Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Succinctness and tractability of closure operator representations

Sebastian Rudolph

Technische Universität Dresden, Germany

ARTICLE INFO

Article history:
Received 2 April 2015
Received in revised form 12 December 2015
Accepted 20 December 2015
Available online xxxx

Keywords:
Closure operators
Formal contexts
Implications
Tractability
Succinctness
Computational properties

ABSTRACT

It is widely known that closure operators on finite sets can be represented by sets of implications (also known as inclusion dependencies) as well as by formal contexts. In this article, we consider these two representation types, as well as generalizations of them: extended implication sets and context families. We discuss the mutual succinctness of these four representations and the tractability of certain operations used to compare and modify closure operators.

1. Introduction

Closure operators and closure systems are a basic notion in algebra and occur in various computer science scenarios such as logic programming or databases. One central task when dealing with closure operators algorithmically is to represent them in a succinct way while still allowing for their efficient computational usage. Formal concept analysis (FCA) naturally provides two complementary ways of representing closure operators: by means of *formal contexts* on one side and *implication sets* on the other. Although being complementary, these two representations share the property that they allow for tractable closure computation. In fact, this property is also exhibited by further representation types, which properly generalize the ones mentioned above: *context families* consist of several contexts and the closure is specified as the "simultaneous fixpoint" of all the separate contexts' closures; *extended implications* are implications where auxiliary elements are allowed.

For a given closure operator, the space needed to represent it in one or the other way may differ significantly: it is well known that there are closure operators whose minimal implicational representation is exponentially larger than their minimal contextual one and vice versa (see Section 3).

Thus, when designing algorithms which use and manipulate closure operators (as many FCA algorithms do) it is important to know which of the possible representation types allow for efficient storage and still guarantee fast (that is: PTime) execution of typical computations.

This paper investigates the four representation types in this respect. To this end, we will consolidate known results from diverse areas into one framework and provide some findings which are – to the best of our knowledge – novel and original to fill the remaining gaps. Our main results can be generalized as follows:

E-mail address: sebastian.rudolph@tu-dresden.de.

http://dx.doi.org/10.1016/j.tcs.2015.12.028
0304-3975/

- We show that context families allow for succinct representation of both contexts and implications, and that extended implication sets can succinctly represent all the three other representation types. We also show that a succinct translation (i.e., one where the size of the result is polynomially bounded by the input) in all other directions is not possible.
- We clarify the complexities for comparing closure operators in different representations in terms of whether one is a refinement of the other. Interestingly, some of the investigated comparison tasks are tractable (i.e., time-polynomial), others are not (assuming $\mathrm{P} \neq \mathrm{NP}$). We provide algorithms for the tractable cases and coNP-hardness arguments for the others.
- We go through standard manipulation tasks for closure operators (refinement by adding a closed set, coarsening through an implication, projection, meet and join in the lattice of closure operators) and clarify which are tractable and which are not.

This paper is a significantly refined and extended version of two precursor publications [26,27]. All statements for which proofs are given are original to the best of our knowledge, unless explicitly stated otherwise.

2. Preliminaries

We start providing a condensed overview of the notions used in this paper. After recalling some complexity notations, we introduce closure operators as well as the four representation types we want to discuss in this article: (formal) contexts, context families, implication sets and extended implication sets.

2.1. Complexity notations

In order to asymptotically compare sizes of data structures, we will make use of the Bachmann–Landau notation. In particular, we remind the reader that for two infinite sequences $(a_n)_{n\in\mathbb{N}}$ and $(b_n)_{n\in\mathbb{N}}$, we write

- $a_n \in \Omega(b_n)$ if b_n is an asymptotic lower bound of a_n, i.e., there exists some k with $a_n \geq k \cdot b_n$ for sufficiently large n, and
- $a_n \in \Theta(b_n)$ if b_n is an asymptotic lower and upper bound of a_n, i.e., there exist some k_1 and k_2 with $k_1 \cdot b_n > a_n > k_2 \cdot b_n$ for sufficiently large n.

2.2. Closure operators

We now introduce and formally define the central notion of this paper: closure operators.

Definition 1. Let M be an arbitrary set. A function $\varphi : 2^M \to 2^M$ is called a *closure operator* on M if it is

1. *extensive*, i.e., $A \subseteq \varphi(A)$ for all $A \subseteq M$,
2. *monotone*, i.e., $A \subseteq B$ implies $\varphi(A) \subseteq \varphi(B)$ for all $A, B \subseteq M$, and
3. *idempotent*, i.e., $\varphi(\varphi(A)) = \varphi(A)$ for all $A \subseteq M$.

A set $A \subseteq M$ is called *closed* (or *φ-closed* in case of ambiguity), if $\varphi(A) = A$. The set of all closed sets $\{A \mid A = \varphi(A) \subseteq M\}$ is called *closure system* of φ.

It is easy to show that for an arbitrary closure system $\mathcal{S}$, the corresponding closure operator φ can be reconstructed by

$$\varphi(A) = \bigcap_{B \in \mathcal{S},\ A \subseteq B} B.$$

Hence, there is a one-to-one correspondence between a closure operator and the according closure system.

In the following, we provide some closure operators which will serve as running examples in the course of the paper.

Example 2. Considering $M = \{a, b, c, d, e\}$, the functions α, β, γ, and δ defined in the below table are all closure operators (due to extensivity, every closure operator φ satisfies $A \subseteq \varphi(A)$, thus for better readability, we underline elements of $\varphi(A) \setminus A$).