Journal of Integer Sequences, Vol. 22 (2019),
Article 19.2.5

Solving Computational Problems in the Theory of Word-Representable Graphs

Özgür Akgün and Ian Gent
School of Computer Science
University of St Andrews
St Andrews, Fife KY16 9SX
United Kingdom
ozgur.akgun@st-andrews.ac.uk
ian.gent@st-andrews.ac.uk

Sergey Kitaev
School of Computer and Information Sciences
University of Strathclyde
Glasgow, G1 1HX
United Kingdom
sergey.kitaev@cis.strath.ac.uk

Hans Zantema
Department of Computer Science
Eindhoven University of Technology
P. O. Box 513
5600 MB Eindhoven
The Netherlands
H.Zantema@tue.nl

Abstract

A simple graph $G = (V, E)$ is word-representable if there exists a word w over the alphabet V such that letters x and y alternate in w iff $xy \in E$. Word-representable graphs generalize several important classes of graphs. A graph is word-representable iff it admits a semi-transitive orientation. We use semi-transitive orientations to enumerate connected non-word-representable graphs up to the size of 11 vertices, which led to a correction of a published result. Obtaining the enumeration results took 3 CPU years of computation.

Also, a graph is word-representable iff it is k-representable for some k, that is, if it can be represented using k copies of each letter. The minimum such k for a given graph is called graph's representation number. Our computational results in this paper not only include distribution of k-representable graphs on at most 9 vertices, but also have relevance to a known conjecture on these graphs. In particular, we find a new graph on 9 vertices with high representation number. Also, we prove that a certain graph has highest representation number among all comparability graphs on odd number of vertices.

Finally, we introduce the notion of a k-semi-transitive orientation refining the notion of a semi-transitive orientation, and show computationally that the refinement is not equivalent to the original definition, unlike the equivalence of k-representability and word-representability.

1 Introduction

Letters x and y alternate in a word w if after deleting in w all letters but the copies of x and y we either obtain a word $xyxy\cdots$ (of even or odd length) or a word $yxyx\cdots$ (of even or odd length). For example, the letters 2 and 5 alternate in the word 11245431252, while the letters 2 and 4 do not alternate in this word. A simple graph $G = (V, E)$ is *word-representable* if there exists a word w over the alphabet V such that letters x and y alternate in w iff $xy \in E$. By definition, w *must* contain *each* letter in V. We say that w *represents* G, and that w is a *word-representant.*

The definition of a word-representable graph works both for vertex-labeled and unlabeled graphs because any labeling of a graph G is equivalent to any other labeling of G with respect to word-representability (indeed, the letters of a word w representing G can always be renamed). For example, the graph to the left in Figure 1 is word-representable because its labeled version to the right in Figure 1 can be represented by 1213423. For another example, each *complete graph* K_n can be represented by any permutation π of $\{1, 2, \ldots, n\}$, or by π concatenated any number of times. Also, the *empty graph* E_n (also known as *edgeless graph*, or *null graph*) on vertices $\{1, 2, \ldots, n\}$ can be represented by $12\cdots(n-1)nn(n-1)\cdots 21$, or by any other permutation concatenated with the same permutation written in the reverse order.

We note that the class of word-representable graphs is *hereditary*. That is, removing a vertex v in a word-representable graph G results in a word-representable graph G'. Indeed,

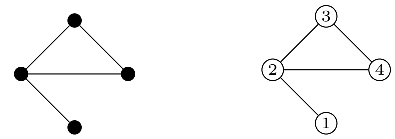


Figure 1: An example of a word-representable graph

if w represents G then w with v removed represents G'.

There is a long line of research on word-representable graphs (see, e.g. [1, 3, 4, 5, 7, 10, 12, 13, 14, 15, 18, 21, 25]) that is summarized in [19, 20]. The roots of the theory of word-representable graphs are in the study of the celebrated *Perkins semigroup* in [23], which has played a central role in semigroup theory since 1960, particularly as a source of examples and counterexamples. However, the significance of word-representable graphs is in the fact that they generalize several important classes of graphs such as 3-colorable graphs, comparability graphs and circle graphs.

One of the key tools to study word-representable graphs is the notion of a semi-transitive orientation to be defined next.

1.1 Semi-transitive orientations

The notion of a semi-transitive orientation was introduced in [14, 15], but we follow [20, Section 4.1] to introduce it here. A graph $G = (V, E)$ is *semi-transitive* if it admits an *acyclic* orientation such that for any directed path $v_1 \to v_2 \to \cdots \to v_k$ with $v_i \in V$ for all i, $1 \leq i \leq k$, either

- there is no edge $v_1 \to v_k$, or
- the edge $v_1 \to v_k$ is present and there are edges $v_i \to v_j$ for all $1 \leq i < j \leq k$. In other words, in this case, the (acyclic) subgraph induced by the vertices $v_1, \dots, v_k$ is transitive (with the unique source v_1 and the unique sink v_k).

We call such an orientation *semi-transitive*. In fact, the notion of a semi-transitive orientation is defined in [14, 15] in terms of *shortcuts* as follows. A *semi-cycle* is the directed acyclic graph obtained by reversing the direction of one edge of a directed cycle in which the directions form a directed path. An acyclic digraph is a shortcut if it is induced by the vertices of a semi-cycle and contains a pair of non-adjacent vertices. Thus, a digraph on the vertex set $\{v_1, \dots, v_k\}$ is a shortcut if it contains a directed path $v_1 \to v_2 \to \cdots \to v_k$, the edge $v_1 \to v_k$, and it is missing an edge $v_i \to v_j$ for some $1 \leq i < j \leq k$; in particular, we must have $k \geq 4$, so that any shortcut is on at least four vertices. Clearly, this definition is just another way to introduce the notion of a semi-transitive orientation presented above.

It is not difficult to see that all transitive (that is, comparability) graphs are semi-transitive, and thus semi-transitive orientations are a generalization of transitive orientations. A key theorem in the theory of word-representable graphs is presented next.

Theorem 1 ([14, 15]). *A graph G is word-representable if and only if it admits a semi-transitive orientation (that is, if and only if G is semi-transitive).*

A corollary to Theorem 1 is the useful fact that any 3-colorable graph is word-representable.

1.2 Comparability graphs and permutational representation

An orientation of a graph is *transitive* if the presence of edges $u \to v$ and $v \to z$ implies the presence of the edge $u \to z$. An unoriented graph is a *comparability graph* if it admits a transitive orientation. A graph $G = (V, E)$ is *permutationally representable* if it can be represented by a word of the form $p_1 \cdots p_k$ where p_i is a permutation.

The following theorem is an easy corollary of the fact that any partially ordered set can be represented as intersection of linear orders, and that a linear order can be represented by a permutation.

Theorem 2 ([23]). *A graph is permutationally representable if and only if it is a comparability graph.*

Permutational representation of a graph is a special case of uniform representation, to be discussed next.

1.3 Uniform representations

A word w is *k-uniform* if each letter in w occurs k times. For example, the word 342321441231 is 3-uniform, while 43152 is a 1-uniform word (a permutation). A graph G is *k-word-representable*, or *k-representable* for brevity, if there exists a k-uniform word w representing it. We say that w *k-represents* G. A somewhat surprising fact establishes equivalence of word-representability and uniform word-representability:

Theorem 3 ([21]). *A graph is word-representable iff it is k-representable for some k.*

Thus, in the study of word-representable graphs, word-representants can be assumed to be uniform. *Graph's representation number* is the *least* k such that the graph is k-representable. For non-word-representable graphs, we let $k = \infty$. It is known [15] that the upper bound on the length of a shortest uniform word-representant for a graph G on n vertices is essentially $2n^2$, that is, one needs at most $2n$ copies of each letter to represent G. We let $\mathcal{R}(G)$ denote G's representation number and $\mathcal{R}_k = \{G : \mathcal{R}(G) = k\}$.

The class of complete graphs is clearly the class of graphs with representation number 1. Further, the class of graphs with representation number 2 is precisely the class of *circle graphs* without complete graphs, that is, the *intersection graphs* of sets of chords of a circle [14]. The later fact implies that, in particular, the number of connected graphs on n vertices with representation number 2 (see Table 1) is precisely the number of connected circle graphs given by the sequence A156808 in [28]. Unlike the cases of graphs with representation numbers 1 or 2, no characterization of graphs with representation number 3, or higher, is known.

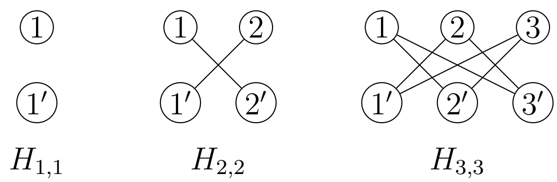


Figure 2: Crown graphs

However, there is a number of interesting results on graphs with representation numbers higher than 2, some of which we mention next (see [19] for references to the original sources in relation to the results, and for more results in this direction).

The representation number of the *Petersen graph* and any *prism* is 3. Also, for every graph G there are infinitely many 3-representable graphs H that contain G as a *minor*. Such a graph H can be obtained from G by subdividing *each* edge into *any* number of, but *at least* three edges.

1.4 Graphs with high representation number

As for graphs with high representation number, only *crown graphs* and graphs G_n based on them (see the definitions below) were known until this paper; Figure 5 gives an example of another such graph. A *crown graph* (also known as a *cocktail party graph*) $H_{n,n}$ is obtained from the complete bipartite graph $K_{n,n}$ by removing a *perfect matching*. That is, $H_{n,n}$ is obtained from $K_{n,n}$ by removing n edges such that *each* vertex was incident to *exactly one* removed edge. See Figure 2 for examples of crown graphs.

By Theorem 2, $H_{n,n}$ can be represented by a concatenation of permutations, because $H_{n,n}$ is a comparability graph (to see this, just orient all edges from one part to the other). In fact, $H_{n,n}$ is known to require n permutations to be represented (the maximum possible amount for a comparability graph on $2n$ vertices by the well known theorem on the poset dimensions by Hiraguchi [16]). However, we can provide a shorter representation for $H_{n,n}$, to be discussed next, which is still long (linear in n).

Note that $H_{1,1} \in \mathcal{R}_2$. Further, $H_{2,2} \neq K_4$, the complete graph on four vertices, and thus $H_{2,2} \in \mathcal{R}_2$ because it cannot be represented by a permutation but can be 2-represented by $2'12'121'21'$. Also, $H_{3,3} = C_6$, a *cycle graph*, which belongs to $\mathcal{R}_2$ as is shown, e.g. in [19, 20]. Finally, $H_{4,4} \in \mathcal{R}_3$ because $H_{4,4}$ is a prism (it is the 3-dimensional cube). The following theorem gives the representation number $\mathcal{R}(H_{n,n})$ in the remaining cases.

Theorem 4 ([10])**.** *If $n \geq 5$, then the representation number of $H_{n,n}$ is $\lceil n/2 \rceil$.*

Conjecture 5. $H_{n,n}$ has the highest representation number among all bipartite graphs on $2n$ vertices.

The graph G_n is obtained from a crown graph $H_{n,n}$ by adding an apex (all-adjacent vertex). See Figure 3 for the graph G_4. It turns out that G_n is the *worst* known word-

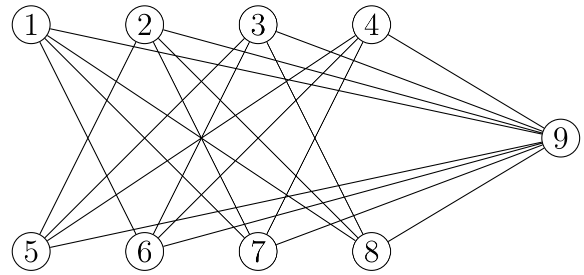


Figure 3: The graph G_4 with representation number 4

representable graph in the sense that it requires the maximum number of copies of each letter to be represented, as recorded in the following theorem.

Theorem 6 ([21]). *The representation number of G_n is $\lfloor (2n+1)/2 \rfloor$.*

It is unknown whether there exist graphs on n vertices with representation number between $\lfloor n/2 \rfloor$ and essentially $2n$ (the known upper bound), but one has the following conjecture.

Conjecture 7. G_n has the highest representation number among all graphs on $2n+1$ vertices.

It is easy to see that G_n is a comparability graph (just make the apex to be a source, or a sink, and orient the remaining crown graph from one part to the other). Surprisingly, the following result on G_n does not seem to be recorded in the literature.

Theorem 8. *G_n has the highest representation number among all comparability graphs on $2n+1$ vertices.*

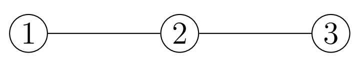
Proof. Let G be a comparability graph on $2n+1$ vertices. By Theorem 2, G can be represented by a concatenation of permutations, which is equivalent to representing the partially ordered set corresponding to G by intersection of linear orders. It is known [16] that for any finite poset P, the dimension of P is at most half of the number of elements in P. Thus, the number of permutations required to represent G cannot exceed n, which in turn implies that $\mathcal{R}(G) \leq n$ (dropping the requirement to represent G permutationally, we can only shorten a word-representant). Thus, by Theorem 6, $\mathcal{R}(G) \leq \mathcal{R}(G_n)$. □

1.5 Organization of the paper

Our concern in this paper is word-representation of *connected* graphs, because a graph is word-representable if and only if each of its connected components is word-representable [20]. In Section 2 we explain our computational approach using satisfiability module theories (SMT) to study k-word-representable graphs and present the results obtained. In particular, we raise some concerns about Conjecture 7, while confirming it for graphs on at most 9 vertices. In Section 3 we present a complementary computational approach using constraint programming, enabling us count connected non-word-representable graphs. In particular, in Section 3 we report that using 3 years of CPU time, we found out that 64.65% of all connected graphs on 11 vertices are non-word-representable. Another important corollary of our results in Section 3 is the correction of the published result [19, 20] on the number of connected non-word-representable graphs on 9 vertices (see Table 2). In Section 4 we introduce the notion of a k-semi-transitive orientation refining the notion of a semi-transitive orientation, and show that 3-semi-transitively orientable graphs are not necessarily semi-transitively orientable. Finally, in Section 5 we suggest a few directions for further research and experimentation.

2 Finding word-representants by SMT

How to find a k-uniform word-representation of a given graph $G = (V, E)$? In this section we discuss how this can be done by means of *SMT*: *satisfiability modulo theories*. In particular, we focus on the *theory of linear inequalities*, and want to exploit the fact that current SMT solvers are strong in establishing whether a Boolean formula composed from $\wedge$, $\vee$, $\neg$ and linear inequalities admits a solution, and if so, finds one. Here by a solution we mean a choice for the values of the variables such that the formula yields true; if such a solution exists the formula is called '*satisfiable*', and the solution is called a '*satisfying assignment*'. So, our goal is to find such a Boolean formula for which any solution corresponds to a k-uniform word-representation of a given graph. For doing so, we need a way to express the unknown k-uniform word of length kn, where $n = \#V$ is the number of vertices in the graph in question, by a number of variables. This is done as follows. Number the vertices from 1 to n, and represent a word w that we are looking for by kn integer variables $A_{i,j}$, for $i = 1, \ldots, n$, $j = 1, \ldots, k$. The intended meaning of $A_{i,j}$ is the position of the j-th occurrence of symbol i in w, for $i = 1, \ldots, n$, $j = 1, \ldots, k$. For example, for the following graph



the word $w = 132312$ is a 2-uniform word-representing the graph, and is expressed by the values $A_{1,1} = 1$, $A_{1,2} = 5$, $A_{2,1} = 3$, $A_{2,2} = 6$, $A_{3,1} = 2$, $A_{3,2} = 4$.

Now, our formula is the conjunction of a number of requirements on these integer variables $A_{i,j}$ that all together describe a word w representing a given graph $G = (V, E)$. These requirements are:

- $A_{i,j} > 0$, for all $i = 1, \ldots, n$, $j = 1, \ldots, k$;
- $A_{i,j} \leq kn$, for all $i = 1, \ldots, n$, $j = 1, \ldots, k$;
- all $A_{i,j}$ are distinct (distinctness is a feature included in SMT format);

- for all $i_1 i_2 \in E$,

$$(A_{i_1,1} < A_{i_2,1} < A_{i_1,2} < A_{i_2,2} < \cdots < A_{i_1,j} < A_{i_2,j})$$
$$\vee\, (A_{i_2,1} < A_{i_1,1} < A_{i_2,2} < A_{i_1,2} < \cdots < A_{i_2,j} < A_{i_1,j});$$

- for all $i_1 i_2 \notin E$,

$$\neg(A_{i_1,1} < A_{i_2,1} < A_{i_1,2} < A_{i_2,2} < \cdots < A_{i_1,j} < A_{i_2,j})$$
$$\wedge\, \neg(A_{i_2,1} < A_{i_1,1} < A_{i_2,2} < A_{i_1,2} < \cdots < A_{i_2,j} < A_{i_1,j}).$$

.

So, for our graph above, the formula reads

$$A_{1,1} > 0 \wedge A_{1,2} > 0 \wedge A_{2,1} > 0 \wedge A_{2,2} > 0 \wedge A_{3,1} > 0 \wedge A_{3,2} > 0 \wedge$$
$$A_{1,1} \leq 6 \wedge A_{1,2} \leq 6 \wedge A_{2,1} \leq 6 \wedge A_{2,2} \leq 6 \wedge A_{3,1} \leq 6 \wedge A_{3,2} \leq 6 \wedge$$
$$\text{distinct}(A_{1,1}, A_{1,2}, A_{2,1}, A_{2,2}, A_{3,1}, A_{3,2}) \wedge$$
$$((A_{1,1} < A_{2,1} < A_{1,2} < A_{2,2}) \vee (A_{2,1} < A_{1,1} < A_{2,2} < A_{1,2})) \wedge$$
$$((A_{2,1} < A_{3,1} < A_{2,2} < A_{3,2}) \vee (A_{3,1} < A_{2,1} < A_{3,2} < A_{2,2})) \wedge$$
$$\neg(A_{1,1} < A_{3,1} < A_{1,2} < A_{3,2}) \wedge \neg(A_{3,1} < A_{1,1} < A_{3,2} < A_{1,2})).$$

For the values $A_{1,1} = 1$, $A_{1,2} = 5$, $A_{2,1} = 3$, $A_{2,2} = 6$, $A_{3,1} = 2$, $A_{3,2} = 4$ this formula yields true, as is found by the SMT solver Z3, yielding the 2-uniform word-representation $w = 132312$ of the graph.

Up to syntactic details (boolean operators are written as 'not', 'and', 'or', all operators are written in prefix notation), it is exactly this formula on which an SMT solver like Z3 [30] or YICES [29] can be applied, yielding 'satisfiable', and the corresponding satisfying assignment gives our values of $A_{i,j}$.

We wrote a tool doing this in a way where the internal use of an SMT solver is hidden from the user. It is available at

`http://www.win.tue.nl/~hzantema/reprnr.html` .

The tool reads a graph and then tries to find a k-representation for $k = 2, 3, 4, \ldots$ by building the formula as presented above and then calling an SMT solver. As soon as a satisfying assignment is found, the computation stops and the resulting values are transformed to the corresponding k-uniform word-representation, which is returned to the user. The tool is available both for Windows (calling the SMT solver Z3) and for Linux (calling the SMT solver YICES), together with several examples. Typically, for graphs like the cube, the prism on the triangle, Petersen graph, and G_4 (see below), the k-uniform word representing the graph is found in a second or less.

# of vertices	# of conn. graphs	representation number 1	2	3	4	> 4
3	2	1	1	0	0	0
4	6	1	5	0	0	0
5	21	1	20	0	0	0
6	112	1	109	1	0	1
7	853	1	788	39	0	25
8	11,117	1	8335	1852	0	929
9	261,080	1	117,282	88,838	2	54,957

Table 1: Distribution of connected graphs with representation number k, $1 \leq k \leq 4$, on at most 9 vertices. The cases $k = 2, 3$ are the sequences A319489 and A319490 in [28], respectively.

As this tool works quite quickly, it is feasible to run it on a great number of graphs. In particular, we ran it on all connected graphs on ≤ 9 vertices as they are available from

`http://users.cecs.anu.edu.au/~bdm/data/graphs.html` .

The results are listed in Table 1, where 'representation number > 4' means that no 4-representation exists, so either the representation number is > 4, or the graph is not word-representable (for which the representation number is ∞). However, as these numbers coincide with the respective numbers in Table 2, we conclude that only the latter occurs, and no word-representable graph exists on ≤ 9 vertices with representation number > 4.

The single graph on 6 vertices with representation number 3 is the prism on the triangle; the single non-word-representable graph on 6 vertices is the wheel on 5 vertices. The 39 graphs on 7 vertices with representation number 3 are given in Figure 4.

The most surprising result in Table 1 is the two graphs on 9 vertices with representation number 4. One of them was known before, namely, G_4 presented in Figure 3, and it was believed to be the only graph on 9 vertices with representation number 4. However, our computations have shown the existence of another such graph, namely the graph J_4 shown in Figure 5. We note that J_4 is a non-comparability graph, which is easy to check, while G_4 is. This may suggest that Conjecture 7 might not be true, since there are many more non-comparability graphs than comparability graphs, and one may expect finding those of them that have higher representation number than G_n. Having said that, we were not able to extend the construction of J_4 (in a natural way) to more than 9 vertices.

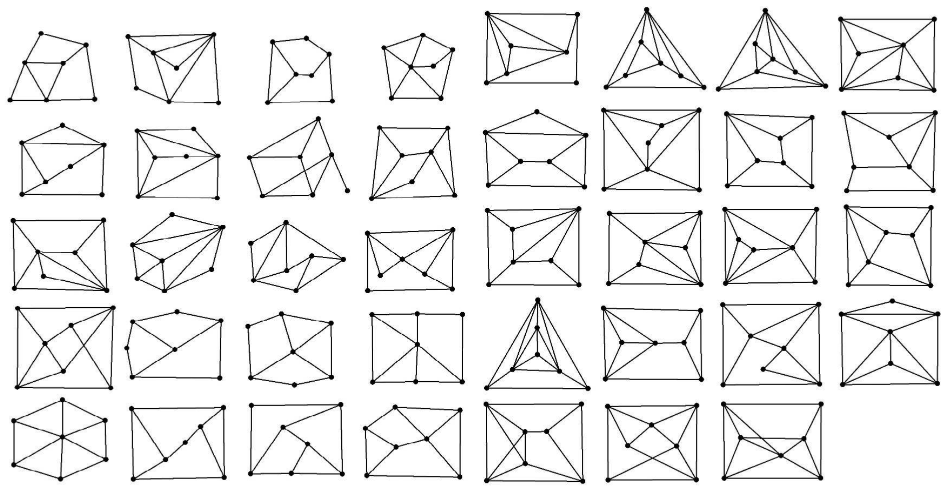


Figure 4: The 39 connected graphs on 7 vertices with representation number 3

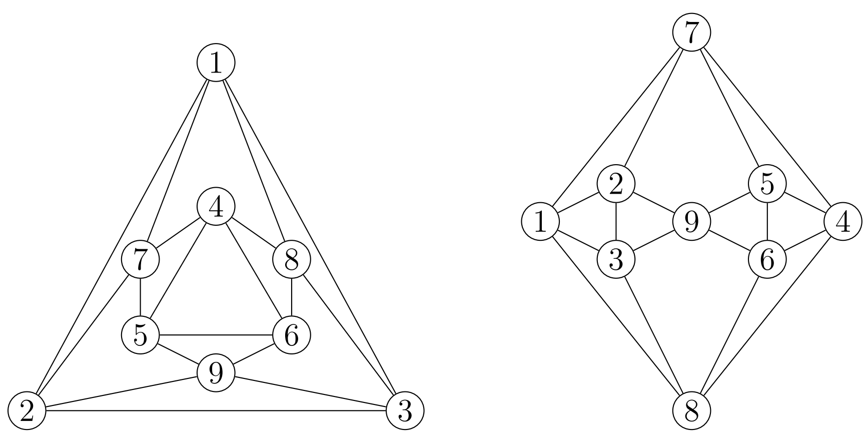


Figure 5: The graph J_4 with representation number 4. It is shown in two ways to demonstrate different symmetries.

3 Counting non-word-representable graphs using constraint programming

Similarly to our studies of k-word-representable graphs, we performed large computations using constraint programming [27] to count the numbers of non-word-representable connected graphs with up to 11 vertices. To do this, we used the constraint modelling tool Savile Row [26] and the constraint solver Minion [8]. These tools have been used successfully in the past to obtain novel enumerations of a variety of combinatorial structures including semigroups [6], equidistant frequency permutation arrays [17], and S-crucial and bicrucial permutations with respect to squares [9].

Our starting point was to model the concept of word-representability in a way similar to that when using SMT in Section 2. However, here we use Theorem 1 showing the equivalence between word-representability and semi-transitivity, so that semi-transitive orientations are now used to determine whether or not a graph is word-representable. As with SMT in Section 4, we use a boolean u_{ij} to indicate an undirected edge between i and j, and a boolean e_{ij} to indicate a directed edge from i to j. Moreover, we use a boolean t_{ij} to indicate the *transitive closure* of e, which is true when there is a path of directed edges from i to j. Most of the model expresses the appropriate linkages between these sets of variables. For example, constraints in the model express that t is the transitive closure of e. The acyclicity of e is elegantly expressed by each t_{ii} being false, i.e. no vertex being reachable from itself in the transitive closure. The final constraint expresses the property of semi-transitivity. This states that if two vertices are connected by a directed path, and there is an undirected edge between them, then all pairs of intermediate vertices must have a directed edge between them in the appropriate direction.

The model we used is shown in full in Figure 6. There are three points of detail about the model which deserve mention. First, this model neither check graphs for being connected, nor for being non-isomorphic to each other. This is not easy to do very efficiently in constraints, so instead we constructed a list of all connected undirected graphs with no two graphs being isomorphic, using the program `geng` [24]. Second, we originally modelled an undirected graph as an input to the constraint model, which was then checked for word-representability. However, this proved to be very inefficient as the vast majority of the constraint modelling processes was the same for each graph. Instead, we provide the constraint model with a list of graphs produced by `geng` and insist that the solution is one of those graphs. This is achieved in constraints using the 'table' constraint, which can be propagated very efficiently [2]. As well as saving work at the modelling stage, it also provides the capability to save work at the solving stage. For example, if all graphs remaining for consideration contain a certain undirected edge ij, the variable u_{ij} can be set true immediately. A major advantage of this approach is that it makes it particularly easy to parallelise the enumeration process, simply by splitting the list of distinct connected graphs into appropriately sized chunks. Finally, the line '`branching on [u]`' tells the constraint tools that we only wish to solve the problem once for each different assignment of `u`, i.e. for each undirected graph. Without this, any

# of vert.	# of conn. graphs	All non-word-representable graphs				
		Total	% of cand.	Time	Min.	Non-Min.
6	112	1	0.89%	3.0s	1	0
7	853	25	2.93%	4.0s	10	15
8	11,117	929	8.36%	26s	47	882
9	261,080	54,957	21.05%	29m	179	54,778
10	11,716,571	4,880,093	41.65%	74h	-	-
11	1,006,690,565	650,856,040	64.65%	1,100d	-	-

Table 2: The numbers of all non-word-representable connected graphs (the sequence A290814 in [28]), as well as the numbers of such graphs, called *non-minimal*, that include smaller non-word-representable subgraphs, and those, called *minimal*, that do not. The percentage of non-word-representable connected graphs to all connected graphs is given to 2 decimal places. Times indicate the CPU time used to compute all non-word-representable connected graphs, to 2 significant figures in an appropriate unit (seconds, minutes, hours, days). The time to count minimal/non-minimal connected graphs is not shown.

graph admitting more than one semi-transitive orientation would be repeated in the output, wasting both search time and necessitating extra work in removing duplicates.

Results of our computations are shown in Table 2. Note that in one case numbers are different to those previously reported. The true number of connected non-word-representable graphs on 9 vertices is 54,957, *not* 68,545 as was reported in [19, 20] (which was a copy/paste mistake).

It is also interesting to identify minimal non-word-representable graphs of each size, i.e. graphs containing no non-word-representable strict induced subgraphs. To do this, we stored all non-word-representable graphs of each size. After computing with `geng` all possible graphs with one more vertex, we eliminate graphs containing one of the stored graphs as an induced subgraph. We did this with a simple constraint model which tries to find a mapping from the vertices of the induced subgraph to the vertices of the larger graph, and if successful discards the larger graph from consideration. This enabled us to count all minimal non-word-representable graphs of each size up to 9, which is shown in Table 2. The filtering process we used was too inefficient to complete the cases $n \geq 10$.

4 Refining semi-transitivity

The notion of k-word-representability refines that of word-representability. However, Theorem 3 shows that these notions are equivalent. Still, k-word-representability plays a very important role in the theory of word-representable graphs.

Thinking along similar lines, we introduce the potentially useful notion of a *k-semi-transitive orientation* refining semi-transitive orientations linked to word-representability via

```
language ESSENCE' 1.0
given n : int
given triangle_table : matrix indexed by [int(1..numgraphs),int(1..(n-1)*(n)/2)]
                       of int(0,1)
letting LETTER be domain int(1..n)

find upper_triangle : matrix indexed by [int(1..((n-1)*n/2))] of int(0,1)
find u : matrix indexed by [LETTER, LETTER] of int(0,1) $ graph undirected edges
find e : matrix indexed by [LETTER, LETTER] of int(0,1) $ graph directed edges
find t : matrix indexed by [LETTER, LETTER] of int(0,1) $ transitive closure
branching on [u]

such that
    $ the diagonal is empty
    forAll i : LETTER . u[i,i] = 0,
    $ the graph is undirected
    forAll i,j : LETTER . u[i,j] = u[j,i],

    $ linking u and the upper triangle
    forAll i,j : LETTER . i < j -> (u[i,j] = upper_triangle[n*(i-1)+j-((i+1)*i/2)]),
    $ the graph is one of the preprocessed graphs
    table(upper_triangle,triangle_table),

    $ linking e and u
    forAll i,j : LETTER . u[i,j] = 0 -> e[i,j]=0,
    forAll i,j : LETTER . u[i,j] = 1 -> ((e[i,j]=1) \/ e[j,i]=1),

    $ directed graph is irreflexive and antisymmetric
    forAll i : LETTER . e[i,i] = 0,
    forAll i,j : LETTER . i < j -> ( (e[i,j] = 0) \/ (e[j,i] = 0)),

    $ t is transitive closure of e and is acyclic
    forAll i,j : LETTER . (e[i,j] = 1) -> (t[i,j] = 1),
    forAll i,j,k : LETTER . ( (t[i,j] = 1) /\ (t[j,k] = 1)) -> (t[i,k] = 1),
    forAll i : LETTER . t[i,i] = 0,

    $ semi transitive ordering
    forAll i,k: LETTER .
        ((t[i,k] = 1) /\ (u[i,k] = 1)) ->
            ((ordering[i,k] = 1) /\
              forAll j : LETTER .
                   ((t[i,j]=1 /\ t[j,k]=1) ->  (e[i,j] = 1 /\ e[j,k]=1)))
```

Figure 6: Essence Prime model of word-representable graphs

Theorem 1. Recall the definition of a shortcut in Section 1.1. An undirected graph is k-semi-transitively oriented, or *k-semi-transitive* for brevity, if it admits an acyclic orientation avoiding shortcuts of length k (longer shortcuts are allowed). In particular, an undirected graph is 3-semi-transitive if it admits an acyclic orientation such that for any directed path $v_0 \to v_1 \to v_2 \to v_3$ of length 3 for which $v_0 \to v_3$ is an edge, also $v_0 \to v_2$ and $v_1 \to v_3$ are edges.

The notion of 3-semi-transitivity is easily expressed in SMT. Writing u_{ij} for the boolean expressing whether there is an undirected edge from i to j, and e_{ij} for the boolean expressing whether there is a directed edge from i to j, the connection between directed and undirected graph is expressed by

$$u_{ij} \Leftrightarrow (e_{ij} \vee e_{ji})$$

for all vertices i, j. Being acyclic is expressed by the existence of a weight function w such that

$$e_{ij} \Rightarrow w(i) > w(j)$$

for all vertices i, j. Finally, the path condition is expressed by

$$((\exists k, m : ((e_{ik} \wedge e_{jk}) \vee (e_{ki} \wedge e_{kj})) \wedge e_{im} \wedge e_{mj}) \;\Rightarrow\; e_{ij}$$

for all vertices i, j, where $\exists$ runs over the vertices. For a given undirected graph, we take the conjunction of the above requirements and for all i, j we add $\wedge u_{ij}$ if there is an edge from i to j, and add $\wedge \neg u_{ij}$ otherwise. Then, by construction, the resulting formula is satisfiable if and only if the undirected graph is 3-semi-transitive. We built these formulas for all connected graphs on ≤ 9 vertices, and applied Z3 on them. As a result, we determined that for ≤ 8 vertices a graph is 3-semi-transitive if and only if it is word-representable. In contrast, for 9 vertices we determined that there are exactly 4 graphs that are 3-semi-transitive but not word-representable, and hence not semi-transitive. They are depicted in Figure 7. An SMT encoding of checking semi-transitivity is also included in the tool linked to in Section 2.

Using a similar encoding of the problem, these computational results were extended to finding the number of all 3-semi-transitively orientable connected graphs on up to 10 vertices using the constraint programming methods described in Section 3. We refer to Table 3 where these results are recorded along with the number of minimal (not containing smaller such graphs as induced subgraphs) non-3-semi-transitively orientable connected graphs. Comparing Tables 2 and 3, we see that there are 585 3-semi-transitively orientable, but not semi-transitively orientable connected graphs on 10 vertices.

Thus, the notions of k-semi-transitively orientable graphs and semi-transitively orientable graphs are not equivalent.

5 Concluding remarks

We conclude by suggesting a few directions of further research relevant to our paper. In each of these directions one can use the computational approaches/tools developed by us to

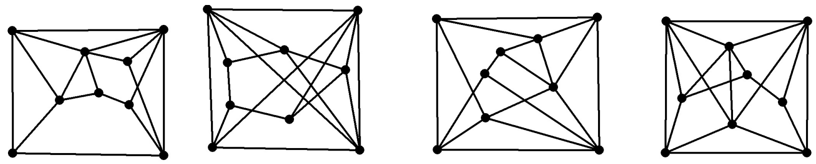


Figure 7: 3-semi-transitively, but not semi-transitively orientable graphs

# of vert.	# of conn. graphs	All non-3-semi-transitively orientable graphs				
		Total	% of cand.	Time	Minimal	Non-Minimal
6	112	1	0.89%	4.0s	1	0
7	853	25	2.93%	6.0s	10	15
8	11,117	929	8.36%	80s	47	882
9	261,080	54,953	21.05%	2.8h	175	54,778
10	11,716,571	4,879,508	41.65%	22d	-	-

Table 3: Numbers of (minimal) non-3-semi-transitively orientable connected graphs (the sequence A319492 in [28]) and the CPU time to obtain them. The time to count minimal/non-minimal graphs is not shown.

support finding new results. In particular, one could try to use our tools to take all bipartite graphs and to test Conjecture 5 for larger graphs.

It would be interesting to extend the construction of J_4 in Figure 5 (in a natural way) to more than 9 vertices so that new graphs with high representation numbers would be obtained. This may help to prove or disprove Conjecture 7.

Also, an intriguing question is whether or not there exists k such that semi-transitive orientability is equivalent to k-semi-transitively orientability. If such a k exists, it must be > 3 (e.g. because of the graphs in Figure 7). In either case, to study the properties of k-semi-transitively orientable graphs (at least 3-semi-transitively orientable graphs) is an interesting and challenging direction of research. Many questions that can be asked about word-representable graphs [19, 20] can be asked about k-semi-transitively orientable graphs, e.g. how many such graphs there are, or how we can describe these graphs in terms of forbidden subgraphs, etc, etc.

Finally, even though it seems that our current methods would not be able to extend the results of Table 2 to 12 vertices, it is interesting if it would be ever possible to achieve.

References

[1] P. Akrobotu, S. Kitaev, and Z. Masárová, On word-representability of polyomino triangulations, *Siberian Adv. Math.* **25** (2015), 1–10.

[2] C. Bessiere, Constraint propagation. In F. Rossi, P. van Beek, and T. Walsh, eds., *Handbook of Constraint Programming*, Elsevier Science, 2006, pp. 29–83.

[3] T. Z. Q. Chen, S. Kitaev, and B. Y. Sun, Word-representability of face subdivisions of triangular grid graphs, *Graphs and Combin.* **32(5)** (2016), 1749–1761.

[4] T. Z. Q. Chen, S. Kitaev, and B. Y. Sun, Word-representability of triangulations of grid-covered cylinder graphs, *Discr. Appl. Math.* **213** (2016), 60–70.

[5] A. Collins, S. Kitaev, and V. Lozin, New results on word-representable graphs, *Discr. Appl. Math.* **216** (2017), 136–141.

[6] A. Distler, C. Jefferson, T. Kelsey, and L. Kotthoff, The semigroups of order 10. In *Proceedings of the Principles and Practice of Constraint Programming — 18th International Conference, CP 2012*, Lect. Notes in Comp. Sci., Vol. 7514, Springer, 2012, pp. 883–899.

[7] A. Gao, S. Kitaev, and P. Zhang, On 132-representable graphs. *Australasian J. Combin.* **69** (2017), 105–118.

[8] I. P. Gent, C. Jefferson, and I. Miguel, Minion: A fast scalable constraint solver. In *Proceedings of the 2006 conference on ECAI 2006: 17th European Conference on Artificial Intelligence*, 2006, pp. 98–102.

[9] I. Gent, S. Kitaev, A. Konovalov, S. Linton, and P. Nightingale, S-crucial and bicrucial permutations with respect to squares, *J. Integer Sequences* **18** (2015), Article 15.6.5.

[10] M. Glen, Colourability and word-representability of near-triangulations, preprint, 2016. Available at `https://arxiv.org/abs/1605.01688` .

[11] M. Glen, Software to deal with word-representable graphs, preprint, 2017. Available at `https://personal.cis.strath.ac.uk/sergey.kitaev/word-representable-graphs.html` .

[12] M. Glen and S. Kitaev, Word-representability of triangulations of rectangular polyomino with a single domino tile, *J. Combin. Math. Combin. Comput.* **100** (2017), 131–144.

[13] M. M. Halldórsson, S. Kitaev, and A. Pyatkin, Graphs capturing alternations in words. In Y. Gao, H. Lu, S. Seki, and S. Yu, eds., *Developments in Language Theory. DLT 2010*, Lect. Notes in Comp. Sci., Vol. 6224, Springer, pp. 436–437.

[14] M. M. Halldórsson, S. Kitaev, and A. Pyatkin, Alternation graphs. In P. Kolman and J. Kratochvíl, eds., *Graph-Theoretic Concepts in Computer Science. WG 2011*, Lect. Notes in Comp. Sci., Vol. 6986, Springer, 2011, pp. 191–202.

[15] M. Halldórsson, S. Kitaev and A. Pyatkin, Semi-transitive orientations and word-representable graphs, *Discr. Appl. Math.* **201** (2016), 164–171.

[16] T. Hiraguchi, On the dimension of partially ordered sets, *Sci. Rep. Kanazawa Univ.* **1** (1951), 77–94.

[17] S. Huczynska, P. McKay, I. Miguel, and P. Nightingale, Modelling equidistant frequency permutation arrays: an application of constraints to mathematics. In *Proceedings of the Principles and Practice of Constraint Programming — 15th International Conference*, Lect. Notes in Comp. Sci., Vol. 5732, Springer, 2009, pp. 50–64.

[18] S. Kitaev, On graphs with representation number 3, *J. Autom. Lang. Combin.* **18** (2013), 97–112.

[19] S. Kitaev, A comprehensive introduction to the theory of word-representable graphs. In É. Charlier, J. Leroy, and M. Rigo, eds., *Developments in Language Theory. DLT 2017*, Lect. Notes in Comp. Sci., Vol. 10396, Springer, 2017, pp. 36–67.

[20] S. Kitaev and V. Lozin, *Words and Graphs*, Springer, 2015.

[21] S. Kitaev and A. Pyatkin, On representable graphs, *J. Autom. Lang. Combin.* **13** (2008), 45–54.

[22] S. Kitaev, P. Salimov, C. Severs, and H. Úlfarsson, On the representability of line graphs. In G. Mauri and A. Leporati, eds., *Developments in Language Theory. DLT 2011*, Lect. Notes in Comp. Sci., Vol. 6795, Springer, 2001, pp. 478–479.

[23] S. Kitaev and S. Seif, Word problem of the Perkins semigroup via directed acyclic graphs, *Order* **25** (2008), 177–194.

[24] B. D. McKay and Adolfo Piperno, Practical graph isomorphism, II, *J. Symb. Comput.* **60** (2014), 94–112.

[25] Y. Mandelshtam, On graphs representable by pattern-avoiding words, preprint, 2016. Available at https://arxiv.org/abs/1608.07614.

[26] P. Nightingale, Ö. Akgün, I. P. Gent, C. Jefferson, I. Miguel, and P. Spracklen, Automatically improving constraint models in Savile Row, *Artificial Intelligence* **251** (2017), 35–61.

[27] F. Rossi, P. Van Beek, and T. Walsh, *Handbook of Constraint Programming*, Elsevier, 2006.

[28] N. J. A. Sloane, The Online Encyclopedia of Integer Sequences, published electronically at https://oeis.org.

[29] The SMT solver Yices, 2018. Available at http://yices.csl.sri.com/.

[30] The SMT solver Z3, 2018. Available at https://github.com/Z3Prover/z3.

2010 *Mathematics Subject Classification*: Primary 05C62; Secondary 68R15.

Keywords: word-representable graph, representation number, enumeration, semi-transitive orientation, k-semi-transitive orientation.

(Concerned with sequences A156808, A290814, A319489, A319490, A319491, and A319492.)

Received October 1 2018; revised versions received February 22 2019; February 23 2019. Published in *Journal of Integer Sequences*, February 24 2019.

Return to Journal of Integer Sequences home page.