A Term Rewriting System for Kuratowski's Closure-Complement Problem*

Osama Al-Hassani[1], Quratul-ain Mahesar[1], Claudio Sacerdoti Coen[2], and Volker Sorge[1]

1 **School of Computer Science, University of Birmingham**
Edgbaston, United Kingdom
O.Al-Hassani@cs.bham.ac.uk,Q.Mahesar@cs.bham.ac.uk,V.Sorge@cs.bham.ac.uk

2 **Dipartimento di Scienze dell'Informazione, Università di Bologna**
Mura Anteo Zamboni, 7 (BO) Italy
sacerdot@cs.unibo.it

Abstract

We present a term rewriting system to solve a class of open problems that are generalisations of Kuratowski's closure-complement theorem. The problems are concerned with finding the number of distinct sets that can be obtained by applying combinations of axiomatically defined set operators. While the original problem considers only closure and complement of a topological space as operators, it can be generalised by adding operators and varying axiomatisation. We model these axioms as rewrite rules and construct a rewriting system that allows us to close some so far open variants of Kuratowski's problem by analysing several million inference steps on a typical personal computer.

1998 ACM Subject Classification G.2.1 Combinatorics

Keywords and phrases Kuratowski's closure-complement problem, Rewriting system

Digital Object Identifier 10.4230/LIPIcs.RTA.2012.38

Category Regular Research Paper

1 Introduction

In 1922 Kuratowski asked and solved the following question on an arbitrary topological space: how many different combinations of the operators of complement and closure exist? The number turns out to be just 14 and the proof is quite small. The problem has been generalised in many different ways to consider other operators, such as union or intersection, or slightly different settings, such as point free topology (locale theory). The solution to a generalised version can be a significantly larger number of combinations, but it could also be a proof that infinitely many combinations exist.

Computing finite large solutions, or obtaining an intuition for infinite variants is unfeasible by hand and therefore computer automation is crucial to our solutions of the problems. Solutions or partial solutions can be represented as directed graphs whose vertices are equivalence classes of provably equal combinations of operators and whose arcs represent the order relation. Graphs can be constructed iteratively by systematically adding more combinations of iterators, merging distinct vertices once it can be shown that they contain

* This work was partially supported by the Italian PRIN project McTafi (Metodi Costruttivi in Topologia, Algebra e Fondamenti dell'Informatica.)

23rd International Conference on Rewriting Techniques and Applications (RTA'12).
Editor: A. Tiwari; pp. 38–52

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$\frac{}{x \leq x}(\texttt{reflexive}) \qquad \frac{x \leq y \quad y \leq z}{x \leq z}(\texttt{transitive}) \qquad \frac{x \leq y \quad y \leq x}{x = y}(\texttt{antisymmetric})$$

$$\frac{x \leq y}{-y \leq -x}(\texttt{antimonotone}) \qquad \frac{}{x \leq - - x}(\texttt{saturates}) \qquad \frac{}{- - -x = -x}(\texttt{quasi-idempotent})$$

$$\frac{}{i(x) \leq x}(\texttt{reduces}) \qquad \frac{x \leq y}{i(x) \leq i(y)}(\texttt{monotone}) \qquad \frac{}{i(x) = i(i(x))}(\texttt{idempotent})$$

$$\frac{}{x \leq c(x)}(\texttt{saturates}) \qquad \frac{x \leq y}{c(x) \leq c(y)}(\texttt{monotone}) \qquad \frac{}{c(x) = c(c(x))}(\texttt{idempotent})$$

$$\frac{}{c(-x) \leq -i(x)}(\texttt{compatible-1}) \qquad \frac{}{i(-x) \leq -c(x)}(\texttt{compatible-2})$$

Figure 1 The inference rules that define the problem.

provably equal combinations. For problems with finite solutions, this process terminates once a fixpoint is reached, in the sense that new vertices are not added, or vertices will never be merged in the future. The resulting graph then represents all possible distinct combinations. In a case where a solution is infinite, there exist infinitely many distinct combinations and finite graphs can only ever represent an approximation of the solution. Nevertheless these approximations can offer crucial information by exhibiting regularities in the growth of the graph, to suggest the existence of an infinite subgraph of distinct vertices that will never become equal, thereby leading to the essential proof idea.

We present a generalised Kuratowski problem (§2) — originally proposed by Sambin in [7], Section 2.4, and still listed as an open problem in [8] — which is particularly demanding in terms of size of the approximating graphs. Indeed, in order to exhibit sufficient evidence of regularities in the graph we need to compute several million edges, i.e., we need to prove several million lemmas on relations between pairs of operator combinations. Since traditional automated theorem proving techniques are unsuitable for this task, we have developed a term rewriting system that models all inference rules of the problem in a uniform way and, coupled with a particular strategy and some standard graph algorithms, can show the large numbers of necessary lemmas in a few minutes (§3). We then demonstrate how the rewriting system can be further enhanced, whilst retaining both correctness and completeness by exploiting regularities in the partial solutions of the problem (§4). The implementation of our ideas have not only enabled us to prove the infinite nature of the generalised Kuratowski problem, which was up to now unknown (§6), but also serves as a basis to tackle other variants of Kuratowski's problem (§5).

2 The Problem

Kuratowski's classical closure-complement problem [5] can be solved by observing that the following identities hold for the interior operator i, the closure operator c, and the complement '$-$' for subsets x of an arbitrary topological space. (i) $c(c(x)) = c(x)$, (ii) $--x = x$, (iii) $i(x) = -c(-x)$, and (iv) $c(i(c(i(x)))) = c(i(x))$. Applying the previous equalities only, one can show that there exist at most 14 distinct subsets one can obtain by different combinations of the three operators. As a consequence of equation (iii), all combinations can be expressed in terms of closure and complement, only. Furthermore, one can indeed construct a model, that is, a space in which this upper bound can be achieved.

The observation that the previous identities are sufficient to solve the problem allows to

restate Kuratowski's classical problem in point-free topology. It is sufficient to forget the concrete definition of the operators and to define them axiomatically by means of the four identities above. The domain of the operators is thus no longer required to be the power set of a topological space, but it can be any arbitrary set.

In the rest of the paper we are interested in the generalisation of the point-free version of Kuratowski's problem introduced by Sambin in [7], Section 2.4. Corsi's master thesis [3] studied the problem under the supervision of Sambin, but the problem remained open and only minimal progress towards a solution was achieved. The generalisation is obtained by introducing a partial order relation $\leq$ — that captures the inclusion relation for subsets — and relaxing the axioms for the operators as given in Figure 1 in a rule format. Observe that the relaxed axiomatisation, effectively turns i into a reduction operator, c into a saturation operator, and $-$ into a pseudo-complement. Observe also that the two compatibility requirements are reminiscent of the classical equation $i(x) = -c(-x)$ (dually $c(x) = -i(-x)$). Indeed, if we define $i(x)$ as $-c(-x)$ and we also assume $--x = x$ for all x, then both compatibility axioms can be derived.

An example model for the axioms can be obtained by combining the definitions of interior, closure and complement with the rules of intuitionistic logic. Given a topological space $(P, \mathcal{O})$, the interior of a set x is defined as $\{\alpha \mid \exists y \in \mathcal{O},\ \alpha \in y \wedge \forall \beta \in y.\beta \in x\}$, and a definition of the closure of x that avoids any reference to negation is $\{\alpha \mid \forall y \in \mathcal{O},\ \alpha \in y \Rightarrow \exists \beta \in y.\beta \in x\}$ (the set of all accumulation points of x). Thus the notion of interior hides an $\exists\forall$ combination and that of closure a $\forall\exists$. The complement of x is $\{\alpha \mid \neg(\alpha \in x)\}$ and it hides a negation. Finally, the subset relation (axiomatised as $\leq$) hides implication: $x \subseteq y$ iff $\forall \alpha,\ \alpha \in x \Rightarrow \alpha \in y$. All the axioms presented are thus simply obtained from the properties of negation and the quantifiers in intuitionistic logic. For instance, from the intuitionistic principle $A \Rightarrow \neg\neg A$ we obtain $x \leq --x$ and from the DeMorgan laws for quantifiers we obtain the two compatibility relations: For example, $\forall\exists\neg \Rightarrow \neg\exists\forall$ becomes $c(-x) \leq -i(x)$.

Since we are effectively interested in the number of different combinations of operators that can lead to distinct sets when applied to any subset of a topological space, we define the generalised Kuratowski problem in terms of equivalent operator combinations.

▶ **Definition 1** (Generalised Kuratowski closure-complement problem). Let $(P, \leq)$ be any partially ordered set and let $\{i, c, -\}$ be the set of operators on P axiomatised as in Figure 1. Let $S = \{i, c, -\}^*$ be the set of all words over the operators (i.e., all possible finite combinations). We define the order relation $\leq$ on S for all $w_1, w_2 \in S$ by: $w_1 \leq w_2$ iff $w_1(x) \leq w_2(x)$ for all $x \in P$. Finally, let $\equiv$ over S be the symmetric closure of $\leq$. The generalised Kuratowski closure-complement problem then consists in computing the cardinality of $S_{/\equiv}$, the set of equivalence classes of S modulo $\equiv$.

Useful for the remainder of our considerations is to define the canonical representative of an equivalence class $[w]_{/\equiv} \in S_{/\equiv}$ as the minimum element of the set according to the shortlex order[1]. Furthermore, we can naturally extend the relation $\leq$ on S to equivalence classes.

Since the cardinality of $S_{/\equiv}$ is not necessarily finite, for practical purposes it is necessary to define finite approximations to the solution.

▶ **Definition 2** (n^{th} approximation). Let $S_n = \{i, c, -\}^{\leq n} \subset S$ be the set of all operator combinations up to order n. For $w_1, w_2 \in S_n$ we define $\leq_n$ as $w_1 \leq_n w_2$ iff for all $x \in P$ we

[1] Two words are in the shortlex order relation when the first is shorter or, in case they have the same length, when the first comes first in lexicographical order.

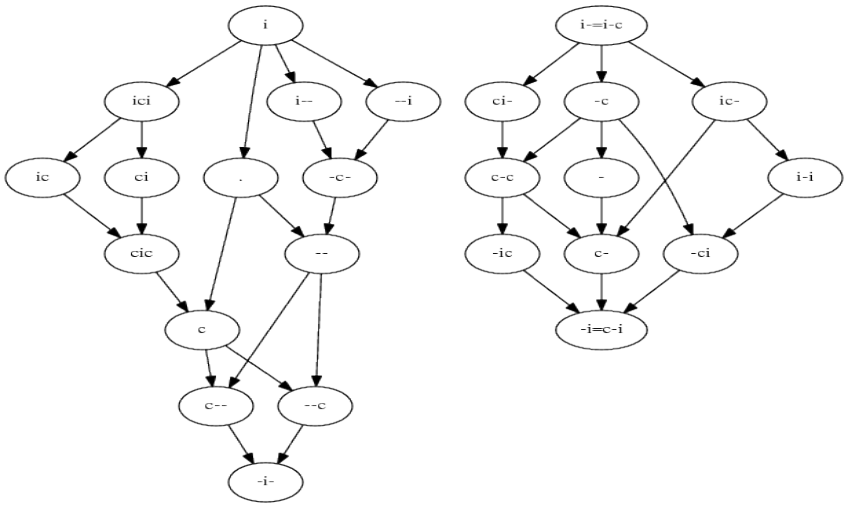

Figure 2 Approximating graph of order 3 for the generalised problem.

can derive $w_1(x) \leq w_2(x)$ by applying the axioms from Figure 1 to elements $w \in S_n$ only (i.e., we restrict derivations to combinations of maximally n operators).

Finally, let $\equiv_n$ the symmetric closure of $\leq_n$. Then the n^{th} approximation of the generalised Kuratowski closure-complement problem is defined as computing the cardinality of $S_{n/\equiv_n}$.

The n^{th} approximation of the problem can be visually represented as a directed graph whose vertices are the equivalence classes of $S_{n/\equiv_n}$ and whose edges represent one step of the $\leq_n$ relation.

▶ **Definition 3** (Approximating graph of order n)**.** Let $G = (V, A)$ be a directed graph, where we define the set of vertices $V = S_n$ and the set of arcs A by $(v_1, v_2) \in A$ iff $v_1 \leq_n v_2$ for $v_1, v_2 \in V$.

Now let V' be the set of all strongly connected components in G. We then define the *approximating graph of order* n as $G' = (V', A')$ where $(v'_1, v'_2) \in A'$ iff $v'_1 \leq_n v'_2$ for $v'_1, v'_2 \in V'$.

Note that the approximating graph can be represented in transitively reduced form, exploiting the transitivity of the $\leq_n$ relation. We also observe that every vertex in the approximating graph contains all the elements of the equivalence class it represents. Thus the graph itself provides a solution to the n^{th} approximation problem as the number of vertices in the graph is the cardinality of $S_{n/\equiv_n}$. Consequently, our goal is effectively to construct the graph by partitioning S_n into equivalence classes, which amounts to an inference procedure that determines if $[w_1]_{/\equiv} \leq_n [w_2]_{/\equiv}$ for $[w_1]_{/\equiv_n}, [w_2]_{/\equiv_n} \in S_n/\equiv_n$.

Figure 2 shows the approximating graph of order 3 for the generalised Kuratowski closure-complement problem. The vertices are subsets of S_3, where only three vertices represent equivalence classes with more than one element. Note that '.' corresponds to the empty word ϵ.

We note that the n^{th} approximation is an approximation to the original problem in two ways. First of all it only shows classes whose canonical representative has length at most n. More importantly, however, it does not grant that two distinct classes in the n^{th} approximation will remain distinct for every $(n+m)^{\text{th}}$ approximation. Thus the cardinality of the graph may decrease or increase when moving to larger values of n. Nevertheless, the

approximation procedure is monotone in the following sense: if two words belong to the same class in the n^{th} approximation, they will belong to the same class in any $(n{+}m)^{\text{th}}$ approximation. Moving to the $(n{+}1)^{\text{th}}$ approximation can only collapse more classes or create new ones made only of words of length $n{+}1$.

The following theorem holds.

▶ **Theorem 4.** *If the solution of the generalised problem is finite, then there exists an n such that every $(n{+}m)^{th}$ approximation is isomorphic (as a directed acyclic graph) to the solution.*

The theorem says that approximations *stabilise*, in the sense that larger approximations only augment the cardinality of the equivalence classes, but they do not collapse any existent distinct classes, nor do they add new arcs to the approximating graph.

The theorem does not provide an effective way to decide if an approximation is (isomorphic to) the solution. We postulate the following conjecture that would provide a simple decision procedure to recognise solutions.

▶ **Conjecture 5.** *There exists an m such that, if for a given n the $n^{th}$and the $(n + m)^{th}$approximations are isomorphic, then they are isomorphic to the solution.*

We have not tried to prove the conjecture yet and the proof does not seem to be simple. In particular, we do not know what is the m for the set of axioms considered. Nevertheless, we can employ an alternative to the conjecture to recognise which approximations are solutions. Let us assume that at a certain point the approximations seem to stabilise, i.e., the $(n{+}1)^{\text{th}}$ approximation is equal to the n^{th} approximation. We can build a *syntactic model* of the solution as follow. We take the set P of all strings w made from $\{i, c, -\}$ such that w is a canonical representative of an equivalence class in the n^{th} approximation. Then we define an $\leq$ relation over P by taking the $\leq_n$ relation. The i, c and $-$ operators are obtained as finite maps that associate to each $w \in P$ the canonical representative of $i \circ w$ (respective $c \circ w$ and $- \circ w$) in the n^{th} approximation. In order to verify if $(P, i, c, -)$ is a model for the problem, we can use a computer algebra system or a theorem prover to verify that all axioms hold. If they do, the n^{th} approximation is isomorphic to the solution of the generalised problem because the model shows that all classes are distinct and moreover the number of classes is maximal because we have only equated combinations that had to be equated because proved to be equal. Otherwise, we start computing larger approximations and we will eventually find an $(n{+}m)^{\text{th}}$ approximation that is not isomorphic to the n^{th} approximation that, a posteriori, was not stable.

A priori, if the conjecture is false it may be that all syntactic models built from approximations that seem to be stable (i.e. $(n{+}1)^{\text{th}}$ isomorphic to n^{th}) turn out to be wrong. However, as we will see in the conclusions, this has not been the case for the different instances of the generalised problem that we considered and that seemed to be stable.

The following theorem, instead, is obvious:

▶ **Theorem 6.** *If the solution of the generalised problem is infinite, then there exists an infinite increasing sequence of approximations with larger and larger cardinalities.*

Our experience shows that in this case a clear pattern emerges, which after some time allows us to predict what new classes will be generated passing from any n^{th} approximation to the $(n{+}1)^{\text{th}}$ approximation. This prediction can then be manually turned into a proof that these new classes will never be collapsed in later approximations and therefore the solution is infinite.

Consequently, in the rest of the paper we will focus on finding a solution to the n^{th} approximation of the problem.

3 The Basic Rewriting System

We now present a method to compute approximating graphs by constructing a rewriting system directly from the axioms given in Figure 1. We present the term rewriting system in a standard way employing terminology used in standard references such as [1]. Nevertheless, the system can also be understood as an instance of generalised equational reasoning as defined in [6], which corresponds more to the actual form in which the system was developed.

A preliminary observation is the fact that in the axiomatisation of the problem we can replace the equality with a $\leq$ in all three idempotency inference rules as the (anti)monotonicity of the respective operator yields the equality automatically. For instance, idempotency of the i operator can be replaced by the axiom $i(x) \leq i(i(x))$ since, by monotonicity, we already have $i(i(x)) \leq i(x)$. Therefore, the only rule that employs an equality remains the antisymmetry rule for $\leq$.

The approximating graph of order n can now be computed in two steps. In the first step we compute the initial directed graph G from Definition 3 whose vertices are all the elements of S_n (words of length at most n) and whose arcs represent all pairs $(w_1, w_2) \in \leq_n$. The anti-symmetric rule of the $\leq$ relation and, more generally, equalities are not used in this step. In the second step we apply a standard connected component algorithm to this graph. Since a connected component is made of all vertices that are mutually reachable, i.e., mutually less or equal, by antisymmetry of $\leq$ they are all equal. The resulting graph is then the approximating graph of order n we are looking for.

Since the second step is completely standard and we can employ implementations out-of-the-box, we will only focus on developing the first step here.

In order to compute the first directed graph $G = (V, A)$ it is sufficient to find all pairs of vertices $(w_1, w_2) \in A$ in the transitive reduction of the graph of $\leq_n$ (recall that following Definition 3 w_1, w_2 are words of length $\leq n$ in S_n), since the connected components algorithm does not distinguish between a transitively reduced and a transitively closed graph. In other words, we are looking for all pairs $(w_1, w_2) \in A$ such that $w_1 \leq_n w_2$ and there is no $w_3 \in V$ such that $w_1 \leq_n w_3$ and $w_3 \leq_n w_2$.

By a close inspection of the rules that have a premise, it is easy to notice that all applications of the transitive rules can be pushed towards the root of the derivation tree. For instance, consider the monotone rule for i and assume (by induction hypothesis) that the derivation of the premise $x \leq y$ is obtained by means of a transitive rule whose premises are $x \leq z$ and $z \leq y$. It is therefore possible to conclude both $i(x) \leq i(z)$ and $i(z) \leq i(y)$ and then, with one final application of transitivity, that $i(x) \leq i(z)$. Thus, since we are interested only in the transitively reduced graph, we can also avoid the use of the transitive and reflexive properties of $\leq$.

The final preliminary observation is that, to compute all pairs (w_1, w_2) in the transitively reduced graph G it is sufficient for every word $w \in S_n$ to compute the two sets $w{\downarrow} = \{w' \mid w' \leq_n w \wedge |w'| \leq |w|\}$ and $w{\uparrow} = \{w' \mid w \leq_n w' \wedge |w'| \leq |w|\}$ where $|.|$ is the length of the two combinations. The final set is just given by

$$\bigcup_{w \in S_n} (\{(w, w') \mid w' \in w{\uparrow}\} \cup \{(w', w) \mid w' \in w{\downarrow}\})$$

In order to compute $w{\downarrow}$ and $w{\uparrow}$ we introduce the non confluent, noetherian term rewriting system presented in Figure 3. The term rewriting system manipulates both active configurations of the form $\langle w_1, w_2, d\rangle$ (where $d \in \{\leq, \geq\}$) and stuck terms w. The intended big step semantics of the rewriting system is the following: an initial term $\langle \epsilon, w, d\rangle \rhd^* w'$ iff

$$\begin{array}{ccc}
\text{(saturates)} & \text{(antimonotone)} & \text{(quasi-idempotent)} \\
\langle w_1, --w_2, \geq\rangle \rhd w_1 w_2 & \langle w_1, -w_2, d\rangle \rhd \langle w_1 -, w_2, d^{-1}\rangle & \langle w_1, ---w_2, \geq\rangle \rhd w_1 - w_2 \\
\text{(reduces)} & \text{(monotone)} & \text{(idempotent)} \\
\langle w_1, iw_2, \leq\rangle \rhd w_1 w_2 & \langle w_1, iw_2, d\rangle \rhd \langle w_1 i, w_2, d\rangle & \langle w_1, iiw_2, \geq\rangle \rhd w_1 iw_2 \\
\text{(saturates)} & \text{(monotone)} & \text{(idempotent)} \\
\langle w_1, cw_2, \geq\rangle \rhd w_1 w_2 & \langle w_1, cw_2, d\rangle \rhd \langle w_1 c, w_2, d\rangle & \langle w_1, ccw_2, \leq\rangle \rhd w_1 cw_2
\end{array}$$

$$\begin{array}{cc}
\text{(compatible-1)} & \text{(compatible-2)} \\
\langle w_1, c-w_2, \leq\rangle \rhd w_1 - iw_2 & \langle w_1, i-w_2, \leq\rangle \rhd w_1 - cw_2
\end{array}$$

Figure 3 The non confluent, noetherian term rewriting system to compute $w{\downarrow}$ and $w{\uparrow}$.

wdw' and $|w'| \leq |w|$. In particular, $w{\downarrow}$ can be computed as $\{w' \mid \langle \epsilon, w, \geq\rangle \rhd^* w'\}$ and $w{\uparrow}$ as $\{w' \mid \langle \epsilon, w, \leq\rangle \rhd^* w'\}$.

The small step semantics of the rewriting system is more technical and it involves generic configurations $\langle w_1, w_2, d\rangle$. The idea is that an initial reduction trace $\langle \epsilon, w, d\rangle \rhd^n \langle w_1, w_2, d'\rangle$ represents a partial derivation of wdw' for some yet unknown w'. Two invariants say that $|w_1| = n$ and $w = w_1 w_2$. The partial derivation built in a top-down manner starts with exactly n monotonicity/anti-monotonicity rules: if $w_1 = o_1 \dots o_n$ where $o_j \in \{-, i, c\}$ then the j-th inference rule in the partial derivation is the monotonicity/anti-monotonicity rule for o_j. Moreover, the hypothesis of the partial derivation is $w_2 d' w_2'$ for some yet unknown w_2' such that $w' = w_1 w_2'$. According to this interpretation, a reduction trace $\langle \epsilon, w, d\rangle \rhd^* \langle w_1, w_2, d'\rangle \rhd w'$ corresponds to a derivation of wdw' where there is a w_2' such that $w' = w_1 w_2'$, the last inference rule in the top-down construction is an axiom that proves $w_2 d' w_2'$ and $|w_2'| \leq |w_2|$. The proof that reduction traces of length n correspond to partial derivation trees of height n having the property just described is by induction on n. We only sketch here one case of the proof.

Each rule in Figure 3 corresponds to the rule with the same name in Figure 1. It means that applying the reduction rule adds the corresponding inference rule to the partial proof tree. The most interesting rule is the rule `antimonotone`: In order to proceed in the derivation we use one more application of antimonotonicity of complement by pushing $-$ on top of the stack w_1 and looking for a new derivation for $w_2 d^{-1} w'$. To see that the rule is correct, assume that $\langle \epsilon, w, d\rangle \rhd^n \langle w_1, -w_2, d'\rangle \rhd \langle w_1 -, w_2, d'^{-1}\rangle$. By induction hypothesis there is a partial proof derivation of wdw' built top-down that starts with monotone/anti-monotone rules for the operators in w_1 and whose hypothesis is $-w_2 d' w''$ for some yet unknown w'' such that $w' = w_1 w''$. By applying anti-monotonicity of $-$ we obtain a new partial proof derivation of wdw' whose new hypothesis is $w_2 d'^{-1} w'''$ and such that $w' = w_1 w'' = w_1 - w'''$. The reduction rule is therefore correct and by applying it we discover that $w'' = -w'''$ or, equivalently, that the next rule in the combination w' after w_1 is $-$.

Strong normalisation of the term rewriting system can simply be proved by induction on the length of the second component of active configurations, which always decreases by one in all (anti)monotonicity rules. All remaining rules produce a stuck term.

By inspection of all the rules, it is easy to prove (by induction on the second component of an active configuration) that if $\langle \epsilon, w, d\rangle \rhd^* w'$ then $|w'| \leq |w|$. Moreover, if $\langle \epsilon, w, d\rangle \rhd^* w'$ and $|w'| = |w|$ then $\langle \epsilon, w', d^{-1}\rangle \not\rhd^* w$. This is important for efficiency reasons since it means that we are never generating the same arc twice (as $w_1 d w_2$ and $w_2 d^{-1} w_1$).

The system clearly has several critical pairs between (anti)monotonicity rules and the

remaining rules. Actually, it turns out that every critical pair is not joinable and the system is thus non confluent. Non-joinability is a feature of our system; because our rewriting rules are never applied under a context,from non-joinability it follows that we never compute the same arc twice in different ways.

Computing all normal forms of a term can be done very efficiently (in terms of actual, non asymptotic computational cost of the program): At every step at most two rules can be applied, one produces a stuck term and the other can be implemented as a tail recursive call. It is thus possible to simplify the code of an implementation for a generic term rewriting system.

4 The Advanced Rewriting System

Given a combination $w \in S_n$, the computation of $w{\downarrow}$ and $w{\uparrow}$ by means of the term rewriting system presented in the previous section is very efficient. Nevertheless, the number of combinations to be reduced is exponential in n and the number of reducts for each w is also exponential in n. The limiting factor for the computation of larger and larger approximating graphs is thus the memory required to hold the graph defined by $w{\downarrow}$ and $w{\uparrow}$, which is the initial directed graph G from Definition 3 before the computation of connected components.

To be able to compute larger approximations, we exploit the following result: There exist only 7 distinct equivalence classes of combinations of closure and interior. While this results is well known in the literature and we can obtain it with our technique for very small values of n, we additionally observed that every class can be associated with a regular expression that generates all elements of the class[2]. These seven regular expressions are:

$$\epsilon, i^+, c^+, (ic)^+, (ci)^+, i(ci)^+, c(ic)^+$$

Taking as canonical representatives the shortest expressions in each class, we have that the set of representatives is $\{\epsilon, i, c, ic, ci, ici, cic\}$. Let K be any regular expression that generates the set. When we consider combinations that also contain complement, and remembering that $---x = -x$, we obtain that all combinations can be partitioned into an infinite number of sets of equivalent combinations whose representatives are all generated by the following regular expression E: $(-|--)?(K--?)^*K?$. The set that corresponds to a representative is the set obtained by replacing any occurrence of $-$ with an odd number of occurrences of $-$ and any occurrence of a term generated by K with an element of its equivalence class. For instance $-----icicicic----$ is a member of the set whose representative is $-ic--$. The sets that correspond to different representatives are not distinct according to the $\equiv$ relation. For instance $c-i-$ and $-i-$ are representatives of different sets, but $c-i- \equiv -i-$. Nevertheless, if two elements belong to the same set, than they are equivalent. Thus the $\equiv$ equivalence relation is more fine grained than the equivalence relation that is induced by partitioning with respect to regular expressions.

The idea to speed up our previous algorithm is to avoid to generate the vertices (and relative arcs) that correspond to non-canonical representatives of the equivalence classes discussed above. These vertices will all belong to the connected component that will be collapsed to its canonical representative. For instance, for $n = 7$, our previous algorithm would handle the vertices $\{-, ---, -----, -------\}$ as potentially distinct.

To implement the idea, we change the already presented algorithms in two ways.

[2] This property does not hold any longer when we consider combinations with complement.

$$\underset{--w \rhd w}{(--)} \qquad \underset{ccw \rhd w}{(\texttt{cc})} \qquad \underset{iiw \rhd w}{(\texttt{ii})} \qquad \underset{ciciw \rhd ciw}{(\texttt{cici})} \qquad \underset{icicw \rhd icw}{(\texttt{icic})}$$

$$\underset{\langle w_1, c{-}iw_2, \leq\rangle \rhd w_1{-}iw_2}{(\texttt{compatible-1 + i-idempotent})} \qquad \underset{\langle w_1, i{-}cw_2, \geq\rangle \rhd w_1 i{-}w_2}{(\texttt{compatible-2 + c-idempotent})}$$

Figure 4 Additional rewriting rules.

1. We change the definition of S_n with the following one. The changes apply everywhere in Section 3, and in particular to Definitions 2 and 3.

$$x \in S_n \text{ iff } x \text{ is generated by the regular expression } E \text{ and } |x| \leq n$$

2. We integrate the rewriting system with the rules of Figure 4 after dropping the rule `quasi-idempotent` and the two `idempotent` rules from the previous rewriting system. The reason why we drop these rules is that their left hand side will never match any active configuration due to restricting the definition of S_n.

Considering the rules in Figure 4, we observe that all rules of the first line simplify a combination. Applied repeatedly they put any combination into their K-normal form. The rules of the second line are obtained by applying Knuth-Bendix completion. Note, however, that our rewriting rules come from a non-symmetric relation ($\leq$) and we have to take care of this during the superposition phase of Knuth-Bendix completion. The names of the new rules are a concatenation of the names of the rules superimposed. The new rules are necessary to keep completeness after having changed the definition of S_n. For instance, because $c{-}ii$ does no longer belong to S_4, we are no longer considering the combinations like $(c{-}ii)\uparrow \ni -i$. The new rewriting rule generated by Knuth-Bendix completion takes care of adding $-i$ to $(c{-}i)\uparrow$ by implicitly performing a step of ii-expansion. Note that, in the original rewriting system, monotonicity of i was only used to perform a step of ii-contraction.

In Figure 4 we only list two rules obtained from the Knuth-Bendix completion because all the others are logically redundant: they allow to derive $w_1 \in w_2{\downarrow}$ when there exists a w_3 such that $w_1 \in w_3{\downarrow}$ and $w_3 \in w_2{\downarrow}$. The redundant rules have been pruned by hand, but it is surely possible to automate the procedure.

The new term rewriting system remains noetherian: all the new rules decrease the length of either the (no longer stuck) combinations or the second component of the active configurations. Of the new rules only those in the first line need to be applied several times in order to obtain the normal form of a term. However, it is easy to show that all critical pairs are joinable. Therefore, by Newman's lemma, the normalisation step implemented by the rules in the first line is confluent, as expected.

▶ **Theorem 7** (Correctness and completeness). *The algorithm based on the advanced rewriting system just described correctly computes the n^{th} approximation of the problem for each n.*

The advanced rewriting system is obtained by rewriting in one step all combinations to their canonical representatives in the equivalence classes identified by the regular expression considered. The same trick can used more aggressively when we build the n^{th}approximation after the $(n-1)^{\text{th}}$. Indeed, we can add to the n^{th}term rewriting system one rewriting rule per combination of length $(n-1)$ that in one step rewrites the combination to its $(n-1)^{\text{th}}$canonical representative.

Since the number of these additional rules is exponential in n, we avoid running the Knuth-Bendix completion, by using the new rules only to normalise terms that are not

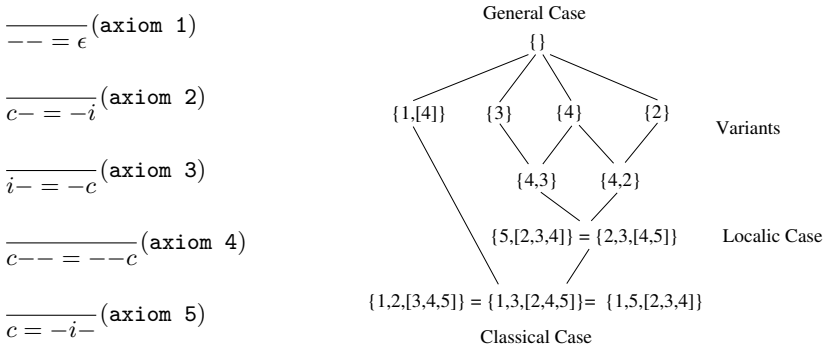

Figure 5 Variations of Kuratowski's problem.

active configurations. The consequence is that we have to normalise exactly the same set of combinations and so we do not save time during the graph generation phase with the rewriting system. The size of the generated graph, however, will be much smaller since it will no longer contain nodes that are not in $(n-1)^{\text{th}}$normal form. The benefit is thus a significant reduction of the computational cost for the computation of the connected components when generating the approximating graph of order n.

The proof of correctness and completeness of the rewriting system obtained with this final improvement is a simple corollary of Theorem 7. The implementation of the improvement is very cheap: the additional rewriting rules generated at the $(n-1)^{\text{th}}$step can only be applied to terms that are stuck according to all other rules. Moreover, they only generate stuck terms. Therefore we can implement this final step as a simple look-up in a trie.

5 More Variations of Kuratowski's Problem

Although the rewriting system presented so far has been developed as a bespoke approach to solve the generalised Kuratowski problem, it turns out that with a parametric implementation our procedure can be applied to a variety of related problems lying between the classical and general problem. These problems are generated by introducing axioms which restrict the general problem, or generalise the classical one. Figure 5 demonstrates variations of Kuratowski's problem, where the axioms on the left hand side gradually refine the generalised problem to the classical problem according to the graph on the right. The nodes are given as sets of included axioms, with the root as the empty set representing the generalised case. Furthermore, axioms derivable from already included ones are given in square brackets.

The variations are motivated by Sambin's work who proposed the generalised problem in the context of intuitionistic point-free topology. Axioms 1–5 are likewise inspired by axioms commonly found in topological problems. For example, axiom 1 postulates the complement operator as idempotent, corresponding to its use in classical logic. Axiom 4, $c-- = --c$, is another axiom that is frequently satisfied by concrete basic topologies (see [8]). Adding axiom 5 to the generalised problem, further restricts the saturation operator c. The axiomatisation obtained is the one for locale theory, for which it is already known in the literature [9] that a maximum of 21 combinations exists. Weaker cases than the localic one can be obtained by effectively splitting axiom 5 into axioms 2 and 3, and considering those either separately or in combination with axiom 4.

All the presented problems in the generalised problem domain can be obtained using our approach, by simply adding the corresponding axioms to our advanced term rewriting

systems as pairs of reductions over active configurations. The Knuth-Bendix completion must also be applied to combine the new rules with the ones of the advanced term rewriting system.

6 Implementation and Results

Our procedure has been implemented in a combination of bespoke code and existing tools. The rewriting engine has been written in pure OCaml and is fully parametric on the list of reduction rules. The connected-components algorithm exploits the `ocamlgraph` library [2] instantiated with an ad-hoc, optimised hashing function for equivalence classes of combinations. For the transitive reduction of the obtained graph we employ the `tred` tool and for its visualisation we use the `dot` tool.

Implementing the rewrite engine from scratch has allowed us to take care of the peculiarities of the rewriting system, (e.g., by exploiting as much as possible tail recursive calls). This choice was also motivated by the need to generate graph representations that were easy to inspect manually as well as to influence the generation of elements in S in order to explore particular subgraphs, which, to the knowledge of the authors would have been difficult to achieve in any existing system. Furthermore, our implementation allowed us to take some care on memory consumption. Nevertheless, when supplying the rewrite engine with the rules of the advanced rewriting system, the program runs out of memory after about 12 minutes on one of the cores of a server equipped with a 2.4GHz Intel Xeon processor and 48GB of RAM producing an approximating graph of order 16.

That means that it explores all words generated by the regular expression in Section 4 of length at most 16, deriving all equations and inequalities that are provable without using combinations of length ≥ 17. The initial graph generated by the rewriting system contains $1,771,825$ vertices, corresponding to all the combinations of length up to $\equiv_{16}$, and $8,687,605$ arcs, corresponding to steps of the $\leq_n$ relation. The approximating graph obtained after the computation of the strongly connected components contains $44,138$ vertices, corresponding to distinct equivalence classes. The number of arcs in the transitively reduced graph could not be computed as the `tred` tool would not terminate within a 2 hour time-limit already for approximating graphs of order greater than 12. Note that `tred` has been run in separate threads of the computations of our rewrite systems.

The resulting graph is quite chaotic, in that, we were not able to find any simple description of either the set of equivalence classes or the elements of most equivalence classes. Nevertheless, by manual inspection of the generated graph we were able to spot sufficient regularity to solve the problem by showing that the number of equivalence classes is infinite. In fact, all the equivalence classes whose representatives are generated by the following regular expression are distinct: $c?(--c)^*(--)?$. Moreover, each one is less than or equal to every other class generated by a longer representative (e.g. $--c \leq --c--$) and they are all bounded by $-i-$, which is also distinct from them and is the minimum of the lattice.

Figure 6 contains a clipping of the approximating graph of order 12 for the generalised problem visualised with the `dot` tool. The clip contains the approximation of the infinite subgraph with elements of the $c?(--c)^*(--)?$, together with some surrounding nodes. The outgoing arc at the bottom leads to the bottom element of the graph, $-i-$, that is not visible. It is obvious to see that the entire subgraph (i) has only one outgoing arc to the bottom element, (ii) it is less than all elements in the remainder of the graph, and (iii) grows downwards with increasing word length. We briefly sketch the formal argument that leads to the above result using our rewriting formalism.

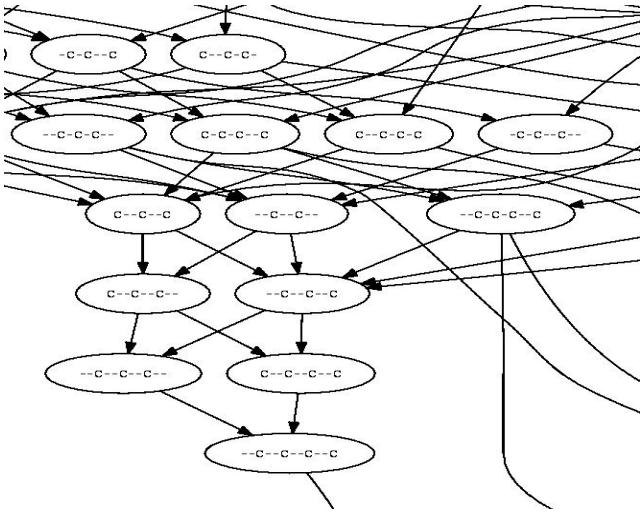

Figure 6 Infinite subgraph for the generalised problem.

First to demonstrate that the equivalence classes generated by the regular expression $r = c?(--c)^*(--)?$ constitute indeed an increasing sequence wrt. $\leq$, we let $\langle w_1, w_2, \geq\rangle$ be any configuration such that $w_2 \neq \epsilon$ and w_1 and w_1w_2 are generated by r.

If $\langle w_1, w_2, \geq\rangle \rhd^* w$ then w is generated by r and is shorter. The argument is by induction over the length $|w_2|$:

1. Suppose w_2 starts with c or with $--$. Then either one of the `saturates` rules is applicable, resulting in a shorter expression.
2. Suppose w_2 starts with c and `monotone` is applicable. Thus w_2 is shorter and we can apply the induction hypothesis.
3. Suppose w_2 starts with $-$ and `antimonotone` is applicable. The new configuration is $\langle w_1-, -w_2', \leq$. The only applicable rule is now `antimonotone` again and we can conclude using the induction hypothesis.

Similarly we can show that $-i-$ is indeed the bottom element: Let $\langle w_1, w_2, \geq\rangle$ be any configuration such that $w_2 \neq \epsilon$ and w_1 and w_1w_2 are generated by r. If $\langle w_1, w_2, \leq\rangle \rhd^* w$ then w is in the same class as $-i-$. Again by induction over $|w_2|$ we can show:

1. Suppose w_2 starts with $c--$ with `compatible-1` we get $w_1-i-w_2 = -i-$.
2. Suppose w_2 starts with c then `monotone` is applicable and we can apply the induction hypothesis.
3. Suppose w_2 starts with $-$ then `antimonotone` is applicable. The new configuration is $\langle w_1-, -w_2', \leq$. The only applicable rule is now again `antimonotone` and we can conclude using the induction hypothesis.

Applying our implementation to other problems in the domain introduced in the previous section, we could quickly verify the results known from the literature of 14 and 21 combinations in the classic and localic case, respectively. For the other problems we obtain a mixed picture of both finite and infinite cases.

Table 1 lists the approximating graphs from order 14 to 16 for the infinite cases in terms of vertices and arcs as well as infinite subgraphs identified. Again no arc count could be computed for the general case due to non-termination of `tred`.

Table 1 Approximating graph for all the variants that do not stabilise.

Axiom set	Order 14		Order 15		Order 16		Infinite
	Classes	Arcs	Classes	Arcs	Classes	Arcs	subgraph
$\emptyset$	10439	?	16869	?	27315	?	$(--c)^*$
$\{2\}$	135	269	142	285	149	299	$(--ci)^*$
$\{3\}$	135	269	142	285	149	299	$(--ic)^*$
$\{4\}$	278	640	283	649	288	660	$(--ici)^*$

So far we have proved formally only the infinite subgraph of the general case to consist of distinct classes. While when adding axiom 2 or 3 or 4 only, the approximating graphs also continue to grow, the infinite subgraph that we spotted in the general case collapses to a finite one as the equation $c-- = --c$ forces all combinations generated by the regular expression $c?(--c)^*(--)?$ into less than four classes. Consequently, the argument we used to show that the general case is infinite does no longer hold. Thus the formal proof for these cases is still outstanding.

For the remaining problem variants the approximating graphs stabilise. The exact figures for the graphs are given in Table 2. Axiom sets $\{1,2,3\}$ and $\{2,3\}$ are the classical and localic case from the literature and we can observe that in our system their approximating graphs stabilise after only few iterations. Similarly, for axiom combinations $\{2,4\}$ and $\{3,4\}$, the set of equivalence classes stabilises quickly at 35 and 44 after 8 and 9 iterations, respectively. Finally when adding axiom 1 alone we get a stable approximation after 13 iterations with 126 classes. For these latter three cases we also do not yet have a formal proof because we did not check yet the induced syntactic model.

7 Conclusions and future work

We have presented the study of the generalised version of Kuratowski's classical closure-complement problem from point-set topology. To solve the problem we used a computational procedure that combines a term rewriting system with bespoke graph algorithms. The procedure is capable of showing several million lemmas about relations between combinations of operators, which allowed us to iteratively approximate the solution to the problem. The resulting graph exhibited enough regularity to enable us to show that the solution space of the problem is infinite, thereby successfully closing the problem. A posteriori, the proof that the number of combinations is infinite was quite easy and the infinite set of distinct combinations is generated by a simple regular expression. Nevertheless, the problem has remained open for more than nine years, and the clutter in the rest of the graph made it difficult to spot the infinite subgraph.

Table 2 Approximating graphs for all variants that stabilise.

Axiom set	Classes	Arcs	Stabilising Iteration
$\{1\}$	126	268	13
$\{1,2,3\}$	14	16	4
$\{2,3\}$	21	31	5
$\{2,4\}$	35	57	8
$\{3,4\}$	35	57	8

From the mathematical point of view, the result is quite interesting. It shows that the generalisation to a saturation operator partially independent from the reduction operator greatly adds to the expressive power of the system. This is one of the main intuitions at the base of the Basic Picture of Sambin [8], a complete re-formulation of point-wise and point-free topology that is deeply rooted in intuitionistic and predicative logic.

Our graph algorithms, mainly the connected components computation, take care of the transitivity of the less than relation and computes the smallest representatives of the equivalence classes according to the shortlex order. The noetherian term rewriting system where all critical pairs are non joinable takes care of monotonicity and anti-monotonicity of the operators and is used to quickly derive a large number of inferences that do not require transitivity. Non joinability implies that all inferences have different conclusions. Finally, another noetherian and confluent rewriting system allows to aggressively prune the number of inferences by constraining the shape of the terms under consideration. When combined together, we need to apply Knuth-Bendix completion between some rules of the first rewriting system and some of the second one because we implicitly choose the strategy when a term is first simplified using the second set of rules, then reduced using the first, then simplified again. The strategy is implicit since we start from terms already in normal form according to the second set of rules. The completion allows to preserve completeness under the implicit strategy.

Our initial problem that generalises Kuratowski's classical result is not the only possible generalisation. Other typical examples of generalisations are obtained by considering other topological or set theoretical operators like union or intersection. A recent reference for variations on the original problem and applications of the classification to the characterisation of properties of subsets of topological space can be found in [4]. The paper also presents a large bibliography on variants and applications of the problem. It does not cite, however, the generalisation and variants studied in this paper, nor the paper [9] that shows that in locale theory there are exactly 21 different combinations.

We have mapped out a landscape of generalised problems that lie between the finite classical and localic cases, where results were previously known, and the infinite generalised case, which is a new result. We are currently systematising the results about the intermediate cases considered. The general picture obtained, once perfectly clear, will be the subject of a future publication in a mathematical venue.

Already the presented results demonstrate that our approach scales well to all these generalisations. Indeed, the system we implemented is fully parametric on the list of reduction rules of the advanced rewriting system. The computation of the Knuth-Bendix like completion, that at the moment is done manually, could be easily automated. Note, however, that we cannot use any of the many off-the-shelf tools available. The reason is that we need to perform a Knuth-Bendix completion, but we are not dealing with a canonical word problem because the starting relation to be oriented is not symmetric. Rewriting in presence of non symmetric relations has been previously investigated by the third author in [6] to implement semi-equational reasoning and the technique employed here can be effectively understood as an application of this study.

Acknowledgements The authors want to thank Prof. Sambin that stated the generalised Kuratowski's closure-complement problem studied in this paper and personally brought it to the attention of the third author.

References

1 Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, Cambridge, 1998.

2 Sylvain Conchon, Jean-Christophe Filliâtre, and Julien Signoles. Designing a generic graph library using ML functors. In *The Ninth Symposium on Trends in Functional Programming*, volume 8, pages 124–140. Intellect, 2008.

3 Lara Corsi. Combinazioni di operatori di interno, chiusura e loro complemento in LJ. Tesi di laurea in Matematica, Università di Padova, 2006.

4 Barry J. Gardner and Marcel Jackson. The Kuratowski closure-complement theorem. *New Zealand Journal of Mathematics*, 38:9–44, 2008.

5 Kazimierz Kuratowski. Sur l'operation a de l'analysis situs. *Fund. Math.*, 3:182–199, 1922.

6 Claudio Sacerdoti Coen. A semi-reflexive tactic for (sub-)equational reasoning. In *Types for Proofs and Programs*, volume 3839/2006 of *LNCS*, pages 98–114. Springer-Verlag, 2006.

7 Giovanni Sambin. Some points in formal topology. *Theoretical Computer Science*, 305(1-3):347–408, 2003.

8 Giovanni Sambin. *The Basic Picture: a structural basis for constructive topology*. Oxford University Press, 2012.

9 He Wei and Zhang Yasoming. Interior and boundary in a locale. *Advances in Mathematics*, 29(5):439–443, 2000.