

We will apply this observation to the expressions of graphs as circular gluings associated with cycle-blocks.

A *bond edge* in a 3-connected block B is an edge coming from the fusion of the edges of a bond of the decomposed graph G (cf. Section 2.1 for the definition).

Proposition 29 : The Tutte decomposition (T, f) of a loop-free atom G in $\mathcal{G}_4$ satisfies the following properties:

P1 : Every node $x \in N_{2T}$ has degree 2 and at most one of its neighbours is a cycle-block; if $f(x) = \{u, v\}$, then $u - v$ is a virtual edge.

P2 : Let $B(y)$ be a cycle-block expressed as a sequence of consecutive edges $(e_1, ..., e_k)$. We know that it is not splittable. Each edge e_i is either virtual or comes from the fusion of the edges of a 2-bond of $B(y)$.

P3 : In a 3-connected block $B(z)$, no vertex is comon to two virtual edges, to two bond edges or to a bond edge and a virtual edge. Each vertex of $B(z)$ has degree 4 where we count a virtual edge or a bond edge for two parallel edges.

Condition P3 means that in $B(x)$, a vertex is incident to four edges of G or to two edges of G and a single, virtual or bond edge.

Proof: P1 : Let $x \in N_{2T}$ and $f(x) = \{u, v\}$. The graph G can be expressed as $H_1//_{u,v}H_2//_{u,v}\cdots //_{u,v}H_k$ where $k \geq 2$ is the degree of x. The graphs H_i are connected. It follows that $k \leq 4$. Assume that $k \geq 3$; then u and v have degree at most 2 in all components H_i and u has degree 1 in one of them, say H_j. Hence, v has also degree 1 in H_j (it cannot have degree 3). But then, the two edges of H_j incident with u and v form a 2-cut, which is impossible since

G is an atom. Hence, $k = 2$ and u and v have degree 2 in H_1 and in H_2. There is no edge in G between u and v, because otherwise, H_1 or H_2 would have one edge incident with u and one edge incident with v that would form a 2-cut. Hence, $u - v$ is a virtual edge in the two blocks $B(y)$ and $B(z)$ such that y and z are the neighbours of x in T. By Condition T5 of the definition of a Tutte decomposition, at most one of them is a cycle-block.

P2: Since G is an atom, a cycle-block $B(y)$ is not splittable, hence the corresponding expression of G as $\bigodot_{u_1,...,u_k}(H_1, ..., H_k)$ is of type (i) of the initial observation. This yields the result.

P3: Let $B(z)$ be a 3-connected block. It has no vertex of degree 2, otherwise, it would not be 3-connected. Hence, it has no 3-bond (because an atom has no three parallel edges unless it is the 4-bond). Hence the bonds of $B(z)$ must have two edges. Consider now an edge $e : u - v$ of $B(z)$ such that $f(x) = \{u, v\}$ for some neighbour x of z (hence, $x \in N_{2T}$). By P1, e is a virtual edge and its ends u and v have degree 2 in H_1 and in H_2 (cf. the proof of P1). If u is an end in $B(z)$ of another virtual edge or of a bond edge, then it would have degree 2 in $B(z)$, which we have excluded. Similarly, if $e : u - v$ is a bond edge, then u cannot be the end of another bond edge. The last assertion of P3 is a consequence of these facts. $\square$

We can actually describe the atoms of $\mathcal{G}_4$ in terms of 3-connected graphs. We will denote by $\boldsymbol{U}_2$ (resp. $\boldsymbol{U}_4$) any undirected 2-bond (resp. 4-bond) having the appropriate ends so that an expression like $H//_{u,v}\boldsymbol{U}_2$ is well-defined. In other words, $H//_{u,v}\boldsymbol{U}_2$ is H with two more edges between u and v. (We will not need to designate precisely these edges).

Corollary 30 : A loop-free graph G in $\mathcal{G}_4$ is an atom if and only if it is of the following types:

either it is isomorphic to $\boldsymbol{U}_4$,

or it is 3-connected,

or $G = \bigodot_{u_1,...,u_k}(H_1, ..., H_k)$ where, for each i, $H_i//_{u_i,u_{i+1}}\boldsymbol{U}_2$ is an atom,

or $G = (...((L//_{u_1,v_1}H_1)//_{u_2,v_2}H_2)//_{u_3,v_3}...//_{u_k,v_k}H_k$ where

$u_1, v_1, ..., u_k, v_k$ are pairwise distinct vertices of L,

$(...((L//_{u_1,v_1}\boldsymbol{U}_2)//_{u_2,v_2}\boldsymbol{U}_2)//_{u_3,v_3}...//_{u_k,v_k}\boldsymbol{U}_2$ is a 3-connected atom in $\mathcal{G}_4$ that is not isomorphic to $\boldsymbol{U}_4$,

$H_1//_{u_1,v_1}\boldsymbol{U}_2, ...,$ and $H_k//_{u_k,v_k}\boldsymbol{U}_2$ are atoms in $\mathcal{G}_4$ not isomorphic to $\boldsymbol{U}_4$.

This is a consequence of Proposition 29. Note that $H_1, ..., H_k$ and L are of smaller size than G. Hence, this desciption can be used in an inductive proof for properties of atoms, or similarly, an inductive construction.

This corollary shows that an atom in $\mathcal{G}_4$ is isomorphic to $Und(\mathbf{8})$, to $\boldsymbol{U}_4$ or is 2-connected (which is actually easy to check directly).

3.6 Atomic decompositions of maps and related objects

We have defined the atomic decomposition of every graph in the class $\mathcal{SC}$ (which includes $\mathcal{G}_{2,2}$). We now examine the particular case of (2,2)-regular graphs, of planar graphs and we extend atomic decompositions to maps. This extension is actually a consequence of Definition 1 and Proposition 2, from which we get the following facts, that yield Theorem 31.

Fact 1 : If a graph in $\mathcal{SC}$ is planar, its its atoms are planar and conversely.

Fact 2 : If $G = Graph(M)$ for a map M, then its atoms have a corresponding structure of map. Any term over atoms that defines G can be made in a unique way into a term over atomic maps that defines M. Conversely, if we equip the atoms of the atomic decomposition of a graph $G \in \mathcal{SC}$ (or the graphs occurring as constants in a term defining G) with rotations (cf. Section 2.2), then, since the circular composition of maps is well-defined, we make G into a map M such that $G = Graph(M)$. The restriction to planar maps works as for planar graphs.

Theorem 31: Let $\mathcal{C}$ be any of the classes $\mathcal{PSC}$, $\mathcal{G}_{2,2}, \mathcal{PG}_{2,2}$, $\mathcal{MSC}$, $\mathcal{PMSC}$, $\mathcal{M}_{2,2}$ or $\mathcal{PM}_{2,2}$. Each element of $\mathcal{C}$ has a unique atomic decomposition over $\mathcal{C}$ that is based on the atomic decomposition of the corresponding underlying graph in $\mathcal{SC}$. From it, we get terms over $\mathcal{C}$ that define the considered graph or map and vice versa.

A similar statement holds for the corresponding undirected graphs and maps in $2\mathcal{EC}$, $\mathcal{G}_4$, $\mathcal{P}2\mathcal{EC}$, $\mathcal{M}_4$, $\mathcal{PM}_4$ etc. with the notion of atomic skeleton instead of that of atomic decomposition. (See Definition 19 and the following discussion).

A graph or a map is *labelled* if some of its vertices and edges have a label. Circular composition extends to labelled graphs and maps because it does not modify vertices and edges (it modifies only incidences). In particular, it does not fuse vertices or edges (which would raise the question of choosing a new label for a vertex or an edge resulting from a fusion). It follows that all results about circular composition and atomic decompositions extend in a straightforward manner to labelled objects. In particular, this extension of Theorem 31 is applicable to knot diagrams (see Section 4 about applications to knot diagrams).

3.7 Tree-width

If $G \in \mathcal{SC} \cup 2\mathcal{EC}$ we denote by $twd^{atom}(G)$ the maximum tree-width of its atoms. This notion is well-defined for graphs in $2\mathcal{EC}$ because their atoms are well-defined up to isomorphism. In particular, $twd^{atom}(Und(G)) = twd^{atom}(G)$ if $G \in \mathcal{SC}$.

Proposition 32 : The tree-width of a graph $G \in \mathcal{SC} \cup 2\mathcal{EC}$ is $twd^{atom}(G)$ if this number is at least 2. Otherwise it is 1 or 2.

Proof: It is enough to consider the case of graphs in $\mathcal{SC}$ since tree-width does not depend on edge directions and the function twd^{atom} is invariant under the replacement of $G \in \mathcal{SC}$ by $Und(G)$.

Consider $G = H \boxplus_{e,f} K \in \mathcal{SC}$. Tree-decompositions of H and K of respective widths p and q can be combined into one of width $\max\{2, p, q\}$. Then, from Algorithm 18 that constructs a term from an atomic decomposition $\mathcal{D}$, and Proposition 3(2) which shows that a circular composition can be expressed in terms of binary compositions, we get that if $G = G(\mathcal{D})$, then its tree-width is at most $\max\{2, twd^{atom}(G)\}$. Since every atom is a minor of G as observed in the proof of Proposition 2(2), we get that the tree-width of G is at least $twd^{atom}(G)$, hence it is $twd^{atom}(G)$ if this value is at least 2. The second assertion follows from these facts. $\square$

Remark **:** We have no similar upper-bound to the clique-width of a graph, directed or not, expressible in function of the maximal *clique-width* of its atoms (see Chapters 2 of [CouEng] for the definition): consider graphs G built from a clique K_n for $n \geq 4$ by subdividing some edges; they are 2-edge connected and their atoms are K_n and single vertices with one loop; these atoms have clique-width 2 and 1 respectively, but the graphs G have unbounded clique-width. A similar counter-example can be built with strongly connected tournaments instead of cliques.

3.8 Counting Eulerian orientations

The number of Eulerian orientations of a 4-regular graph is the value at $(0, -2)$ of its Tutte polynomial, and for a planar 4-regular graph, it is also related to the value at (3,3) of this polynomial for an associated planar graph (see [LV]). We consider how to compute it.

Definition 33:

If G is a directed graph and $F \subseteq E_G$, we denote by $G(F)$ the graph obtained by reversing the orientation of each edge in F. That is, we let $V_{G(F)} = V_G$, $E_{G(F)} = E_G$ and $vert_{G(F)}$ be defined by:

$vert_{G(F)}(e) = vert_G(e)$ if $e \notin F$,

$vert_{G(F)}(e) = (y, x)$ if $e \in F$ and $vert_G(e) = (x, y)$.

We call $G(F)$ the *reorientation defined by* F and say it is a $\mathcal{G}_{2,2}$*-orientation* of $Und(G)$ if $G(F) \in \mathcal{G}_{2,2}$. If $G \in \mathcal{G}_{2,2}$, we denote by $\sharp(G)$ the number of sets of edges F such that $G(F) \in \mathcal{G}_{2,2}$. We call such a graph $G(F)$ a $\mathcal{G}_{2,2}$*-reorientation* of G. If $G \in \mathcal{G}_4$, we denote by $\sharp(G)$ the number $\sharp(H)$ where H is any orientation of G. For the graph **8** belonging to $\mathcal{G}_{2,2}$ shown twice on the left of Figure 6, we have $\sharp(\mathbf{8}) = 4$ because each loop can oriented in two ways. We now consider the problem of computing $\sharp(G)$ for $G \in \mathcal{G}_4$.

Proposition 34: For every graph $G \in \mathcal{G}_{2,2} \cup \mathcal{G}_4$ with set of atoms $\mathcal{A}$, we have:

$$\sharp(G) = 2. \prod_{H \in \mathcal{A}} (\sharp(H)/2).$$

Proof: We consider $G \in \mathcal{G}_{2,2}$. We first observe that if $G = H \boxplus_{e,f} K$, then $\sharp(G) = \sharp(H).\sharp(K)/2$, because changing the orientation of edge e in H forces to change the orientation of f in K. Apart from this constraint, the new orientations of H and K can be chosen independently. Since fixing the orientation of one edge divides by 2 the number of possibilities, we get the equality. The general case of G expressed by a term over atoms and written with binary composition operations follows by an induction on the number of such operations. We have the same equality for $G \in \mathcal{G}_4$. $\square$

This proposition shows that we need only compute $\sharp(G)$ for atoms G. (Actually, it holds, and the proof is the same, if we replace $\mathcal{G}_{2,2}$ by $\mathcal{SC}$, $\mathcal{G}_4$ by $2\mathcal{EC}$ and we let $\sharp(G)$ be the number of strongly connected orientations of G.) For computing these numbers, we give a (time-consuming) recursive algorithm that works for all graphs in $\mathcal{G}_4$ and not only for atoms. (We would gain nothing by starting from an Eulerian orientation of these graphs).

We need some notation relative to a graph $G \in \mathcal{G}_4$. If x is a vertex incident with a single loop, then G has edges $x - y$ and $x - z$ (with possibly $y = z$) and we define $G\backslash^0 x$ as the graph $G - x$ augmented with an edge between y and z (a loop if $y = z$). Otherwise, x is incident with four edges $x - y_1$, $x - y_2$, $x - y_3$ and $x - y_4$ (we may have equalities among y_1, y_2, y_3 and y_4) and we define

$G\backslash^{1,2} x$ as $G - x$ augmented with edges $y_1 - y_2$ and $y_3 - y_4$,

$G\backslash^{1,3} x$ as $G - x$ augmented with edges $y_1 - y_3$ and $y_2 - y_4$,

$G\backslash^{1,4} x$ as $G - x$ augmented with edges $y_1 - y_4$ and $y_2 - y_3$.

Proposition 35: Let $G \in \mathcal{G}_4$. The number $\sharp(G)$ can be computed by the following equalities:

$\sharp(Und(\mathbf{8})) = 4$,

$\sharp(G) = 2.\sharp(G\backslash^0 x)$ if x is incident with a single loop,

$\sharp(G) = \sharp(G\backslash^{1,2}x) + \sharp(G\backslash^{1,3}x) + \sharp(G\backslash^{1,4}x)$ if x is not incident with a loop.

Proof: The first equality is an observation. For the second one, consider x incident with a single loop and two edges $x - y$ and $x - z$. Every $\mathcal{G}_{2,2}$-reorientation of G determines (by transitivity over x) a $\mathcal{G}_{2,2}$-reorientation of $G\backslash^0 x$. Conversely, the choice of an orientation of the edge $y - z$ in $G\backslash^0 x$ determines an orientation for the two edges $x - y$ and $x - z$ (that gives the orientation of $y - z$ by transitivity over x). However, the loop can be oriented in two ways. These facts give the second equality.

For the third one, observe that there are 6 ways to orient in G the edges incident with x so that x has indegree and outdegree 2. One of them is $x \leftarrow y_1$, $x \leftarrow y_2$, $x \rightarrow y_3$, $x \rightarrow y_4$. This particular orientation yields (by transitivity over x) one orientation of two edges of $G\backslash^{1,3}x$ namely, $y_1 \rightarrow y_3$ and $y_2 \rightarrow y_4$ and one of two edges of $G\backslash^{1,4}x$ namely $y_1 \rightarrow y_4$ and $y_2 \rightarrow y_3$ but none of $G\backslash^{1,2}x$. A similar situation holds for each of the 5 other possible orientations of the four edges of G incident with x. Hence, we get a bijection of the set of all $\mathcal{G}_{2,2}$-orientations of G and the (disjoint) union of the sets of $\mathcal{G}_{2,2}$-orientations of $G\backslash^{1,2}x, G\backslash^{1,3}x$ and $G\backslash^{1,4}x$. Hence, we have the third equality.$\square$

3.9 Using MSO logic

The following remarks are intended for the readers who know how to express graph properties and graph transformations in monadic second-order logic. We refer the reader to the book [CouEng] and to the articles [Cou99] and [Cou06] for constructions related to the topic of the present article.

The property that a set of edges F of a directed graph G defines a strongly connected reorientation (or a $\mathcal{G}_{2,2}$-reorientation) is expressible by a monadic second-order formula $\varphi(F)$ using *edge set quantifications* (called an *MS_2 formula*). The number $\sharp(G)$ is the cardinality of the set $\{F \mid F \subseteq E_G, G \models \varphi(F)\}$. Such a number can be computed by a fixed-parameter linear algorithm parameterized by tree-width (where arithmetical operations are considered as taking unit time).

We now explain how to define atomic decompositions by monadic second-order formulas. Let $G \in \mathcal{SC} \cup 2\mathcal{SC}$. That a vertex set X is a split is expressible by an MS_2 formula and so is the property that two edges are equivalent. The atomic sets of vertices W can thus be characterized as those that satisfy an MS_2 formula $\psi(W)$ (with W as single free set variable).

We have shown how the cactus of the atomic decomposition can be constructed from the atomic partition of V_G and the incidence function $vert_G$. With the terminology of [CouEng], [Cou99] and [Cou06], this means that the atomic decomposition of $G \in \mathcal{SC}$ and the atomic skeleton of $G \in 2\mathcal{SC}$ can be defined from G by a *monadic second-order transduction* whose formulas use edge set quantifications. (For undirected graphs, MS_2 formulas can specify all orientations, and in particular the strongly connected ones, cf. [CouEng]).

Similar construction have been given in [Cou99], [Cou96] and [Cou06] for, respectively, the Tutte decomposition (defined in Section 2.1), the *modular decomposition* and the *split decomposition*. However, the transductions of [Cou96] and [Cou06] use an auxiliary linear order on the vertex set of the considered graph, whereas here and in [Cou99] this is not needed.

4 Related decompositions and open problems

We now compare the atomic decomposition with two closely related decompositions of strongly connected graphs. We also mention decompositions of knot diagrams and matroids.

We first consider graphs in $\mathcal{SC}'$, the class of simple, loop-free and strongly connected graphs. A graph with one vertex and no edge is in this class.

The split decomposition by Cunnigham.

Cunnigham has defined in [Cun] a canonical decomposition of graphs in $\mathcal{SC}'$ based on the following notion of split: a *C-split* of a graph is a bipartition $\{A, B\}$ of its vertex set such that $|A|, |B| \geq 2$, the edges from A to B form a complete bipartite subgraph and so do those from B to A. A graph is *C-prime* if it has no C-split. Hence, if $G = H \boxplus_{e,f} K, |V_H|, |V_K| \geq 2$, then $\{V_H, V_K\}$ is a C-split. It follows that a C-prime graph (in $\mathcal{SC}'$) of minimal degree at least 3 is an atom. (We need the degree restriction because if x is a vertex of degree 2, then $\{x\}$ is a split but $\{\{x\}, V_G - \{x\}\}$ is not a C-split). The decomposition of [Cun] can thus be considered as a refinement of the atomic decomposition because certain atoms may be further decomposed. We leave for future work to examine if the construction of the atomic decomposition is an interesting intermediate step for the construction of the split decomposition (see [Joe, GPTCa] for algorithms).

An axiomatic description of certain canonical graph decompositions is presented in [CunEdm]. The Tutte decomposition and the split decomposition are two instances. Although the atomic decomposition is close to these two ones, we could not formalize it in that framework.

A decomposition by Knuth

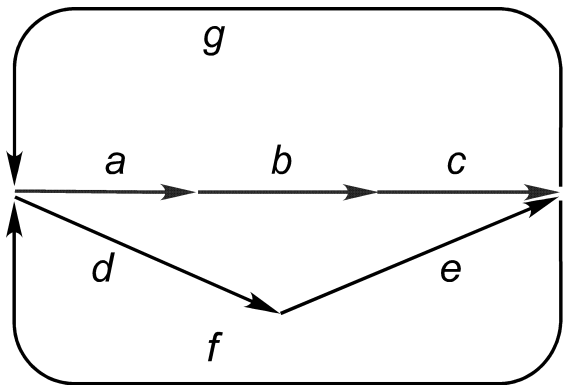

Figure 13: A graph with several decompositions

Knuth defines in [Knu] a decomposdition of the graphs in $\mathcal{SC}'$ that is close to, but actually different from, the atomic decomposition. With our terminology and notation, the basic lemma of [Knu] states that every graph G in $\mathcal{SC}'$

(i) either is reduced to a vertex without loop,

(ii) or has one edge e such that $G - e \in \mathcal{SC}'$,

(iii) or can be expressed as $G = \boxplus_{e_1,...,e_k}(H_1, ..., H_k)$ where for each i, the graph $H_i - e_i$ belongs to $\mathcal{SC}'$. (Here, e_i can be a loop of H_i).

Conditions (ii) and (iii) are not exclusive.

Since $G - e$ and the graphs $H_i - e_i$ are in $\mathcal{SC}'$, the same lemma applies to them, so that we obtain a hierarchical decomposition of every graph in $\mathcal{SC}'$. This decomposition looks similar to the atomic decomposition, however there are important differences: its undecomposable elements are the vertices (this is clear) and, it is not canonical.

For proving the latter point, we define $\{e\}$ and $\{e_1, ..., e_k\}$ as the *keys* of the decomposition steps based on cases (ii) and (iii) respectively. A decomposition is thus represented by a rooted tree whose nodes are the keys of its steps; these keys form a partition of the set of edges. Consider the graph of Figure 13 with edges $a, ..., g$. (It is an atom). It can be decomposed by using successively keys $\{g\}$, $\{a, b, c\}$ and $\{d, e, f\}$. Another possibility is to use $\{f\}$, $\{a, b, c\}$ and $\{d, e, g\}$, and yet another one is $\{g\}$, $\{d, e\}$ and $\{a, b, c, f\}$. The corresponding decompositions cannot be unified into a single one.

The algorithmic properties and structural uses of this decomposition remain to be investigated. Knuth gives two examples of properties of strongly connected graphs that are provable by induction on this decomposition but are not (or not so easily) from the atomic decomposition. The reason is that the basic elements of his decomposition are vertices and not atoms

Knots and Knot diagrams

The reader will find the definitions and basic facts in the books [Ada, Bol, GodRoy] (or in Wikipedia, look for "knot theory"). We consider oriented knots. A knot diagram is essentially a (2,2)-regular planar map whose vertices are labelled by **over** or **under**, to indicate the type of crossing (we omit technical details, see [Cou13]). We let k denote the mapping from a diagram to the corresponding knot. Knots can be composed by a binary (multivalued) operation denoted by # (in [Ada], Section 1.2) such that for every two knot diagrams D and D', every edge e of D and every edge f of D', we have $k(D \boxplus_{e,f} D') = k(D)\#k(D')$. It follows that the diagrams of prime knots are atoms. We leave as a research topic to investigate the relationships between factorizations of knots and atomic decompositions of knots diagrams.

Matroids

As observed at the end of Definition 1, circular composition is well-defined on matroids (see [Tru]). For matroids M, N and elements e of M and f of N, we define $M \boxplus_{e,f} N$ as the matroid whose circuits are those of M not containing e, those of N not containing f and the sets $C \cup D$ where C is a circuit of M containing e and D is one of N containing f. This operation is, in a certain sense, equivalent to a 2-sum of M and N. Determining if this operation is interesting for matroids remains to be investigated.

Some questions

The numbers of strongly connected orientations and Eulerian orientations are particular values of the Tutte polynomial ([Bol]). This suggests the following question:

> *Does the atomic decomposition of a graph or a matroid help for computing its Tutte polynomial?*

We have defined circular composition for directed graphs and used it only for strongly connected graphs.

> *Does it yield interesting decompositions for directed graphs of wider classes ?*

We have shown how to construct efficiently the atomic decomposition of a strongly connected graph from its Tutte decomposition.

> *Is it possible, conversely, to construct efficiently the former from the latter?*

In the companion article [Cou13], we use atomic decompositions to describe (and denumber) the closed curves in the plane that have a given associated Gauss word.

Acknowledgements : I thank the late M. Las Vergnas for useful comments and the reference to Knuth's article.

References

[Ada] C. Adams, *The knot book*, A.M.S., Providence, Rhode Island, 2004.

[BanGut] J. Bang-Jensen and G. Gutin, *Digraphs: theory, algorithms and applications,* Springer, 2001.

[Bol] B. Bollobas, *Modern graph theory,* Springer, 2001.

[Cou96] B. Courcelle, The monadic second-order logic of graphs X: Linear orders, *Theoret. Comput. Science*, **160** (1996) 87-143.

[Cou99] B. Courcelle, The monadic second-order logic of graphs XI :Hierarchical decompositions of connected graphs, *Theoret. Comput. Sci.* **224** (1999) 35-58.

[Cou00] B. Courcelle, The monadic second-order logic of graphs XII: Planar graphs and planar maps, *Theoret. Comput.Sci.* **237** (2000) 1-32.

[Cou06] B. Courcelle, The monadic second-order logic of graphs XVI : Canonical graph decompositions, *Logical Methods in Computer Science* **2** (2006) 1-46.

[Cou08] B. Courcelle, Circle graphs and Monadic Second-order logic, *Journal of Applied Logic* **6** (2008) 416-442.

[Cou13] B. Courcelle, All curves with a given Gauss word, In preparation.

[CouDus] B. Courcelle and V. Dussaux, Map genus, forbidden maps and monadic second-order logic, *The Electronic Journal of Combinatorics* **9**(1) (2002), R40. (See http://www.combinatorics.org/Volume_9/Abstracts/v9i1r40.html).

[CouEng] B. Courcelle and J. Engelfriet, *Graph structure and monadic second-order logic, a language theoretic approach*, Volume **138** of *Encyclopedia of mathematics and its application*, Cambridge University Press, June 2012.

[CMR] B. Courcelle, J. Makowsky and U. Rotics, Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width. *Theory Comput. Syst.* **33** (2000) 125-150.

[CunEdm] W. Cunnigham and J. Edmonds, A combinatorial decomposition theory, *Can. J. of Math.*, XXXII (1980) 734–765.

[Cun] W. Cunnigham, Decomposition of directed graphs, *SIAM. J. on Algebraic and Discrete Methods* **3** (1982) 214-228.

[Die] R. Diestel, *Graph theory*, Springer-Verlag, 4th edition, 2010. (See http://diestel-graph-theory.com).

[DF] R. Downey, M. Fellows, *Parameterized complexity*, Springer-Verlag, 1999.

[FG] J. Flum, M. Grohe, *Parametrized complexity theory*, Springer, 2006.

[FOM] H. de Fraysseix and P. Ossona de Mendez, On a characterization of Gauss codes, *Discrete Comput. Geom.* **22** (1999) 287-295.

[Gal] F. Maffray and M. Preissmann, A translation of T. Gallai's paper: "Transitiv orientierbare Graphen". In: *Perfect graphs*, J.L. Ramirez-Alfonsin and B.A. Reed Eds., J. Wiley (2001), 25-66.

[GPTCa] E. Gioan, C. Paul, M. Tedder and D. Corneil, Practical and efficient split decomposition, *Algorithmica*, March 2013 (online publication).

[GPTCb] E. Gioan, C. Paul, M. Tedder and D. Corneil, Practical and efficient circle graph recognition, *Algorithmica*, March 2013 (online publication).

[GodRoy] C. Godsil and G. Royle, *Algebraic Graph Theory*, Springer, 2001.

[HopTar1] J. Hopcroft and R. Tarjan, Efficient algorithms for graph manipulation. *Communications of the ACM* **16**(6) (1973) 372–378.

[HopTar2] J. Hopcroft and R. Tarjan, Dividing a graph into triconnected components, *SIAM Journal on Computing* **2** (1973) 135–158.

[Joe] B. Joeris *et al.*, $O(m.\log(n))$ split decomposition of strongly connected graphs, *Discrete Applied Maths* **158** (2010) 779-799.

[Knu] D. Knuth, Wheels within wheels, *J. of Comb. Theory B* **16** (1974) 42-46.

[LV] M. Las Vergnas, On the evaluation at (3,3) of the Tutte polynomial of a graph, *J. Comb. Th. B* **45** (1988) 367-372.

[LM] L. Lovasz and M. Marx, A forbidden substructure characterization of Gauss codes, *Bulletin of the AMS* **82** (1976) 121-122.

[MoRad] R.H. Möhring and F. Radermacher, Substitution Decomposition for Discrete Structures and Connections with Combinatorial Optimization, *Ann. Discrete Mathematics* **19** (1984) 257-356.

[MT] B. Mohar and C. Thomassen, *Graphs on Surfaces*, The Johns Hopkins University Press, 2001.

[Ros] P. Rosenstiehl, Solution algébrique du problème Gauss sur la permutation des points d'intersection d'une ou plusieurs courbes fermées du plan, *C.R. Acad. Sc. Paris, Sér. A* **283** (1976) 551-553.

[Tru] K. Truemper, *Matroid decomposition*, 2009, free electronic edition on: http://www.emis.de/monographs/md/

[Tut] W. Tutte, *Graph Theory*, Addison-Wesley, 1984.

Appendix : Index of some definitions.

Isomorphisms

Isomorphism of graphs : notation $G \cong H$, Section 2.1.

v-isomorphism of graphs (identity on vertices) : Section 2.1.

Automorphisms and v-automorphisms : Section 2.1.

Isomorphism of atomic decompositions : notation $\mathcal{D} \cong \mathcal{D}'$, Definition 16, Section 3.3, and discussion after Definition 19

Homeomorphisms

Homeomorphisms that preserve the orientation correspond to the equality of maps; general homeomorphisms correspond to equivalence of maps : Section 2.2.

Equivalences

Und-equivalent maps, for defining maps of undirected graphs : Section 2.2.

Equivalent terms (same value) : Section 3.1, Definition 5.

Equivalent vertices (not separated by a split) : notation $x \sim_G y$, Section 3.2, Definition 6.

Equivalent edges (belong to a 2-cut) : notation $x \approx_G y$, Section 3.2, Definition 11.