The Ideal Approach to Computing Closed Subsets in Well-Quasi-orderings

Jean Goubault-Larrecq, Simon Halfon, Prateek Karandikar, K. Narayan Kumar, Philippe Schnoebelen

HAL Id: hal-03083290
https://hal.science/hal-03083290

Submitted on 8 Jun 2021

The Ideal Approach to Computing
Closed Subsets in Well-Quasi-Orderings

J. Goubault-Larrecq[1,2], S. Halfon[1], P. Karandikar[1,3], K. Narayan Kumar[3], and Ph. Schnoebelen[1]

[1] LSV, ENS Cachan, CNRS, France
[2] INRIA Saclay, France
[3] CMI, Chennai, India

Abstract. Elegant and general algorithms for handling upwards-closed and downwards-closed subsets of WQOs can be developed using the filter-based and ideal-based representation for these sets. These algorithms can be built in a generic or parameterized way, in parallel with the way complex WQOs are obtained by combining or modifying simpler WQOs.

Table of Contents

1 Introduction

The theory of well-quasi-orderings (WQOs for short) has proved useful in many areas of mathematics, logic, combinatorics, and computer science. In computer science, it appears prominently in termination proofs [12], in formal languages [11], in graph algorithms (e.g., via the Graph Minor Theorem [45]), in verification (e.g., with well-structured systems [2,19,57]), automated deduction, distributed computing, but also in machine learning [4], program transformation [44], etc. We refer to [36] for "four [main] reasons to be interested in WQO theory".

In computer science, tools from WQO theory were commonly seen as lacking algorithmic contents. This situation is changing. For example, tight complexity bounds for WQO-based algorithms have recently been established and are now used when comparing logics or computational models [29,53,54,55]. As another example, the field of well-structured systems grows not just by the identification of new families of models,

but also by the development of new generic algorithms based on WQO theory, see, e.g., [6,9].

In this chapter we are concerned with the issue of reasoning about, and computing with, downwards-closed and upwards-closed subsets of a WQO. These sets appear in verification (prominently in model-checking of well structured systems [6], in verification of Petri nets [28], in separability problems [26,60], but also as an effective abstraction tool [5,59]). The question of how to handle downwards-closed subsets of WQOs *in a generic way* was first raised by Geeraerts et al.: in [21] the authors postulated the existence of an *adequate domain of limits* satisfying some representation conditions. It turns out that the *ideals* of WQOs always satisfy these conditions, and usually enjoy further algorithmic properties.

Outline of this chapter. We start by recalling, as a motivating example, the algorithmic techniques that have been successfully used to handle upwards-closed and downwards-closed subsets in two different WQOs: the tuples of natural numbers with component-wise ordering, and the set of finite words with subword ordering. We then describe the fundamental structures that underlie these algorithms and propose in Section 3 a generic set of effectiveness assumptions on which the algorithms can be based.

The second part of the chapter, Sections 4 and 5, shows how many examples of WQOs used in applications fulfill the required effectiveness assumptions. Since in practice complex WQOs are most often obtained by composing or modifying simpler WQOs, our strategy for showing their effectiveness involves proving that WQO constructors preserve effectiveness.

A final section discusses our choices —of effectiveness assumptions and of algorithms— and lists some of the first questions raised by our approach.

Genesis of this chapter. This text grew from [25] (unpublished) where Goubault-Larrecq proposed a notion of effective WQOs, and where Theorem 6.2 was first proven. There, Goubault-Larrecq also shows that products, sequence extensions, and tree extensions of effective WQOs are effective. Then, in 2016 and 2017, Karandikar, Narayan Kumar and Schnoebelen developed the framework and handled WQOs obtained by extensions, by quotients, and by substructures. Finally, in 2017 and 2018, Halfon joined the project and contributed most of the results on powersets and multisets. He also studied variant sets of axioms for effective WQOs as reported in Section 6.1. In the meantime, the constructions initiated by [25] have been used in several papers, starting from [16,17], and including [8,9,15,26,40,41,43].

2 Well-quasi-orderings, ideals, and some motivations

A *quasi-ordering* (a QO) $(X, \leq)$ is a set X equipped with a reflexive and transitive relation. We write $x < y$ when $x \leq y$ and $y \not\leq x$, and $x \equiv y$ when $x \leq y$ and $y \leq x$. For $S \subseteq X$, we let $\uparrow S$ and $\downarrow S$ denote the upward and downward closures, respectively, of S in X. Formally, $\uparrow S \stackrel{\text{def}}{=} \{x \in X \mid \exists y \in S : y \leq x\}$ and $\downarrow S \stackrel{\text{def}}{=} \{x \in X \mid \exists y \in S : x \leq y\}$. We will also use $\downarrow_< S$ and $\uparrow_< S$ to collect elements that are strictly above, or below, elements of S, i.e., $\downarrow_< S \stackrel{\text{def}}{=} \{x \in X \mid \exists y \in S : x < y\}$ and similarly for $\uparrow_< S$.

When $S = \{x\}$ is a singleton, we may simply write $\uparrow x$ or $\downarrow_< x$. A subset of X of the form $\uparrow x$ is called a *principal filter* while a subset of the form $\downarrow x$ is a *principal ideal*. A subset $S \subseteq X$ is *upwards-closed* when $S = \uparrow S$, and *downwards-closed* when $S = \downarrow S$. Note that arbitrary unions and intersections of upwards-closed (resp. downwards-closed) sets are upwards-closed (resp. downwards-closed). Observe also that the complement of an upwards-closed set is downwards-closed, and conversely. We write $Up(X)$ for the set of upwards-closed subsets of X, with typical elements $U, U', \ldots$ Similarly, $Down(X)$ denotes the set of its downwards-closed subsets, with typical elements $D, D', \ldots$

2.1 Two motivating examples

Consider the set $X = \mathbb{N}^2$ of pairs of natural numbers. These are the points with integral coordinates in the upper-right quadrant. We order these points with the coordinate-wise ordering, also called *product ordering*:

$$\langle a, b \rangle \leq \langle a', b \rangle \overset{\text{def}}{\Leftrightarrow} a \leq b \wedge a' \leq b' \ .$$

Note that this is only a partial ordering: $\langle 1, 2 \rangle$ and $\langle 3, 0 \rangle$ are incomparable.

2.1.1 $\mathbb{N}^2$ and its upwards-closed subsets. In many applications, we need to consider upwards-closed subsets U, U' , $\ldots$, of $\mathbb{N}^2$. These may be defined by simple, or not so simple, constraints such as U_{ex_1} and V_{ex_1} in Fig. 1.

$$U_{\mathrm{ex}_1}: \quad a > 2 \wedge 2a + b > 10 \qquad\qquad V_{\mathrm{ex}_1}: \quad a^2 + 3b^2 > 100$$
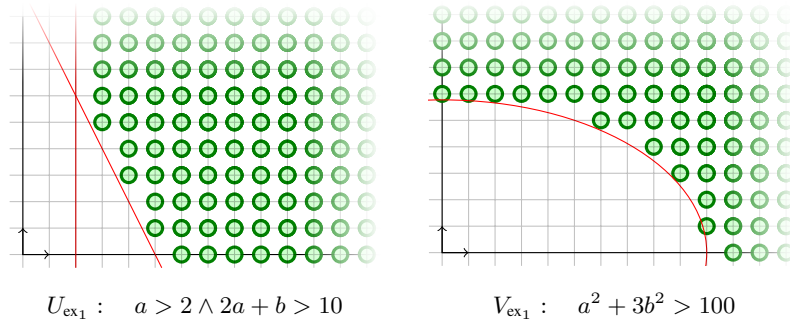
Fig. 1. Two upwards-closed subsets of $\mathbb{N}^2$

A striking aspect of these depictions of U_{ex_1} and V_{ex_1} —see also Fig. 2— is that both can be seen as unions of a few principal filters:

$$U_{\mathrm{ex}_1} = {\uparrow}\langle 3, 5 \rangle \cup {\uparrow}\langle 4, 3 \rangle \cup {\uparrow}\langle 5, 1 \rangle \cup {\uparrow}\langle 6, 0 \rangle \ ,$$
$$V_{\mathrm{ex}_1} = {\uparrow}\langle 0, 6 \rangle \cup {\uparrow}\langle 6, 5 \rangle \cup {\uparrow}\langle 8, 4 \rangle \cup {\uparrow}\langle 9, 3 \rangle \cup {\uparrow}\langle 10, 1 \rangle \cup {\uparrow}\langle 11, 0 \rangle \ .$$

We write $U = \bigcup_{i<n} \uparrow x_i$ to say that the upwards-closed subset U of X is the union of $\uparrow x_0, \ldots, \uparrow x_{n-1}$. The elements x_i are the *generators*, and the finite set $\{x_0, \cdots, x_{n-1}\}$ is a *finite basis* of U. We also say that $\bigcup_{i<n} x_i$ is a *finite basis representation* of U. By removing elements that are not minimal, we obtain a *minimal finite basis* of U.
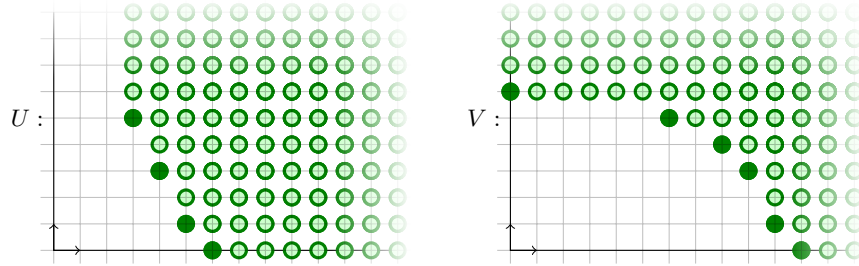
Fig. 2. Finite bases for U_{ex_1} and V_{ex_1}

We shall see later that all upwards-closed subsets of $(\mathbb{N}^2, \leq)$ admit such a representation. For the time being we want to stress how this representation of upwards-closed subsets is convenient *from an algorithmic viewpoint*. To begin with, it provides us with a finite data structure for subsets that are infinite and thus cannot be represented in extension on a computer. Interestingly, some important set-theoretical operations are very easy to perform on this representation: testing whether some point $\langle a, b \rangle$ is in U or V just amounts to comparing $\langle a, b \rangle$ with the points forming the basis of U or V respectively. Testing whether $U \subseteq V$ reduces to checking whether all points in the basis of U belong to V. We see that $U_{\mathrm{ex}_1} \not\subseteq V_{\mathrm{ex}_1}$ since there is a point in U_{ex_1}'s base that is not in V_{ex_1}, i.e., not larger than (or equal to) any of the points in V_{ex_1}'s basis: for instance $\langle 3, 5 \rangle \notin V_{\mathrm{ex}_1}$. Similarly, $\langle 0, 6 \rangle \notin U_{\mathrm{ex}_1}$ hence $V_{\mathrm{ex}_1} \not\subseteq U_{\mathrm{ex}_1}$.

Two further operations that are easily performed are computing $W = U \cup V$ and $W' = U \cap V$ for upwards-closed U and V (recall that such unions and intersections are upwards-closed as observed earlier). For $U \cup V$, we just join the two finite bases and (optionally) remove any element that is not minimal. For example

$$U_{\mathrm{ex}_1} \cup V_{\mathrm{ex}_1} = \big(\uparrow\langle 3, 5 \rangle \cup \uparrow\langle 4, 3 \rangle \cup \uparrow\langle 5, 1 \rangle \cup \uparrow\langle 6, 0 \rangle\big)$$
$$\cup \big(\uparrow\langle 0, 6 \rangle \cup \uparrow\langle 6, 5 \rangle \cup \uparrow\langle 8, 4 \rangle \cup \uparrow\langle 9, 3 \rangle \cup \uparrow\langle 10, 1 \rangle \cup \uparrow\langle 11, 0 \rangle\big)$$
$$= \uparrow\langle 0, 6 \rangle \cup \uparrow\langle 3, 5 \rangle \cup \uparrow\langle 4, 3 \rangle \cup \uparrow\langle 5, 1 \rangle \cup \uparrow\langle 6, 0 \rangle \, .$$

For $U \cap V$, we first observe that principal filters can be intersected with

$$\uparrow\langle a, b \rangle \cap \uparrow\langle a', b' \rangle = \uparrow\langle \max(a, a'), \max(b, b') \rangle \tag{1}$$

and then use the distributivity law $\left(\bigcup_{i<n} \uparrow x_i\right) \cap \left(\bigcup_{j<m} \uparrow y_j\right) = \bigcup_{i,j}(\uparrow x_i \cap \uparrow y_j)$ to handle the general case. This gives, for example,

$$
\begin{aligned}
U_{\mathrm{ex}_1} \cap V_{\mathrm{ex}_1} = & \left[\uparrow\langle 3,5\rangle \cap \uparrow\langle 0,6\rangle\right] \;\cup\; \left[\uparrow\langle 3,5\rangle \cap \uparrow\langle 6,5\rangle\right] \;\cup\; \left[\uparrow\langle 3,5\rangle \cap \uparrow\langle 8,4\rangle\right] \\
& \cup\; \left[\uparrow\langle 4,3\rangle \cap \uparrow\langle 9,3\rangle\right] \;\cup\; \left[\uparrow\langle 5,1\rangle \cap \uparrow\langle 10,1\rangle\right] \;\cup\; \left[\uparrow\langle 6,0\rangle \cap \uparrow\langle 11,0\rangle\right] \\
& \cup\; \cdots \textit{ more filters on elements that are not minimal } \cdots \\
= & \uparrow\langle 3,6\rangle \cup \uparrow\langle 6,5\rangle \cup \uparrow\langle 8,4\rangle \cup \uparrow\langle 9,3\rangle \cup \uparrow\langle 10,1\rangle \cup \uparrow\langle 11,0\rangle \,.
\end{aligned}
$$

$W: \quad U \cup V$ $W': \quad U \cap V$
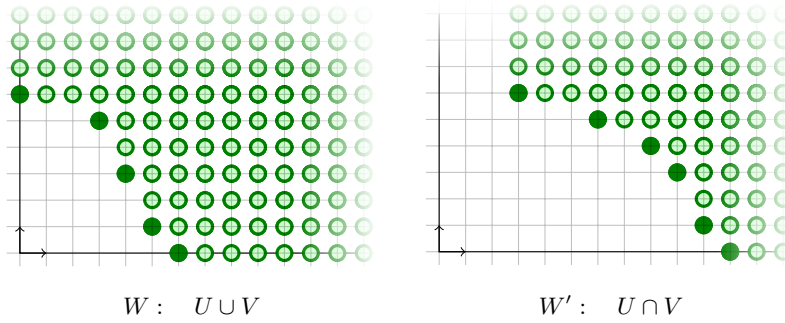
Fig. 3. Computing intersections and unions via finite bases

Finally, a last feature of the finite basis representation for upwards-closed subsets of $\mathbb{N}^2$ is that, if we only consider minimal bases, namely bases of incomparable elements —in essence, if we systematically remove unnecessary generators that are subsumed by smaller generators,— then the representation is *canonical*: there is a unique way of representing any $U \in Up(\mathbb{N}^2)$. Algorithmically, this allows one to implement the required structures using *hash-consing* [14,23], where structures with the same contents are allocated at the same address, with the help of auxiliary hash-tables. In particular, finite *sets* can be implemented this way, efficiently [24]. Equality tests can then be performed in constant time, notably.

2.1.2 Words and their subwords. Our second example comes from formal languages and combinatorics [52]. Let us fix a three-letter alphabet $A = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and write $A^* = \{u, v, \cdots\}$ for the set of all finite words over A. Standardly,

the empty word is denoted by ϵ, concatenation is denoted multiplicatively, and $|u|$ is the length of u. We write $u \preceq v$ when u is a *subword* of v, i.e., a subsequence: u can be obtained from v by erasing some (occurrences of) letters. It is easy to check whether $u \preceq v$ by attempting to construct a leftmost embedding of u into v: this only requires at most one traversal of u and v and takes time linear in $|u| + |v|$. For example, the box to the right shows that $u = \mathtt{abba}$ is not a subword of $v = \mathtt{bacabab}$.
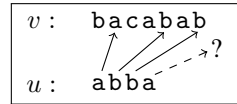
With the subword ordering comes the notion of upwards-closed and downwards-closed languages (i.e., sets of words). For example the language $U_{\text{ex}_2} \subseteq A^*$ of words with at least one $\mathtt{a}$ and at least two $\mathtt{b}$s is upwards-closed, as is V_{ex_2}, the language of words with length at least 2. These upwards-closed languages occur in many applications and one would like to know good data structures and algorithms for manipulating them. It turns out that any such upwards-closed language can be represented as a finite union of principal filters.[4] For example, U_{ex_2} and V_{ex_2} can be written

$$U_{\text{ex}_2} = \uparrow\mathtt{abb} \cup \uparrow\mathtt{bab} \cup \uparrow\mathtt{bba}\,, \quad V_{\text{ex}_2} = \uparrow\mathtt{aa} \cup \uparrow\mathtt{ab} \cup \cdots \cup \uparrow\mathtt{cc} = \bigcup_{|u|=2} \uparrow u\,.$$

In the subword setting, a principal filter is always a regular language. Indeed, for any $u \in A^*$, of the form $u = a_1 a_2 \cdots a_\ell$, one has $\uparrow u = A^* a_1 A^* a_2 A^* \cdots A^* a_\ell A^*$, which is a language at level $\frac{1}{2}$ in the Straubing-Thérien hierarchy [48]. Being simple star-free regular languages, the upwards-closed subsets can be handled with well-known automata-theoretic techniques. However, one can also use the same simple ideas we used for $\mathbb{N}^2$: testing $U \subseteq V$ reduces to comparing the generators, computing unions is trivial, and bases made of incomparable words provide a canonical representation. Finally, computing intersections reduces to intersecting principal filters, exactly as in $\mathbb{N}^2$. For this, we observe that $\uparrow u \cap \uparrow v$ is generated by the minimal words that contain both u and v as subwords. This set of minimal words, written $u \sqcap v$, is called the *infiltration product* of u and v [10]. For example $\mathtt{ab} \sqcap \mathtt{ca} = \{\mathtt{abca}, \mathtt{acba}, \mathtt{acab}, \mathtt{cab}\}$. Infiltrations are a generalization of shuffles and we shall describe a simple algorithm for a generalized infiltration product in Section 4.4.

2.2   What about downwards-closed subsets?

With the previous two examples, we showed how it is natural and easy to work with upwards-closed subsets of a quasi-ordered set when these subsets are represented as a finite union $\bigcup_{i<n} \uparrow x_i$ of principal filters.

Let us now return to our previous setting, $X = \mathbb{N}^2$, and look at the downwards-closed subsets $D, E, \ldots \in Down(\mathbb{N}^2)$. As an example, consider $D_{\text{ex}_1} \stackrel{\text{def}}{=} \mathbb{N}^2 \smallsetminus U_{\text{ex}_1}$ and $E_{\text{ex}_1} \stackrel{\text{def}}{=} \mathbb{N}^2 \smallsetminus V_{\text{ex}_1}$. We shall sometimes write $D_{\text{ex}_1} = \neg U_{\text{ex}_1}$ and $E_{\text{ex}_1} = \neg V_{\text{ex}_1}$.
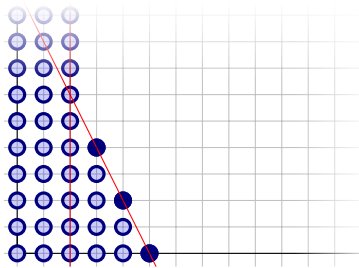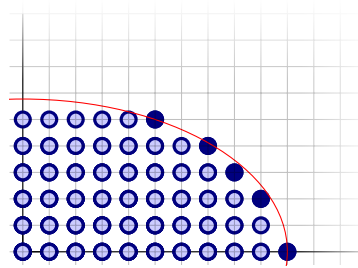
$$D_{\text{ex}_1}: \quad a \leq 2 \vee 2a + b \leq 10 \qquad\qquad E_{\text{ex}_1}: \quad a^2 + 3b^2 \leq 100$$

---

[4] This result is known as Haines' Theorem [30], and is also a consequence of Higman's Lemma: see Section 4.4.

Here, E_{ex_1} can be represented using its maximal points as generators:

$$E_{\mathrm{ex}_1} = \downarrow\langle 5, 5\rangle \cup \downarrow\langle 7, 4\rangle \cup \downarrow\langle 8, 3\rangle \cup \downarrow\langle 9, 2\rangle \cup \downarrow\langle 10, 0\rangle\,.$$

Representing downwards-closed sets via a finite "basis", i.e., as a finite union of principal ideals, of the form $\bigcup_{i<n} \downarrow x_i$, allows for simple and efficient algorithms, exactly as for upwards-closed subsets: one tests inclusion by comparing the generators of the ideals, and computes unions by gathering all generators and (optionally) removing non-maximal ones. For intersections one uses

$$\downarrow\langle a, b\rangle \cap \downarrow\langle a', b'\rangle = \downarrow\langle\min(a, a'), \min(b, b')\rangle \tag{2}$$

and the distribution law $(\bigcup_i \downarrow x_i) \bigcap (\bigcup_j \downarrow y_j) = \bigcup_i \bigcup_j (\downarrow x_i \cap \downarrow y_j)$, valid in every QO.

However, there is an important limitation here that we did not have with upwards-closed subsets: not all downwards-closed subsets in $\mathbb{N}^2$ can be generated from finitely many elements. Indeed, for any $x \in \mathbb{N}^2$, the ideal $\downarrow x$ is finite and thus only the finite downwards-closed subsets of $\mathbb{N}^2$ can be represented via principal ideals. Hence D_{ex_1} in the previous figure, or even $\mathbb{N}^2$ itself, while perfectly downwards-closed, cannot be represented in this way.

A possible solution is to represent a downwards-closed subset $D \in Down(X)$ via the finite basis of its upwards-closed complement, writing $D = X \smallsetminus \bigcup_{i<n} \uparrow x_i$, or also $D = X(\smallsetminus \uparrow x_i)_{i<n}$. Continuing our example, $D_{\mathrm{ex}_1} = \neg U_{\mathrm{ex}_1}$ can be written $D_{\mathrm{ex}_1} = \mathbb{N}^2 \smallsetminus \uparrow\langle 3, 5\rangle \smallsetminus \uparrow\langle 4, 3\rangle \smallsetminus \uparrow\langle 5, 1\rangle \smallsetminus \uparrow\langle 6, 0\rangle$. This representation *by excluded minors* is contrapositive and thus counter-intuitive. Computing intersections become easier while unions become harder, which is usually not what we want in applications. More annoyingly, constructing a representation of $\downarrow x$ from x involves actually computing complements, a task that can be difficult in general as we shall see later. Even in the easy $\mathbb{N}^2$ case, it is not transparent how from, e.g., $x = \langle 2, 3\rangle$, one gets to $\downarrow x = \mathbb{N}^2 \smallsetminus \uparrow\langle 0, 4\rangle \smallsetminus \uparrow\langle 3, 0\rangle$.

2.2.1 Downwards-closed subsets with ω's. In the case of $\mathbb{N}^2$, there exists an elegant solution to the representation problem for downwards-closed sets: one use pairs $\langle a, b\rangle \in \mathbb{N}_\omega^2$ where $\mathbb{N}_\omega$ extends $\mathbb{N}$ with an extra value ω that is larger than all natural numbers. We can now denote $D_{\mathrm{ex}_1} = \neg U_{\mathrm{ex}_1}$ (see last figure) as $\downarrow\langle 2, \omega\rangle \cup \downarrow\langle 3, 4\rangle \cup \downarrow\langle 4, 2\rangle \cup \downarrow\langle 5, 0\rangle$. We note that $\downarrow\langle 2, \omega\rangle$ should probably be written more explicitly as $(\downarrow 2) \times \mathbb{N}$ since it denotes $\{\langle c, d\rangle \mid a \le 2 \wedge b \in \mathbb{N}\}$, a subset of $\mathbb{N}^2$, not of $\mathbb{N}_\omega^2$, however the ω-notation inherited from vector addition systems [35] is now well-entrenched and we retain it here.

The sets of the form $\downarrow\langle a, b\rangle$ where $a, b \in \mathbb{N}_\omega$ are the *ideals*[5] of $\mathbb{N}^2$, and we see that they comprise the principal ideals as a special case. They also comprise infinite subsets and, for example, $\mathbb{N}^2 = \downarrow\langle\omega, \omega\rangle$ is one of them.

Using such ideals, all the downwards-closed subsets of $\mathbb{N}^2$ can be represented, and the algorithms for membership, inclusions, unions and intersections are just minor extensions of what we showed for finite downwards-closed sets, when all generators were

---

[5] We shall soon give the general definition. For now, the reader has to accept the $\mathbb{N}^2$ case.

proper elements of $\mathbb{N}^2$. The only difference is that we have to handle ω's in the obvious way when comparing generators (e.g., in inclusion tests) and when computing $\min$'s, e.g., in (2). Additionally, and like for upwards-closed subsets of $\mathbb{N}^2$, the representation of downwards-closed sets by the downward closure of incomparable elements is canonical, which here too brings in important algorithmic benefits.

Now that we have finite representations for both upwards-closed and downwards-closed subsets of $\mathbb{N}^2$, it is natural to ask whether we can compute complements.

It turns out that, for $\mathbb{N}^2$, this is an easy task. For complementing filters, one uses

$$\neg\uparrow\langle a, b\rangle = \left\{ \begin{array}{cc} \downarrow\langle a-1, \omega\rangle & \text{if } a > 0 \\ \emptyset & \text{otherwise} \end{array} \right\} \bigcup \left\{ \begin{array}{cc} \downarrow\langle \omega, b-1\rangle & \text{if } b > 0 \\ \emptyset & \text{otherwise} \end{array} \right\} . \qquad (3)$$

We see where the ω's are needed. In fact, only $\neg\uparrow\langle 0, 0\rangle = \emptyset$ does not involve ω's. We note that $\emptyset$, a downwards-closed subset, is indeed a finite union of ideals: it is the empty union.

Complementing an ideal is also easy:

$$\neg\downarrow\langle a, b\rangle = \left\{ \begin{array}{cc} \uparrow\langle a+1, 0\rangle & \text{if } a < \omega \\ \emptyset & \text{otherwise} \end{array} \right\} \bigcup \left\{ \begin{array}{cc} \uparrow\langle 0, b+1\rangle & \text{if } b < \omega \\ \emptyset & \text{otherwise} \end{array} \right\} . \qquad (4)$$

We see here that complementing an ideal in $\mathbb{N}^2$ always returns a union of principal filters, with no ω's.

Complementing an arbitrary upwards-closed subset U is easy if $U = \bigcup_{i<n} \uparrow x_i$ is given as a finite union of filters: we compute $\bigcap_{i<n}(X \smallsetminus \uparrow x_i)$. This needs complementing filters and intersecting downwards-closed sets, two operations we know how to perform on $\mathbb{N}^2$. Complementing an arbitrary downwards-closed subset $D = \bigcup_{i<n} \downarrow x_i$ is done similarly, even with $x_i \in \mathbb{N}_\omega^2$: we complement each ideal and intersect the resulting upwards-closed sets.

Finally, let us observe that, since any upwards-closed set is a finite union of filters, the proof that the complement $\neg\uparrow\langle a, b\rangle$ of any filter, and the intersection of any two ideals of $\mathbb{N}^2$, can be expressed as a finite union of ideals, entails that any downwards-closed $D \in Down(\mathbb{N}^2)$ is a finite union of ideals, as result known as *expressive completeness*.

2.2.2 Downwards-closed sets of subwords.
What about downwards-closed sets in $(A^*, \preceq)$? As with $\mathbb{N}^2$, finite unions of principal ideals, of the form $\downarrow u_1 \cup \cdots \cup \downarrow u_\ell$, are easy to compare and combine but they can only describe the finite downwards-closed languages. The contrapositive representation by excluded minors can describe any downwards-closed set but here too it is cumbersome. For example, let us fix $A = \{a, b, c\}$ and consider the language $D_{ex_3} = a^*b^*$, i.e., the set of all words composed of any number of a's followed by any number of b's: it is clear that D_{ex_3} is closed by taking subwords, hence $D_{ex_3} \in Down(A^*)$. Its representation by excluded minors is $D_{ex_3} = \neg(\uparrow ba \cup \uparrow c)$. That is, "a word $w \in A^*$ is in D_{ex_3} iff it does not contain any c, nor some b before an a": arguably, using a^*b^* to denote D_{ex_3} is clearer.

We do not develop this example further, and just announce that indeed the regular expression a^*b^* denotes an ideal of $(A^*, \preceq)$, as we shall show in Section 4.4. Furthermore, and as with $\mathbb{N}_\omega^2$, algorithms for comparing ideals in A^* are similar to algorithms

that compare elements of A^*. For example, testing whether (the language denoted by) a*b* is a subset of b*c*a* is essentially like testing whether ab is a subword of bca.

2.3 Well-quasi-orders

The previous section has made it clear that writing upwards-closed sets as a finite union of principal filters, when possible, is handy to compute with those sets. The quasi-orders for which it is possible to represent all upwards-closed sets as such is known: it is the class of well-quasi orders, which we introduce below.

A QO $(X, \leq)$ is *well-founded* $\overset{\text{def}}{\Leftrightarrow}$ it does not contain an infinite strictly decreasing sequence $x_0 > x_1 > x_2 > \cdots$. A subset $S \subseteq X$ is an *antichain* if for all distinct $x, y \in S$, neither of $x \leq y$ and $y \leq x$ holds. A QO is *well* (WQO) $\overset{\text{def}}{\Leftrightarrow}$ it is well-founded and does not contain an infinite antichain. Equivalently, $(X, \leq)$ is WQO iff every infinite sequence $(x_i)_{i \in \mathbb{N}}$ contains an infinite monotonic subsequence $x_{i_0} \leq x_{i_1} \leq x_{i_2} \leq \cdots$ with $i_0 < i_1 < i_2 < \cdots$. See [38,56] for proofs and other equivalent characterizations.

Example 2.1 (Some well-known WQOs).

linear orderings: $(\mathbb{N}, \leq)$ is a WQO, as is every ordinal or every well-founded linear-ordering.
words and sequences: $(\Sigma^*, \preceq)$, the set of words over a finite alphabet with the (scattered) subword ordering is a WQO. Variants and extensions abound [11,27,60]. By Higman's Lemma, for any WQO $(X, \leq)$, its sequence extension ordered by embedding, $(X^*, \leq_*)$, is a WQO too.
powersets: $(\mathcal{P}_f(X), \sqsubseteq_H)$, the set of all *finite* subsets of $(X, \leq)$ with Hoare's subset embedding is a WQO when X is. The full powerset $\mathcal{P}(X)$ is a WQO if X is an ω^2-WQO, a slightly stronger requirement than just being WQO, see [47].
trees: Labeled finite trees ordered by embedding form a WQO (Kruskal's Tree Theorem [37]).
graphs: Finite graphs ordered by the minor relation constitute a WQO (Robertson & Seymour's Graph Minor Theorem [50]). □

Coming back to our motivation, here is the result claimed at the beginning of this section:

Lemma 2.2 (Finite basis property). *If $(X, \leq)$ is WQO then every upwards-closed $U \in Up(X)$ contains a* finite basis $B \subseteq U$ such that $U = \bigcup_{x \in B} \uparrow x$.

It is easy to see that the converse holds: if every upwards-closed set has a finite basis, then $(X, \leq)$ is WQO.

Lemma 2.2 validates our choice of representing sets via a finite set of generators, as we did in our two motivating examples.

We close this section by mentioning another useful characterization of WQOs, see [38].

Lemma 2.3 (Ascending/Descending chain condition). *If $(X, \leq)$ is WQO then there exists no infinite strictly increasing sequence $U_0 \subsetneq U_1 \subsetneq U_2 \subsetneq \cdots$ of upwards-closed subsets. Dually, there exists no infinite strictly decreasing sequence $D_0 \supsetneq D_1 \supsetneq D_2 \supsetneq \cdots$ of downwards-closed subsets.*

In other words, $(Up(X), \supseteq)$ and $(Down(X), \subseteq)$ are well-founded posets.

2.4   Canonical prime decompositions of closed subsets

We now recall some basic facts about the canonical decompositions of upwards-closed and downwards-closed subsets in prime components.

Let $(X, \leq)$ be a WQO. We use Up and $Down$ as abbreviations for $Up(X)$ and $Down(X)$.

Definition 2.4 (Prime subsets). *1. A non-empty $U \in Up$ is* (up) prime *if for any $U_1, U_2 \in Up$, $U \subseteq (U_1 \cup U_2)$ implies $U \subseteq U_1$ or $U \subseteq U_2$.*
2. Similarly, a non-empty $D \in Down$ is (down) prime *if $D \subseteq (D_1 \cup D_2)$ implies $D \subseteq D_1$ or $D \subseteq D_2$.*

Observe that all principal filters are up prime and all principal ideals are down prime. Note also that, by definition, the empty subset is not prime.

Lemma 2.5 (Irreducibility). *1. $U \in Up$ is prime if, and only if, for all $U_1, \ldots, U_n \in Up$, $U = U_1 \cup \cdots \cup U_n$ implies $U = U_i$ for some i.*
2. $D \in Down$ is prime if, and only if, for all $D_1, \ldots, D_n \in Down$, $D = D_1 \cup \cdots \cup D_n$ implies $D = D_i$ for some i.

The following lemma highlights the importance of prime subsets.

Lemma 2.6. *1. Every upwards-closed set $U \in Up$ is a finite union of up primes.*
2. Every downwards-closed set $D \in Down$ is a finite union of down primes.

Proof. 1. is trivial: the finite basis property of WQOs (Lemma 2.2) shows that any upwards-closed set is a finite union of filters.

2. is a classical result, going back to Noether, see [7, Chapter VIII, Corollary, p.181]. We include a proof for the reader's convenience. That proceeds by well-founded induction on D in the well-founded poset $(Down, \subseteq)$ (Lemma 2.3). If D is empty, then it is an empty (hence finite) union of primes. If D is prime, the claim holds trivially. Finally, if $D \neq \emptyset$ is not prime, then by Lemma 2.5 it can be written as $D = D_1 \cup \cdots \cup D_n$ where each D_i is properly contained in D. By induction hypothesis each D_i is a finite union of primes. Hence D is too.   □

We say that a finite collection $\{P_1, \cdots, P_n\}$ of up (resp. down) primes is a *decomposition* of $U \in Up$ (resp., of $D \in Down$) if $U = P_1 \cup \cdots \cup P_n$ (resp., $D = P_1 \cup \cdots \cup P_n$). The decomposition is *minimal* if $P_i \subseteq P_j$ implies $i = j$.

Theorem 2.7 (Canonical decomposition). *Any upwards-closed U (resp. downwards-closed D) has a finite minimal decomposition. Furthermore this minimal decomposition is unique. We call it the canonical decomposition of U (resp. D).*

Proof. By Lemma 2.6, any U (or D) has a finite decomposition: U (or D) $= \bigcup_{i=1}^{n} P_i$. The decomposition can be made minimal by removing any P_i that is strictly included in some P_j. To prove uniqueness we assume that $\bigcup_{i=1}^{n} P_i = \bigcup_{j=1}^{m} P_j'$ are two minimal decompositions. From $P_i \subseteq \bigcup_j P_j'$, and since P_i is prime, we deduce that $P_i \subseteq P_{k_i}'$ for some k_i. Similarly, for each P_j' there is ℓ_j such that $P_j' \subseteq P_{\ell_j}$. The inclusions $P_i \subseteq P_{k_i}' \subseteq P_{\ell_{k_i}}$ require $i = \ell_{k_i}$ by minimality of the decomposition, hence are equalities $P_i = P_{k_i}'$. Similarly $j = k_{\ell_j}$ and $P_j' = P_{\ell_j}$ for any j. This one-to-one correspondence shows $\{P_1, \cdots, P_n\} = \{P_1', \cdots, P_m'\}$.   □

2.5   Filter decompositions and ideal decompositions

Definition 2.8 (Ideals). *A subset S of X is an* ideal *it if is non-empty, downwards-closed, and directed. We write $Idl(X) = \{I, J, \cdots\}$ for the set of all ideals of X.*

Recall that S is directed *if for all $x_1, x_2 \in S$, there exists $x \in S$ such that $x_1 \leq x$ and $x_2 \leq x$.*

A *filter* is a non-empty, upwards-closed, and filtered set S, where *filtered* means that for all $x_1, x_2 \in S$, there exists $x \in S$ such that $x \leq x_1, x_2$. In a WQO, the filters are exactly the principal filters, hence there is no need to introduce a new notion. We write $Fil(X)$ for the set of all (principal) filters of X.

Every principal ideal $\downarrow x$ is directed hence is an ideal. However not all ideals are principal. For example, in $(\mathbb{N}, \leq)$, the set $\mathbb{N}$ itself is an ideal (it is directed) and not of the form $\downarrow n$ for any $n \in \mathbb{N}$.

Remark 2.9. The above example illustrates a general phenomenon: the limit of an monotonic sequence of ideals (more generally, of a directed family of ideals) is an ideal. In particular, if $x_0 < x_1 < x_2 < \cdots$ is an infinite increasing sequence, $\bigcup_{i=0,1,2,\dots} \downarrow x_i$ is an ideal. It can be seen as the downward closure of a limit point, e.g. when one writes things like "$\bigcup_{n \in \mathbb{N}} \downarrow n = \downarrow \omega$". It turns out that $(Idl(X), \subseteq)$, the domain-theoretical *ideal completion* of X, is isomorphic to the *sobrification* $(\widehat{X}, \leq)$ —a topological completion— of $(X, \leq)$, see [16] for definitions and details.   □

The following appears for example as Lemma 1.1 in [34].

Proposition 2.10. *1. The up primes are exactly the filters.*
2. The down primes are exactly the ideals.

Proof. 1. is clear and we focus on 2.
($\Longrightarrow$): We only have to check that a down prime P is directed. Assume it is not. Then it contains two elements x_1, x_2 such that $\uparrow x_1 \cap \uparrow x_2 \cap P = \emptyset$. In other words, $P \subseteq (P \smallsetminus \uparrow x_1) \cup (P \smallsetminus \uparrow x_2)$. But $P \smallsetminus \uparrow x_i$ is downwards-closed for both $i = 1, 2$, so P being prime is included in one $P \smallsetminus \uparrow x_i$. This contradicts $x_i \in P$.
($\Longleftarrow$): Consider an ideal $I \subseteq X$. Aiming for a contradiction, we assume that $I \subseteq D_1 \cup D_2$ but $I \nsubseteq D_1$, $I \nsubseteq D_2$. Pick a point x_1 from $I \smallsetminus D_1$, and a point x_2 from $I \smallsetminus D_2$. Since I is directed, there is a point $x \in I$ such that $x_1, x_2 \leq x$. Since D_1 is downwards-closed, x is not in D_1, and similarly x is not in D_2, so x is not in $D_1 \cup D_2$, contradiction.   □

Proposition 2.10 explains why ideals appeared in our representation of downwards-closed sets of $\mathbb{N}^2$ in Section 2.2. There is a general reason: ideals are the down primes used in canonical decompositions, just like filters do for upwards-closed sets. Primality explains why the representation is canonical, and why comparing downwards-closed reduces to comparing generators. Meanwhile, the view of ideals as sets of the form $\downarrow x$ where x is either a normal point in X or, possibly, a limit point in $\widehat{X}$ —recall Remark 2.9— explains why comparing ideals is often very similar to comparing points —recall testing whether $\downarrow \langle 3, 4 \rangle \subseteq \downarrow \langle \omega, 1 \rangle$ or whether $\mathtt{a}^*\mathtt{b}^* \subseteq \mathtt{b}^*\mathtt{c}^*\mathtt{a}^*$.

3   Ideally effective WQOs

3.1   What do we mean by "effective" WQOs?

When describing generic algorithms for WQOs, one needs to make some basic computational assumptions on the WQOs at hand. Such assumptions are often summarized informally along the line of "*the WQO* $(X, \leq)$ *is effective*" and their precise meaning is often defined at a later stage, when one gives sufficient conditions based on the algorithm one is describing, a classic example being [19]. Sometimes the effectiveness assumptions are not even spelled out formally, e.g., when one has in mind applications where the WQO is $(\mathbb{N}^k, \leq_\times)$ or $(A^*, \preceq)$ which are obviously "effective" under all expected understandings.

The situation is different in this chapter since our goal is to provide a formal notion of effectiveness that is *preserved* by the main WQO constructions (and that supports the computation on closed subsets illustrated in Section 2.1). As a consequence, we cannot avoid giving a formal definition, even if this mostly amounts to administrative technicalities.

To simplify this task, we start by fixing the representation for closed subsets: these will be represented as finite unions of prime subsets as explained in Section 2. This provides a robust, generic, and convenient data structure for $Up(X)$ and $Down(X)$ based on data structures (to be defined) for $Fil(X)$ and $Idl(X)$. We do not require the decomposition to be canonical and leave this as an implementation choice (the underlying complexity trade-offs depend on the WQO and the application at hand). Moreover, and since all filters are principal in WQOs, any data structure for X can be reused for representing $Fil(X)$, so we will only need to assume that X and $Idl(X)$ have an effective presentation.

This leads to the following definition. Note that, rather than being completely formal and talk of recursive languages or Gödel numberings, we will allow considering more versatile data structures like terms, tuples, graphs, etc., that are closer to actual implementations. All data structures considered in this paper will be recursive sets, and in particular one can enumerate their elements.

Definition 3.1 (Ideally Effective WQOs). *A WQO* $(X, \leq)$ *further equipped with data structures for representing X and $Idl(X)$ is* ideally effective *if:*

(OD)   *the ordering $\leq$ is decidable on (the representation of) X;*

(ID)   *similarly, $\subseteq$ is decidable on $Idl(X)$;*

(PI)   *principal ideals are computable, that is,*
           $x \mapsto \downarrow x$ *is computable;*

(CF)   *complementation of filters, denoted*
           $\neg : Fil(X) \to Down(X)$, *is computable;*

(IF)   *intersection of filters, denoted*
           $\cap : Fil(X) \times Fil(X) \to Up(X)$, *is computable;*

(CI)   *complementation of ideals, denoted*
           $\neg : Idl(X) \to Up(X)$, *is computable;*

(II)   *intersection of ideals, denoted*
           $\cap : Idl(X) \times Idl(X) \to Down(X)$, *is computable.*

Some immediate remarks are in order:

– As mentioned earlier, elements of $Up(X)$ and $Down(X)$ are represented as collections (via lists, or sets, or ...) of elements of X and of $Idl(X)$ respectively. The computability of unions is thus trivial and therefore was not required in the formal definition.
– Similarly, checking membership $x \in D$ for downwards-closed sets reduces to deciding $\downarrow x \subseteq D$, hence was not required either.
– We said earlier that operations on Up and $Down$ boil down to operations on filters and ideals. Note that there are some subtleties. For example, deciding inclusions over $Up(X)$ or $Down(X)$ is made possible because the decompositions only use prime subsets. Explicitly, in order to check whether $D \subseteq D'$ for example, where $D = \bigcup_{i<m} I_i$ and $D' = \bigcup_{j<n} I'_j$, we check whether every I_i is included in some I'_j—this is correct because every ideal I_i is down prime.
– There is some asymmetry in the definition between upwards-closed and downwards-closed sets. This should be expected since WQOs are well-founded but the reverse orderings need not be.
– The astute reader may have noticed that the definition contains some hidden redundancies. Our proposal is justified by algorithmic efficiency concerns, see discussion in Section 6.1.

3.2   Some first ideally effective WQOs

We quickly show that the simplest WQOs are ideally effective. They will be used later as building blocks for more complex WQOs.

3.2.1   Finite orderings. A frequently occurring quasi-ordering in computer science is the *finite alphabet with n symbols*. It consists of a set with n elements, usually denoted A, ordered by equality. This is a WQO since A is finite. The name "alphabet" comes from its applications in language theory but this very basic WQO appears in many other situations, e.g., as colorings of some other objects, as the set of control states in formal models of computations such as Turing machines, communicating automata, etc.

Let us spell out, as a warming-up exercise, why this WQO $(A, =)$ is ideally effective. One can for instance represent elements of A using natural numbers up to $|A| - 1$. The ordering is trivially decidable. All ideals of $(A, =)$ are principal, that is of the form $\downarrow x$ for $x \in A$. We thus represent ideals as elements, exactly as we do for filters. Therefore, ideal inclusion coincides with the ordering, and (PI) is given by the identity function. All other operations are trivial: intersection of filters (*resp.* ideals) is always empty except if the two filters (*resp.* ideals) are equal, and $\neg\uparrow x = \neg\downarrow x = A \smallsetminus \{x\}$.

We could of course have dispensed with these explanations since more generally, any finite QO is a WQO and is ideally effective. In particular, all operations required by Definition 3.1 are always computable, being operations on a finite set. Let us note that all ideals are principal in this setting, which is no surprise since $(X, \geq)$ is also a WQO, and its filters are the ideals of $(X, \leq)$.

3.2.2 Natural numbers. Apart from finite orders, the simplest WQO is $(\mathbb{N}, \leq)$. We now restate our observations from Section 2.1 in the more formal framework of Definition 3.1.

Observe that since $\leq$ is linear, any downwards-closed set is actually an ideal, except for $\emptyset$. The ideals that are bounded from above have the form $\downarrow n$ for some $n \in \mathbb{N}$, and the only unbounded ideal is whole set $\mathbb{N}$ itself, often denoted $\downarrow \omega$ as we did in Section 2.2. Ideal inclusion is thus decidable: principal ideals are compared as elements, and $\downarrow \omega$ is larger than all the others. Thus $(Idl(\mathbb{N}), \subseteq)$ is linearly ordered, which makes intersections trivial: one has $\uparrow n \cap \uparrow m = \uparrow \max(n, m)$ and $\downarrow n \cap \downarrow m = \downarrow \min(n, m)$. Finally, complements are computed as follows:

$$\neg \uparrow (n+1) = \downarrow n \,, \qquad\qquad \neg \downarrow n = \uparrow (n+1) \,,$$
$$\neg \uparrow 0 = \emptyset \,, \qquad\qquad \neg \downarrow \omega = \emptyset \,.$$

3.2.3 Ordinals. The above analysis extends to any recursive linear WQO, i.e., any recursive ordinal (see [51] for definitions). Given an ordinal α, we write $\boldsymbol{\alpha}$ for the set of ordinals $\{\beta \mid \beta < \alpha\}$—the classical set-theoretic construction of ordinals equates α with $\boldsymbol{\alpha}$.

Let $(X, \leq) = (\boldsymbol{\alpha}, \leq)$. Once again, X being linearly ordered, its ideals are its downwards-closed sets (except $\emptyset$). Therefore, there are three types of ideals:

1. $I = X$,
2. I has a maximal element $\beta \in X$, in which case $I = \downarrow \beta$,
3. Or I has a supremum $\beta \in X \smallsetminus I$, in which case $I = \downarrow_{<} \beta = \boldsymbol{\beta}$.

Note that in the second case, $I = \downarrow \beta = \downarrow_{<}(\beta + 1) = \boldsymbol{\beta + 1}$. Thus every ideal of $(X, \leq)$ is a $\boldsymbol{\beta}$ for some $\beta \in \boldsymbol{\alpha + 1} \smallsetminus 0$, and ideal inclusion coincides with the natural ordering on $\boldsymbol{\alpha + 1}$.

Now, assuming that we can represent elements of X in a way that makes $\leq$ decidable, then $(X, \leq)$ is ideally effective. Indeed, the representation is easily extended to $(\boldsymbol{\alpha + 1}, \leq)$ and one can thus decide ideal inclusion. Intersections are computable as the maximum for filters, as the minimum for ideals. Finally, complements of filters and ideals are computed as follows:

$$\left. \begin{array}{ll} \neg \uparrow 0 = \emptyset \,, & \qquad \neg \uparrow \beta = \boldsymbol{\beta} \\ \neg \boldsymbol{\alpha} = \emptyset \,, & \qquad \neg \boldsymbol{\beta} = \uparrow \beta \end{array} \right\} \text{ for } \beta \in \boldsymbol{\alpha} \,.$$

While the above applies to any recursive ordinal, the applications that we are aware of usually only need ordinals below ϵ_0, for which the *Cantor Normal Form* is well known and understood, and leads to natural data structures [46]. One can push this at least to all ordinals below the larger ordinal Γ_0 [20].

Note that, when $\alpha = \omega$, the representation of ideals differs from the representation for $Idl(\mathbb{N})$ proposed in Section 3.2.2: in one case we use $\downarrow_{<} n$ while in the other we use $\downarrow n$. Both options are computationally equivalent.

4 Constructing ideally effective WQOs

We now look at more complex WQOs. In practice these are obtained by combining simpler WQOs via well-known operations like Cartesian product, sequences extension, etc. Our strategy is thus to show that these operations produce ideally effective WQOs when they are applied to ideally effective WQOs.

4.1 Ideally effective WQO constructors

We shall provide generic (i.e., uniform) algorithms that manage filters and ideals of compound WQOs by invoking the algorithms for the filters and ideals of their components. This is made precise in Definition 4.2, and to this end, we have to introduce the following notion:

Definition 4.1. *A presentation of an ideally effective WQO* $(X, \leq)$, *is a list of:*

>   — *data structures for* X *and* $Idl(X)$,
>   — *algorithms for the seven computable functions required by Definition 3.1,*

(XI) — *the ideal decomposition* $X = \bigcup_{i < n} I_i$ *of* X *as a downwards-closed set,*

(XF) — *as well as its filter decomposition* $X = \bigcup_{i < n'} F_i$.

   Obviously, a WQO is ideally effective if and only if it has a presentation as defined in Definition 4.1.

   The notion of presentations as actual objects is needed because they are the actual inputs of our WQO constructions. This explains why we added (XI) and (XF) in the requirements. For a given $(X, \leq)$, the ideal and filter decompositions of X always exist and requiring them in Definition 3.1 would make no sense. However, these decompositions are needed by algorithms who work uniformly on WQOs given via their presentations.

   Let us informally call *order-theoretic constructor* (constructor for short) any operation C that produces a quasi-ordering $C[(X_1, \leq_1), \ldots, (X_n, \leq_n)]$ from given quasi-orderings $(X_1, \leq_1), \ldots, (X_n, \leq_n)$. In subsequent sections, C will be instantiated with very well-known constructions, such as Cartesian product with componentwise ordering, finite sequences with Higman's ordering, finite sets with the Hoare quasi-ordering, and so on. In practice, we will always have $n = 1$ or 2. We also say that an order-theoretic constructor preserves WQO if $C[(X_1, \leq_1), \ldots, (X_n, \leq_n)]$ is a WQO whenever $(X_1, \leq_1), \ldots, (X_n, \leq_n)$ are. The constructors we just mentioned are well-known to be WQO-preserving. We extend this concept to ideally effective WQOs:

Definition 4.2. *An order-theoretic WQO-preserving constructor* C *is said to be* ideally effective *if:*

  – *It preserves ideal effectiveness, that is* $C[(X_1, \leq_1), \ldots, (X_n, \leq_n)]$ *is ideally effective when each* $(X_i, \leq_i)$ *is.*

– *A presentation of $C[(X_1, \leq_1), \dots, (X_n, \leq_n)]$ is uniformly computable from presentations of the WQOs $(X_i, \leq_i)$ $(i = 1, \dots, n)$.*

In the following sections, we proceed to prove that some of the most prominent WQO-preserving constructors are also ideally effective preserving.

4.2 Sums of WQOs

We start with two simple constructions, disjoint sums and lexicographic sums of WQOs. They will be our first examples of ideally effective constructions and will set the template for later constructions.

The *disjoint sum $X_1 \sqcup X_2$* of two QOs $(X_1, \leq_1)$ and $(X_2, \leq_2)$ is the set $\{1\} \times X_1 \cup \{2\} \times X_2$, quasi-ordered by:

$$\langle i, x \rangle \leq_\sqcup \langle j, y \rangle \text{ iff } i = j \text{ and } x \leq_i y \,.$$

We use $X_\sqcup$ to denote $X_1 \sqcup X_2$ and generally use the $\sqcup$ subscript to identify operations associated with the structure $(X_\sqcup, \leq_\sqcup)$. This structure is obviously well quasi-ordered when $(X_1, \leq_1)$ and $(X_2, \leq_2)$ are.

We let the reader check the following characterization.

Proposition 4.3 (Ideals of $X_1 \sqcup X_2$). *Given $(X_1, \leq_1)$ and $(X_2, \leq_2)$ two WQOs, the ideals of $(X_1 \sqcup X_2, \leq_\sqcup)$ are exactly the sets of the form $I = \{i\} \times J$ with $i \in \{1, 2\}$ and J an ideal of X_i.*

Thus $(Idl(X_1 \sqcup X_2), \subseteq)$ is isomorphic to $(Idl(X_1), \subseteq) \sqcup (Idl(X_2), \subseteq)$.

Given data structures for X_1 and X_2, we use the natural data structure for $X_1 \sqcup X_2$. Moreover, Proposition 4.3 shows that ideals of the WQO $(X_1 \sqcup X_2, \leq_\sqcup)$ can similarly be represented using data structure for $Idl(X_1)$ and $Idl(X_2)$.

Theorem 4.4. *With the above representations of elements and ideals, disjoint union is an ideally effective constructor.*

Proof (Sketch). Let $(X_1, \leq_1)$ and $(X_2, \leq_2)$ be two ideally effective WQOs.

In the following, we write $\bar{\imath}$ for $3 - i$ when $i \in \{1, 2\}$, so that $\{i, \bar{\imath}\} = \{1, 2\}$. We also abuse notation and, for a downwards-closed subset $D = \bigcup_a I_a$ of X_i, we write $\langle i, D \rangle$ to denote $\bigcup_a \langle i, I_a \rangle$, a downwards-closed subset of $X_\sqcup$ represented via ideals. Similarly, for an upwards-closed subset $U = \bigcup_a \uparrow_{X_i} x_a$ of X_i, we let $\langle i, U \rangle$ denote $\bigcup_a \uparrow_\sqcup \langle i, x_a \rangle$.

(OD): the definition of $\leq_\sqcup$ is already an implementation.
(ID): we use $\langle i, J \rangle \subseteq \langle i', J' \rangle \iff i = i' \wedge J \subseteq J'$.
(PI): we use $\downarrow_\sqcup \langle i, x \rangle = \langle i, \downarrow_i x \rangle$ for $i \in \{1, 2\}$.
(CF): we use $X_\sqcup \setminus \uparrow_\sqcup \langle i, x \rangle = \langle i, X_i \setminus \uparrow_i x \rangle \cup \langle \bar{\imath}, X_{\bar{\imath}} \rangle$. Note that this relies on (CF) for X_i (to express $X_i \setminus \uparrow_i x$ as a union of ideals) and on (XI) for $X_{\bar{\imath}}$.
(II): we rely on (II) for X_1 and X_2, using

$$\langle i, I \rangle \cap \langle j, J \rangle = \begin{cases} \langle i, I \cap J \rangle & \text{if } i = j, \\ \emptyset & \text{otherwise.} \end{cases}$$

Operations (CI) to complement ideals and (IF) to intersect filters are analogous.

Observe that the presentation of $(X_1 \sqcup X_2, \leq_\sqcup)$ described above is clearly computable from presentations for $(X_i, \leq_i)$ $(i = 1, 2)$. Notably, a filter (resp. ideal) decomposition of $X_1 \sqcup X_2$ is easily obtained by taking the union of filter (resp. ideal) decompositions of X_1 and X_2, thus establishing (XF) (resp. (XI)). $\square$

The *lexicographic sum* $X_1 \oplus X_2$ [6] of two QOs $(X_1, \leq_1)$, $(X_2, \leq_2)$ is the QO $(X_\oplus, \leq_\oplus)$ given by $X_\oplus = \{1\} \times X_1 \cup \{2\} \times X_2$ and

$$\langle i, x \rangle \leq_\oplus \langle j, y \rangle \text{ iff } i < j \text{ or } (i = j \text{ and } x \leq_i y).$$

Therefore $X_1 \oplus X_2$ and $X_1 \sqcup X_2$ share the same underlying set. The ordering $\leq_\oplus$ is an extension of $\leq_\sqcup$ hence is a WQO too, when $(X_1, \leq_1)$ and $(X_2, \leq_2)$ are.

Again, the following characterization is easy to obtain.

Proposition 4.5 (Ideals of $X_1 \oplus X_2$). *Given two WQOs $(X_1, \leq_1)$ and $(X_2, \leq_2)$, the ideals of $X_1 \oplus X_2$ are exactly the sets of the form $\{1\} \times J_1$ with $J_1 \in Idl(X_1)$, or of the form $\{1\} \times X_1 \cup \{2\} \times J_2$ with $J_2 \in Idl(X_2)$.*

Thus $(Idl(X_1 \oplus X_2), \subseteq)$ is isomorphic to $(Idl(X_1), \subseteq) \oplus (Idl(X_2), \subseteq)$, which leads to a simple data structure for the set of ideals[7] when X_1 and X_2 are effective.

Theorem 4.6. *With the above representations, lexicographic union is an ideally effective constructor.*

Proof (Sketch). We reuse the abbreviations $\langle i, U \rangle$, $\langle i, D \rangle$, $\bar{\imath}$, ..., introduced for disjoint sums. Also, we only consider the case where both X_1 and X_2 are non-empty (the claim is trivial otherwise).

(OD): follows from the definition.
(ID): ideal inclusion can be tested as the lexicographic sum of $Idl(X_1)$ and $Idl(X_2)$.
(PI): $\downarrow_\oplus \langle i, x \rangle$ is (represented by) $\langle i, \downarrow_i x \rangle$.
(CF): the complement $X_\oplus \smallsetminus \uparrow_\oplus \langle i, x \rangle$ is (represented by) $\langle i, X_i \smallsetminus \uparrow_i x \rangle$ except when $i = 2$ and $\uparrow_i x = X_2$, in which case $X_\oplus \smallsetminus \uparrow_\oplus \langle 2, x \rangle$ is $\langle 1, X_1 \rangle$.
(II): intersection of two ideals considers two cases. First $\langle 1, I \rangle \cap \langle 2, J \rangle$ is (represented by) $\langle 1, I \rangle$ for ideals issued from different components in $X_\oplus$. For $\langle i, I \rangle \cap \langle i, J \rangle$, i.e., ideals issued from the same component, we use $\langle i, I \cap J \rangle$ except when $i = 2$ and $I \cap J = \emptyset$, in which case $\langle 2, I \rangle \cap \langle 2, J \rangle$ is $\langle 1, X_1 \rangle$.

Procedures for the dual operations (CI) and (IF) are similar. Moreover, the presentation above is obviously computable from presentations for $(X_1, \leq_1)$ and $(X_2, \leq_2)$. Regarding (XI) and (XF), the ideal decomposition of $X_1 \oplus X_2$ is the ideal decomposition of X_2 and the filter decomposition of $X_1 \oplus X_2$ is the filter decomposition of X_1. $\square$

---

[6] A warning about notation: the lexicographic sum should not be confused with the natural sum of ordinals even if they are both denoted with $\oplus$. In particular, the lexicographic sum of ordinals is their usual addition.

[7] Note that with this representation, a pair $\langle i, J \rangle$ where $J \in Idl(X_i)$ denotes $\{1\} \times J$ when $i = 1$, and $\{1\} \times X_1 \cup \{2\} \times J$ —and not $\{2\} \times J$— when $i = 2$.

4.3   Products of WQOs and Dickson's lemma

Given two QOs $(X_1, \leq_1)$ and $(X_2, \leq_2)$, we define the *componentwise quasi-ordering* $\leq_\times$ on the Cartesian product $X_1 \times X_2$ by $\langle x_1, x_2 \rangle \leq_\times \langle y_1, y_2 \rangle \overset{\text{def}}{\Leftrightarrow} x_1 \leq_1 y_1 \wedge x_2 \leq_2 y_2$.

Lemma 4.7 (Dickson's lemma). *If X_1 and X_2 are WQOs, so is $X_1 \times X_2$.*

Proof (Idea). Given an infinite sequence in $X_1 \times X_2$, we extract an infinite sequence which is monotonic in the first component, and from that, an infinite sequence that is monotonic in the second component. □

The ideals of $(X_1 \times X_2, \leq_\times)$ are well known.

Proposition 4.8 (Ideals of $X_1 \times X_2$). *Let $(X_1, \leq_1)$ and $(X_2, \leq_2)$ be two WQOs. A subset I is an ideal of $X_1 \times X_2$ if, and only if, $I = I_1 \times I_2$ for some ideals I_1, I_2 of X_1 and X_2 respectively.*

Proof. ($\Leftarrow$): One checks that $I = I_1 \times I_2$ is non-empty, downwards-closed, and directed, when I_1 and I_2 are. For directedness, we consider two elements $\langle x_1, x_2 \rangle, \langle y_1, y_2 \rangle \in I$. Since I_1 is directed and contains x_1, y_1, it contains some z_1 with $x_1 \leq_1 z_1$ and $y_1 \leq_1 z_1$. Similarly I_2 contains some z_2 above x_2 and y_2 (wrt. $\leq_2$). Finally, $\langle z_1, z_2 \rangle$ is in I, and above both $\langle x_1, y_1 \rangle$ and $\langle x_2, y_2 \rangle$.

($\Rightarrow$): Consider $I \in Idl(X_1 \times X_2)$ and write I_1 and I_2 for its projections on X_1 and X_2. These projections are downwards-closed (since I is), non-empty (since I is) and directed (since I is), hence they are ideals (in X_1 and X_2). We now show that $I_1 \times I_2 \subseteq I$. Consider an arbitrary $x_1 \in I_1$: since I_1 is the projection of I, there is some $y_2 \in X_2$ such that $\langle x_1, y_2 \rangle \in I$. Similarly, for any $x_2 \in I_2$, there is some $y_1 \in X_1$ such that $\langle y_1, x_2 \rangle \in I$. Since I is directed, there is some $\langle z_1, z_2 \rangle \in I$ with $\langle x_1, y_2 \rangle \leq_\times \langle z_1, z_2 \rangle$ and $\langle y_1, x_2 \rangle \leq_\times \langle z_1, z_2 \rangle$. But then $x_1 \leq_1 z_1$ and $x_2 \leq_2 z_2$. Thus $\langle x_1, x_2 \rangle \in I$ since I contains $\langle z_1, z_2 \rangle$ and is downwards-closed. Hence $I = I_1 \times I_2$ and I is a product of ideals. □

Thus $Idl(X_1 \times X_2, \subseteq)$ is isomorphic to $(Idl(X_1), \subseteq) \times (Idl(X_2), \subseteq)$. If $(X_1, \leq_1)$ and $(X_2, \leq_2)$ are ideally effective, we naturally represent elements of $X_1 \times X_2$ as pairs of elements of X_1 and X_2, and similarly ideals of $(X_1 \times X_2, \leq_\times)$ as pairs of ideals of X_1 and X_2. This is notably how we handled $Idl(\mathbb{N}^2)$ in Section 2.2.

Theorem 4.9. *With the above representations, Cartesian product is an ideally effective constructor.*

Proof. Let D_1 and D_2 be downwards-closed sets of $(X_1, \leq_1)$ and $(X_2, \leq_2)$ respectively, given by some ideal decompositions $D_1 = \bigcup_i I_{1,i}$ and $D_2 = \bigcup_j I_{2,j}$. Then $D_1 \times D_2$ is downwards-closed in $X_1 \times X_2$, and it decomposes as $\bigcup_i \bigcup_j I_{1,i} \times I_{2,j}$ since products distribute over unions. The same reasoning holds for upwards-closed sets and their filter decompositions and we rely on these properties in the following explanations.

(OD): the ordering $\leq_\times$ is obviously decidable.

(ID): $I_1 \times I_2 \subseteq J_1 \times J_2$ iff $I_1 \subseteq J_1$ and $I_2 \subseteq J_2$ (exercise; the nonemptiness of ideals is required here).

(PI): $\downarrow\langle x_1, x_2\rangle = \downarrow x_1 \times \downarrow x_2$.

(II): to compute intersections, use $(I_1 \times I_2) \cap (I'_1 \times I'_2) = (I_1 \cap I'_1) \times (I_2 \cap I'_2)$, and build the product of downwards-closed sets as explained above.

(CF): to complement filters, use $(X_1 \times X_2) \smallsetminus \uparrow_\times \langle x_1, x_2\rangle = \big[(X_1 \smallsetminus \uparrow x_1) \times X_2\big] \cup \big[X_1 \times (X_2 \smallsetminus \uparrow x_2)\big]$ and build products of downwards-closed sets.

Procedures for the remaining operations are obtained similarly. Note that here too, the presentation above is computable from presentations for $(X_1, \leq_1)$ and $(X_2, \leq_2)$. Notably, a filter and ideal decomposition of $X_1 \times X_2$ is easily obtained from decompositions of X_1 and X_2, by distributing products over unions.   □

4.4 Sequence extensions of WQOs and Higman's Lemma

Given a QO $(X, \leq)$, we denote by X^* the *sequence extension* of X, i.e., the set of all finite sequences over X, often called *words* when X is an alphabet. We write ϵ for the empty (zero-length) sequence, and denote multiplicatively the concatenation of sequences, as uv or $u \cdot v$. Elements of X^* will be denoted in bold font, such as $u, v, ...$, while elements of X are denoted $x, y, ...$. In particular, if $x \in X$, then $x \in X^*$ denotes the sequence of length one containing only the symbol x.

The set X^* is often quasi-ordered with *Higman's quasi-ordering* $\leq_*$, also known as the *sequence embedding* quasi-ordering, defined by $u = x_1 \cdots x_n \leq_* v = y_1 \cdots y_m$ $\overset{\text{def}}{\Leftrightarrow}$ there are n indices $1 \leq p_1 < p_2 < \cdots < p_n \leq m$ such that $x_i \leq y_{p_i}$ for each $i = 1, \ldots, n$. In other words, and writing $[n]$ for the set $\{1, \cdots, n\}$, there is a strictly increasing mapping p from $[n]$ to $[m]$ such that $x_i \leq y_{p(i)}$. Such a mapping will be called a *witness* of $u \leq_* v$. Equivalently, $u \leq_* v$ if v contains a length n subsequence $v' = y_{p_1} \cdots y_{p_n}$ such that $u \leq_\times v'$ using the product ordering from Section 4.3.

The structure $(X^*, \leq_*)$ is sometimes called the *Higman extension* of $(X, \leq)$. This constructor preserves WQO: this is Higman's Lemma [32].

Showing that this constructor is ideally effective requires some work and Section 4.4 is one of the longest in this chapter. This is justified by the importance of this construction. Being generic, our algorithms apply to non-trivial instances such as $(\mathbb{N}^k)^*$ —used in Timed-arc nets [29], in data nets [39], for runs of Vector Addition Systems [43]—, or $(\Sigma^*)^k \times (\Sigma^*)^*$ —used in Dynamic Lossy Channel Systems [1]—, or to even richer settings like the Priority Channel Systems and the Higher-Order Channel Systems of [27]. The algorithms for $(X^*, \leq_*)$ are also invoked when showing ideal effectiveness of many WQOs derived from X^*.

Before we study the ideals of X^*, let us first lift the concatenation of sequences to sets of sequences: the product (for concatenation) of two sets of sequences $U, V \subseteq X^*$ is denoted $U \cdot V \overset{\text{def}}{=} \{u \cdot v \mid u \in U, v \in V\}$. A useful property of $(X^*, \leq_*)$ is that the concatenation of downwards-closed sets distributes over intersection:

Lemma 4.10. *Let $D_1, D_2, D \in Down(X^*)$. Then $(D_1 \cap D_2) \cdot D = (D_1 \cdot D) \cap (D_2 \cdot D)$.*

Proof. The left-to-right inclusion is obvious. For the right-to-left inclusion, let $w \in D_1 \cdot D \cap D_2 \cdot D$. Then $w = u_1 v_1$ for some $u_1 \in D_1$ and $v_1 \in D$. Also, $w = u_2 v_2$ for some $u_2 \in D_2$ and $v_2 \in D$. One of u_1 and u_2 is a prefix of the other. Assume u_1 is a prefix of u_2 (the other case is analogous). Since D_2 is downwards-closed and $u_1 \leq_* u_2$, $u_1 \in D_2$. Thus, $u_1 \in D_1 \cap D_2$ and $w = u_1 v_1 \in (D_1 \cap D_2) \cdot D$.   □

The structure of ideals of $(X^*, \leq_*)$ is given in [34] where the following theorem is proved. An alternate proof is presented in Section 4.4.2.

Theorem 4.11. *Given a WQO* $(X, \leq)$, *the ideals of* $(X^*, \leq_*)$ *are exactly the finite products of atoms, of the form* $P = A_1 \cdot A_2 \cdots A_n$ *where* atoms *are:*

- *any set of the form* $A = D^*$ *for* $D \in Down(X)$,
- *any set of the form* $A = I + \epsilon \stackrel{def}{=} \{x \mid x \in I\} \cup \{\epsilon\}$ *for* $I \in Idl(X)$.

4.4.1   Ideal effectiveness. The elements of X^* will be represented in the natural way, e.g., via lists of elements of X (assuming a data structure for X). When $(X, \leq)$ is ideally effective, Theorem 4.11 leads to a natural data structure for ideals of X^*, as lists of atoms, where the representation of atoms is directly inherited from those for $Idl(X)$ and $Down(X)$.

Theorem 4.12. *With the above representations, the sequence extension is an ideally effective constructor.*

Proof. Let $(X, \leq)$ be an ideally effective WQO.

(OD): deciding $\leq_*$ over X^* reduces to comparing elements of X, *e.g.* by looking for a *leftmost embedding*.

(PI): given a finite sequence $u = x_1 \cdots x_n$, the principal ideal $\downarrow u$ is represented by the product $(\downarrow x_1 + \epsilon) \cdots (\downarrow x_n + \epsilon)$.

Procedures for the remaining operations required by Definition 3.1 are more elaborate, and we therefore introduce a lemma for each one. This series of lemmas concludes the proof since the fact that a presentation of $(X^*, \leq_*)$ can be uniformly computed from a presentation of $(X, \leq)$ will be clear. As for (XF), the filter decomposition of $X^* = \uparrow \epsilon$ is given by the empty sequence (and does not depend on X), while for (XI) we note that X^* is already an ideal made of a single atom.   □

Subsequently, $(X, \leq)$ denotes an ideally effective WQO. We begin with ideal inclusion. A similar procedure was already obtained by Abdulla et al. in the case where X is a finite alphabet with equality [3].

Lemma 4.13 (ID). *Inclusion between ideals of* $(X^*, \leq_*)$ *can be tested using a linear number of inclusion tests between downwards-closed sets of* X, *using a version of* leftmost embedding *search. The following equations implicitly describe an algorithm deciding inclusion by induction on the length, or number of atoms, of ideals:*

1. *Atoms are compared as follows:*

$$(I_1 + \epsilon) \subseteq (I_2 + \epsilon) \iff I_1 \subseteq I_2, \tag{5.1}$$

$$(I + \epsilon) \subseteq D^* \iff I \subseteq D, \tag{5.2}$$

$$D_1^* \subseteq D_2^* \iff D_1 \subseteq D_2, \tag{5.3}$$

$$D^* \subseteq (I + \epsilon) \iff D = \emptyset. \tag{5.4}$$

2. *For any ideal $\boldsymbol{P}$: $\boldsymbol{\epsilon} \subseteq \boldsymbol{P}$.*
3. *For any ideal $\boldsymbol{P}$ and atom $\boldsymbol{A}$: $\boldsymbol{A} \cdot \boldsymbol{P} \subseteq \boldsymbol{\epsilon} \iff \boldsymbol{A} = \emptyset^* \wedge \boldsymbol{P} \subseteq \boldsymbol{\epsilon}$.*
4. *Finally, for all atoms $\boldsymbol{A}$ and $\boldsymbol{B}$, and ideals $\boldsymbol{P}$ and $\boldsymbol{Q}$:*
   (a) if $\boldsymbol{A} \not\subseteq \boldsymbol{B}$ then:

   $$\boldsymbol{A} \cdot \boldsymbol{P} \subseteq \boldsymbol{B} \cdot \boldsymbol{Q} \iff \boldsymbol{A} \cdot \boldsymbol{P} \subseteq \boldsymbol{Q} \,;$$

   (b) if $\boldsymbol{A} \subseteq \boldsymbol{B}$ as in (5.1), i.e., $\boldsymbol{A} = (I_1 + \epsilon)$, $\boldsymbol{B} = (I_2 + \epsilon)$ for some $I_1, I_2 \in Idl(X)$, then:

   $$\boldsymbol{A} \cdot \boldsymbol{P} \subseteq \boldsymbol{B} \cdot \boldsymbol{Q} \iff \boldsymbol{P} \subseteq \boldsymbol{Q} \,;$$

   (c) if $\boldsymbol{A} \subseteq \boldsymbol{B}$ as in any of Eqs. (5.2) to (5.4), then:

   $$\boldsymbol{A} \cdot \boldsymbol{P} \subseteq \boldsymbol{B} \cdot \boldsymbol{Q} \iff \boldsymbol{P} \subseteq \boldsymbol{B} \cdot \boldsymbol{Q} \,.$$

Proof. The first three cases are trivial. We concentrate on the fourth one.

4a It is always true that $\boldsymbol{A} \cdot \boldsymbol{P} \subseteq \boldsymbol{Q}$ implies $\boldsymbol{A} \cdot \boldsymbol{P} \subseteq \boldsymbol{B} \cdot \boldsymbol{Q}$. Conversely, let $\boldsymbol{u} \in \boldsymbol{A}$ and $\boldsymbol{v} \in \boldsymbol{P}$, so that $\boldsymbol{uv} \in \boldsymbol{A} \cdot \boldsymbol{P} \subseteq \boldsymbol{B} \cdot \boldsymbol{Q}$. Assuming $\boldsymbol{A} \not\subseteq \boldsymbol{B}$, there exists $\boldsymbol{w}' \in \boldsymbol{A} \smallsetminus \boldsymbol{B}$ and by directedness, there exists a word $\boldsymbol{w} \in \boldsymbol{A}$ such that $\boldsymbol{w} \geq_* \boldsymbol{w}', \boldsymbol{u}$. In particular, $\boldsymbol{w}$ is in $\boldsymbol{A} \smallsetminus \boldsymbol{B}$ and $\boldsymbol{w} \geq_* \boldsymbol{u}$.
   If $\boldsymbol{A} = I + \epsilon$ for some $I \in Idl(X)$, then $\boldsymbol{w}$ is of length at most one. Since $\boldsymbol{w}' \notin \boldsymbol{B}$, in particular $\boldsymbol{w}' \neq \epsilon$, so $\boldsymbol{w}$ is of length exactly one. Also, since $\boldsymbol{w} \notin \boldsymbol{B}$, the word $\boldsymbol{wv}$, which is in $\boldsymbol{A} \cdot \boldsymbol{P} \subseteq \boldsymbol{B} \cdot \boldsymbol{Q}$ has to actually be in $\boldsymbol{Q}$. Since $\boldsymbol{Q}$ is downwards-closed, $\boldsymbol{uv} \in \boldsymbol{Q}$.
   Otherwise, $\boldsymbol{A} = D^*$ for some $D \in Down(X)$. In this case, $\boldsymbol{ww} \in \boldsymbol{A}$ and thus $\boldsymbol{wwv} \in \boldsymbol{B} \cdot \boldsymbol{Q}$. We factor $\boldsymbol{wwv}$ as $\boldsymbol{v}_1 \boldsymbol{v}_2$ with $\boldsymbol{v}_1 \in \boldsymbol{B}$ and $\boldsymbol{v}_2 \in \boldsymbol{Q}$. Since $\boldsymbol{w}$ is not in $\boldsymbol{B}$, no word of which $\boldsymbol{w}$ is a prefix is in $\boldsymbol{B}$ either, and that implies that $\boldsymbol{v}_1$ is a proper prefix of $\boldsymbol{w}$, and that $\boldsymbol{v}_2$ has $\boldsymbol{wv}$ as a suffix. In particular, $\boldsymbol{v}_2 \geq_* \boldsymbol{wv}$. Recalling that $\boldsymbol{wv} \geq_* \boldsymbol{uv}$ and that $\boldsymbol{Q}$ is downwards-closed, $\boldsymbol{uv}$ is in $\boldsymbol{Q}$.
4b Here also, the right-to-left implication is trivial. Conversely, assume $\boldsymbol{A} \cdot \boldsymbol{P} \subseteq \boldsymbol{B} \cdot \boldsymbol{Q}$ and $\boldsymbol{A} = (I_1 + \epsilon)$ and $\boldsymbol{B} = (I_2 + \epsilon)$ for some $I_1 \subseteq I_2 \in Idl(X)$. Let $\boldsymbol{u} \in \boldsymbol{P}$. Pick $x \in I_1$: $x\boldsymbol{u} \in \boldsymbol{A} \cdot \boldsymbol{P}$, thus $x\boldsymbol{u} \in \boldsymbol{B} \cdot \boldsymbol{Q}$. Therefore, $\boldsymbol{u} \in \boldsymbol{Q}$ since sequences of $\boldsymbol{B}$ have length at most one.
4c The left-to-right implication is trivial, since $\epsilon \in \boldsymbol{A}$. For the other implication, we consider some $\boldsymbol{u} \in \boldsymbol{A}$ and $\boldsymbol{v} \in \boldsymbol{P}$, and we have to show that $\boldsymbol{uv} \in \boldsymbol{B} \cdot \boldsymbol{Q}$. Since $\boldsymbol{P} \subseteq \boldsymbol{B} \cdot \boldsymbol{Q}$, we can factor $\boldsymbol{v}$ as $\boldsymbol{v}_1 \boldsymbol{v}_2$ with $\boldsymbol{v}_1 \in \boldsymbol{B}$ and $\boldsymbol{v}_2 \in \boldsymbol{Q}$. We claim that $\boldsymbol{uv}_1 \in \boldsymbol{B}$: if $\boldsymbol{B} = D^*$ is an atom of the second kind, the claim follows from $\boldsymbol{A} \subseteq \boldsymbol{B}$; if $\boldsymbol{B} = I + \epsilon$ is an atom of the first kind, then we are in case (5.4), $\boldsymbol{A}$ is $\emptyset^*$, and $\boldsymbol{u} = \epsilon$. With $\boldsymbol{uv}_1 \in \boldsymbol{B}$ we have $\boldsymbol{uv} \in \boldsymbol{B} \cdot \boldsymbol{Q}$ as needed.   □

The next lemma deals with the complementation of filters:

Lemma 4.14 (CF). *Given $\boldsymbol{w} \in X^*$, the downwards-closed set $X^* \smallsetminus \uparrow\boldsymbol{w}$ can be computed inductively using the following equations:*

$$X^* \smallsetminus \uparrow\epsilon = \emptyset \text{ (empty union),} \tag{6}$$

$$X^* \smallsetminus \uparrow x\boldsymbol{v} = \begin{cases} (X \smallsetminus \uparrow x)^* & \text{if } \boldsymbol{v} = \epsilon, \\ (X \smallsetminus \uparrow x)^* \cdot (X + \epsilon) \cdot (X^* \smallsetminus \uparrow\boldsymbol{v}) & \text{otherwise.} \end{cases} \tag{7}$$

Note that X might not be an ideal, in which case $X + \epsilon$ is not an atom in Eq. (7). In this case, one has to first get the ideal decomposition $X = \bigcup_i I_i$ from a presentation of $(X, \leq)$ and use distributivity of concatenation over unions.

In the commonly encountered case where X is a finite alphabet, ordered by equality, there is no need to distribute, and indeed, the complement of a filter is always an ideal. More precisely, if $X = \{a_1, \dots, a_n\}$ is a finite alphabet under equality, then one checks easily that $(X \smallsetminus \uparrow a_i)^* \cdot (X + \epsilon) = \{a_j \mid j \neq i\}^* \cdot (a_i + \epsilon)$. It follows that complement of filters are ideals in this case.

Remark 4.15. Kabil and Pouzet [34] use the following (equivalent) expression to complement filters:

$$X^* \smallsetminus \uparrow xy\boldsymbol{w} = (X \smallsetminus \uparrow x)^* \cdot [\downarrow(\uparrow x \cap \uparrow y) + \epsilon] \cdot (X^* \smallsetminus \uparrow y\boldsymbol{w}) \,. \tag{8}$$

We used a different formula because, in general, our setting does not guarantees that the expression $\downarrow U$ is computable for $U \in Up(X)$, even in the particular case where $U = \uparrow x \cap \uparrow y$. It is fair to mention that Kabil and Pouzet make no claim on computability.

Still, Eq. (8) is interesting when X is a finite alphabet since then the expression $\uparrow x \cap \uparrow y$ either denotes the empty set or $(x + \epsilon)$, depending on whether x and y coincide. Therefore, using Eq. (8), one directly obtains an ideal written in canonical form (a notion defined below, in Section 4.4.3). $\square$

Proof (of Lemma 4.14).
We only prove the second case of Eq. (7) since the other equalities are obvious.
($\supseteq$): Let $\boldsymbol{w}' = \boldsymbol{u}y\boldsymbol{w}$ with $\boldsymbol{u} \in (X \smallsetminus \uparrow x)^*$, $y \in X + \epsilon$ and $\boldsymbol{w} \in (X^* \smallsetminus \uparrow\boldsymbol{v})$. Thus $\boldsymbol{v} \not\leq_* \boldsymbol{w}$. Since y has length at most 1, we deduce $x\boldsymbol{v} \not\leq_* y\boldsymbol{w}$. Since all elements in $\boldsymbol{u}$ are taken from $X \smallsetminus \uparrow x$, we further have $x\boldsymbol{v} \not\leq_* \boldsymbol{u}y\boldsymbol{v}$. Therefore $\boldsymbol{w}' \in X^* \smallsetminus \uparrow x\boldsymbol{v}$.
($\subseteq$): Let $\boldsymbol{w}' \notin \uparrow x\boldsymbol{v}$. Then either $\boldsymbol{w}' \in (X \smallsetminus \uparrow x)^*$, or we can write $\boldsymbol{w}' = \boldsymbol{u}y\boldsymbol{w}$ with $\boldsymbol{u} \in (X \smallsetminus \uparrow x)^*$ and $y \geq x$. Moreover, $\boldsymbol{w} \notin \uparrow\boldsymbol{v}$, since otherwise $x\boldsymbol{v} \leq_* y\boldsymbol{w} \leq_* \boldsymbol{u}y\boldsymbol{w} = \boldsymbol{w}'$. Therefore, $\boldsymbol{w}' \in (X \smallsetminus \uparrow x)^* \cdot X \cdot (X^* \smallsetminus \uparrow\boldsymbol{v})$. Joining the two cases, and since $\epsilon \in (X^* \smallsetminus \uparrow\boldsymbol{v})$, we obtain the required $\boldsymbol{w}' \in (X \smallsetminus \uparrow x)^* \cdot (X + \epsilon) \cdot (X^* \smallsetminus \uparrow\boldsymbol{v})$. $\square$

We now show how to intersect ideals:

Lemma 4.16 (II). *The intersection of two ideals of $(X^*, \leq_*)$ can be computed inductively using the following equations:*

$$\epsilon \cap \boldsymbol{Q} = \boldsymbol{P} \cap \epsilon = \epsilon\,, \tag{9}$$

$$D_1^* \cdot \boldsymbol{P} \cap D_2^* \cdot \boldsymbol{Q} = (D_1 \cap D_2)^* \cdot \begin{bmatrix} (D_1^* \cdot \boldsymbol{P}) \cap \boldsymbol{Q} \\ \cup\ \boldsymbol{P} \cap (D_2^* \cdot \boldsymbol{Q}) \end{bmatrix}, \tag{10}$$

$$(I_1 + \epsilon) \cdot \boldsymbol{P} \cap (I_2 + \epsilon) \cdot \boldsymbol{Q} = \begin{bmatrix} ((I_1 + \epsilon) \cdot \boldsymbol{P}) \cap \boldsymbol{Q} \\ \cup\ \boldsymbol{P} \cap ((I_2 + \epsilon) \cdot \boldsymbol{Q}) \\ \cup\ ((I_1 \cap I_2) + \epsilon) \cdot (\boldsymbol{P} \cap \boldsymbol{Q}) \end{bmatrix}, \tag{11}$$

$$D^* \cdot \boldsymbol{P} \cap (I + \epsilon) \cdot \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{P} \cap ((I + \epsilon) \cdot \boldsymbol{Q}) \\ \cup\ ((D \cap I) + \epsilon) \cdot ((D^* \cdot \boldsymbol{P}) \cap \boldsymbol{Q}) \end{bmatrix}. \tag{12}$$

Here also, some shortcuts are used. For instance, the intersection of two ideals need not be an ideal. Therefore, $(I_1 \cap I_2) + \epsilon$ in Eq. (11) might not be an ideal. As before, by decomposing downwards-closed sets as union of ideals, and distributing concatenations over unions, one can compute the actual ideal decomposition of the intersection of two ideals of $(X^*, \leq_*)$.

Proof (of Lemma 4.16). Equation (9) is obviously correct. The other right-to-left inclusions are easily checked using Lemma 4.13. For the left-to-right inclusions:

Eq. (10): Let $\boldsymbol{u} \in D_1^* \cdot \boldsymbol{P} \cap D_2^* \cdot \boldsymbol{Q}$. Let $\boldsymbol{v}$ be the longest prefix of $\boldsymbol{u}$ which is in D_1^*. Without loss of generality, we assume that the longest prefix of $\boldsymbol{u}$ which is in D_2^* is longer than $|\boldsymbol{v}|$, and thus can be written $\boldsymbol{vw}$ for some $\boldsymbol{w} \in D_2^*$. Moreover, there exists $\boldsymbol{t} \in X^*$ so that $\boldsymbol{u} = \boldsymbol{vwt}$. We have $\boldsymbol{v} \in (D_1 \cap D_2)^*$, $\boldsymbol{wt} \in \boldsymbol{P}$ and $\boldsymbol{t} \in \boldsymbol{Q}$. Therefore, $\boldsymbol{wt} \in \boldsymbol{P} \cap D_2^* \cdot \boldsymbol{Q}$.

Eq. (11): Consider any word in $(I_1 + \epsilon) \cdot \boldsymbol{P} \cap (I_2 + \epsilon) \cdot \boldsymbol{Q}$. If it is empty, it is also in the right-hand side of Eq. (11), so we assume that it of the form $x\boldsymbol{u}$. Depending on whether $x \in I_1 \smallsetminus I_2$, $x \in I_2 \smallsetminus I_1$ or $x \in I_1 \cap I_2$, $\boldsymbol{u}$ is easily proved to be in $((I_1 + \epsilon) \cdot \boldsymbol{P}) \cap \boldsymbol{Q}$, in $\boldsymbol{P} \cap ((I_2 + \epsilon) \cdot \boldsymbol{Q})$, or in $((I_1 \cap I_2) + \epsilon) \cdot (\boldsymbol{P} \cap \boldsymbol{Q})$. If x is neither in I_1 nor I_2, then $x\boldsymbol{u}$ belongs to all three sets.

Eq. (12): This is similar, combining arguments from the previous two cases. $\qquad\square$

We now turn to intersecting filters:

Lemma 4.17 (IF). *The intersection of two filters can be computed inductively using the following equations:*

$$\uparrow\!\boldsymbol{v} \cap \uparrow\!\epsilon = \uparrow\!\epsilon \cap \uparrow\!\boldsymbol{v} = \uparrow\!\boldsymbol{v}\,, \tag{13}$$

$$\uparrow\!x\boldsymbol{v} \cap \uparrow\!y\boldsymbol{w} = \begin{bmatrix} (\uparrow\!\boldsymbol{x}) \cdot (\uparrow\!\boldsymbol{v} \cap \uparrow\!y\boldsymbol{w})\ \cup\ (\uparrow\!\boldsymbol{y}) \cdot (\uparrow\!x\boldsymbol{v} \cap \uparrow\!\boldsymbol{w}) \\ \cup\ (\uparrow_X x \cap \uparrow_X y) \cdot (\uparrow\!\boldsymbol{v} \cap \uparrow\!\boldsymbol{w}) \end{bmatrix}, \tag{14}$$

where $\boldsymbol{v}, \boldsymbol{w} \in X^*$ and $x, y \in X$. The actual filter decomposition in the last equation is obtained using $(\uparrow\!\boldsymbol{u}) \cdot (\uparrow\!\boldsymbol{u}') = \uparrow\!(\boldsymbol{uu}')$ and distributivity over unions.

Proof. Equation (13) and the "$\supseteq$" half of Eq. (14) are obvious. For the remaining "$\subseteq$" half, we consider $\boldsymbol{u} \in \uparrow\!x\boldsymbol{v} \cap \uparrow\!y\boldsymbol{w}$. Let us write $\boldsymbol{u}$ as $\boldsymbol{u} = \boldsymbol{u}_1 z \boldsymbol{u}_2$ where $z\boldsymbol{u}_2$ is the

shortest suffix of $\boldsymbol{u}$ in $\uparrow x\boldsymbol{v} \cap \uparrow y\boldsymbol{w}$—this suffix cannot be empty since it contains $x\boldsymbol{v}$ and $y\boldsymbol{w}$ as embedded sequences. Note that z must be above x or y in X, otherwise $\boldsymbol{u}_2$ would be a shorter suffix of $\boldsymbol{u}$ in $\uparrow x\boldsymbol{v} \cap \uparrow y\boldsymbol{w}$. Depending on whether z is above x, y, or both, we conclude. $\qquad\square$

Finally, we focus on the complementation of ideals. This operation requires more work, and is decomposed in several lemmas. We first show how to complement atoms, and then how to complement products of atoms.

- If $D \subseteq X$ is downwards-closed, then $X^* \smallsetminus D^*$, also written $\neg D^*$, consists of all sequences having at least one element not in D. One first computes $X \smallsetminus D = \uparrow a_1 \cup \cdots \cup \uparrow a_n$, using (CI) for X. Then $\neg D^* = \uparrow_{X^*} \boldsymbol{a}_1 \cup \cdots \cup \uparrow_{X^*} \boldsymbol{a}_n$.
- If $I \subseteq X$ is an ideal, $\neg(I + \epsilon)$ consists of all sequences of length at least 2, as well as all sequences having an element not in I. The latter is obtained as in the previous case, by computing $X \smallsetminus I = \uparrow b_1 \cup \cdots \cup \uparrow b_m$ in X. The former is $\uparrow_{X^*}(X \cdot X)$, easily computed in a similar way using (XF) for X.

We now consider products $\boldsymbol{A}_1 \cdots \boldsymbol{A}_n$ of atoms. We know how to compute $\boldsymbol{U}_i = \neg \boldsymbol{A}_i$. Trivially, $\neg(\boldsymbol{A}_1 \cdots \boldsymbol{A}_n) = \neg(\neg \boldsymbol{U}_1 \cdots \neg \boldsymbol{U}_n)$, and this motivates the following definition:

Definition 4.18. *Define the operator $\odot\colon Up(X^*) \times Up(X^*) \to Up(X^*)$ as* $\boldsymbol{U} \odot \boldsymbol{V} := \neg(\neg \boldsymbol{U} \cdot \neg \boldsymbol{V})$.

Note that $\boldsymbol{U} \odot \boldsymbol{V}$ is upwards-closed when $\boldsymbol{U}$ and $\boldsymbol{V}$ are. The operation $\odot$ is easily shown associative using the associativity of the product, thus $\boldsymbol{U}_1 \odot \cdots \odot \boldsymbol{U}_n = \neg(\neg \boldsymbol{U}_1 \cdot \cdots \cdot \neg \boldsymbol{U}_n)$. The previous relation becomes $\neg(\boldsymbol{A}_1 \cdots \boldsymbol{A}_n) = \boldsymbol{U}_1 \odot \cdots \odot \boldsymbol{U}_n$, and it only remains to show that $\odot$ is computable on upwards-closed sets. In what follows, we will often use the following obvious characterization: $\boldsymbol{w} \in \boldsymbol{S} \odot \boldsymbol{T}$ if and only if for all factorizations $\boldsymbol{w} = \boldsymbol{w}_1\boldsymbol{w}_2$, $\boldsymbol{w}_1 \in \boldsymbol{S}$ or $\boldsymbol{w}_2 \in \boldsymbol{T}$.

We first show that $\odot$ is computable on principal filters, then we show how to complement ideals.

Lemma 4.19. *On principal filters, $\odot$ can be computed using the following equations:*

$$\uparrow \boldsymbol{v} \odot \uparrow \epsilon = \uparrow \epsilon \odot \uparrow \boldsymbol{v} = X^* \,, \tag{15}$$

$$\uparrow \boldsymbol{v}a \odot \uparrow b\boldsymbol{w} = \uparrow(\boldsymbol{v}ab\boldsymbol{w}) \cup (\uparrow \boldsymbol{v}) \cdot (\uparrow_X a \cap \uparrow_X b) \cdot (\uparrow \boldsymbol{w}) \,, \tag{16}$$

where $\boldsymbol{v}, \boldsymbol{w} \in X^*$ and $a, b \in X$.

Proof. Equation (15) is clear. We concentrate on Eq. (16):
($\supseteq$) If $\boldsymbol{u} \geq_* \boldsymbol{v}ab\boldsymbol{w}$, then for every factorization of $\boldsymbol{u} = \boldsymbol{u}_1\boldsymbol{u}_2$, the left factor $\boldsymbol{u}_1$ is above $\boldsymbol{v}a$, or the right factor $\boldsymbol{u}_2$ is above $b\boldsymbol{w}$, and thus $\boldsymbol{u} \in \uparrow \boldsymbol{v}a \odot \uparrow b\boldsymbol{w}$. If $\boldsymbol{u} \geq_* \boldsymbol{v}c\boldsymbol{w}$, where $c \in X$ is such that $c \geq a$ and $c \geq b$, then in every factorization of $\boldsymbol{u}$ as $\boldsymbol{u}_1\boldsymbol{u}_2$, c appears either in the left factor $\boldsymbol{u}_1$ or in the right factor $\boldsymbol{u}_2$, and this suffices to show that either $\boldsymbol{u} \geq_* \boldsymbol{v}a$ or $\boldsymbol{u} \geq_* b\boldsymbol{w}$.
($\subseteq$) Let $\boldsymbol{u} \in (\uparrow \boldsymbol{v}a) \odot (\uparrow b\boldsymbol{w})$. From the factorizations $\boldsymbol{u} = \boldsymbol{u} \cdot \epsilon$ and $\boldsymbol{u} = \epsilon \cdot \boldsymbol{u}$ we get $\boldsymbol{v}a \leq_* \boldsymbol{u}$ and $b\boldsymbol{w} \leq_* \boldsymbol{u}$. Consider the shortest prefix of $\boldsymbol{u}$ above $\boldsymbol{v}a$ and the shortest suffix above $b\boldsymbol{w}$. If these factors do not overlap, we get $\boldsymbol{u} \geq_* \boldsymbol{v}ab\boldsymbol{w}$. If they overlap, the

overlap must have length exactly one (otherwise $\boldsymbol{u}$ can be split anywhere in the middle of the overlap, contradicting the minimality of both factors). Write $\boldsymbol{u} = \boldsymbol{u}_1 c \boldsymbol{u}_2$ where $c \in X$ is the overlap. Then $\boldsymbol{u}_1 \geq_* \boldsymbol{v}$, $c \geq a$, $c \geq b$, and $\boldsymbol{u}_2 \geq_* \boldsymbol{w}$, which proves the statement.   □

Lemma 4.20 (CI). *Complementing ideals of* $(X^*, \leq_*)$ *is computable.*

Proof. Given an ideal $\boldsymbol{P} = \boldsymbol{A}_1 \cdots \boldsymbol{A}_n$, its complement is $\neg \boldsymbol{P} = \neg \boldsymbol{A}_1 \odot \cdots \odot \neg \boldsymbol{A}_n$. Using the procedure to complement downwards-closed sets of $(X, \leq)$, we can write each $\neg \boldsymbol{A}_i$ as a union of filters. Since $\odot$ distributes over unions of upwards-closed sets (from Lemma 4.10 by duality), we can write $\neg \boldsymbol{P}$ as a finite union of sets of the form $F_1 \odot F_2 \odot \cdots \odot F_n$, where the F_i's are filters. Finally, Lemma 4.19 allows us to reduce these expressions to a finite union of filters.   □

4.4.2 A proof of Theorem 4.11. One direction of the theorem is easy to check: products of atoms are indeed ideals (downwards-closed and directed) of $(X^*, \leq_*)$. For the other direction, consider an arbitrary ideal I of $(X^*, \leq_*)$. Its complement is upwards-closed, hence can be written $\neg I = \bigcup_{i<n} F_i$ for some filters $F_1, \ldots, F_n$. Therefore,

$$I = \neg \bigcup_{i<n} F_i = \bigcap_{i<n} \neg F_i.$$

Now, since any $\neg F_i$ is a finite union of products of atoms (see Lemma 4.14), by distributing the intersection over the unions, we are left with a finite union of finite intersections of products of atoms. Since these intersections can be decomposed as finite unions of products of atoms (see Lemma 4.16), we have decomposed the ideal I into a finite union of products of atoms. Since products of atoms are ideals (cf. first direction), and since ideals are prime subsets (by Proposition 2.10), we obtain that I is actually equal to one of those products of atoms (by Lemma 2.5).

   This proof highlights a general technique for identifying the ideals of some WQO: if we have some subclass $\mathcal{J}$ of the ideals such that the complement of any filter can be written as a finite union of ideals of $\mathcal{J}$, and the finite intersection of any two ideals of $\mathcal{J}$ can be written as a finite union of ideals of $\mathcal{J}$, then $\mathcal{J}$ is the class of all ideals.

4.4.3   Uniqueness of ideal representation. Writing ideals as products of atoms can be done in several ways. For example $D^* \cdot D^*$ and D^* coincide. They also coincide with $D^* \cdot (I + \epsilon)$ and $D^* \cdot D'^*$ if I, resp. D', are subsets of D^*.
   More generally, if $\boldsymbol{A}$ is an atom and $D \in Down(X)$ is such that $\boldsymbol{A} \subseteq D^*$, then $\boldsymbol{A} D^* = D^* \boldsymbol{A} = D^*$. Subsequently, we show that these are the only causes of non-uniqueness: avoiding such redundancies, every ideal has a unique representation as a product of atoms. (This was already observed for finite alphabets in [3].) This can be used to define a canonical representation for ideals of X^* (assuming one has defined a canonical representation for the ideals of X) and then for the downwards-closed sets. This representation is easy to use (moving from an arbitrary product of atoms to the

canonical representation just requires testing inclusions between atoms) and can lead to more efficient algorithms.

Below, we use letters such as $\mathtt{A}, \mathtt{P}$, etc., to denote sequences of atoms (syntax), and corresponding letters such as $\boldsymbol{A}$, $\boldsymbol{P}$, etc to denote the ideals obtained by taking the product (semantics). For example if $\mathtt{P} = (\mathtt{A}_1, \mathtt{A}_2, \cdots, \mathtt{A}_n)$, then $\boldsymbol{P} = \boldsymbol{A}_1 \cdot \boldsymbol{A}_2 \cdot \cdots \cdot \boldsymbol{A}_n$. Thus it is possible to have $\mathtt{P} \neq \mathtt{Q}$ and $\boldsymbol{P} = \boldsymbol{Q}$.

Definition 4.21. *A sequence of atoms* $\boldsymbol{A}_1, \cdots, \boldsymbol{A}_n$ *if said to be* reduced *if for every i, the following hold:*

- $\boldsymbol{A}_i \neq \emptyset^* = \{\epsilon\};$
- *if* $i < n$ *and* $\boldsymbol{A}_{i+1}$ *is some* D^*, *then* $\boldsymbol{A}_i \not\subseteq \boldsymbol{A}_{i+1};$
- *if* $i > 1$ *and* $\boldsymbol{A}_{i-1}$ *is some* D^*, *then* $\boldsymbol{A}_i \not\subseteq \boldsymbol{A}_{i-1}.$

Every ideal has a reduced decomposition into atoms, since any decomposition can be converted to a reduced one by dropping atoms which are redundant as per Definition 4.21. It remains to show that reduced representations are unique:

Theorem 4.22. *If* $\mathtt{P}$ *and* $\mathtt{Q}$ *are reduced sequences of atoms such that* $\boldsymbol{P} = \boldsymbol{Q}$, *then* $\mathtt{P} = \mathtt{Q}$.

Proof. By induction on $|\mathtt{P}| + |\mathtt{Q}|$. The result is trivial if $|\mathtt{P}| \leq 1$ or $|\mathtt{Q}| \leq 1$.

Otherwise, let us write $\mathtt{P}$ as $\mathtt{A}_1 \mathtt{A}_2 \mathtt{P}'$ and $\mathtt{Q}$ as $\mathtt{B}_1 \mathtt{B}_2 \mathtt{Q}'$.

We first show that $\boldsymbol{A}_1 = \boldsymbol{B}_1$. If $\boldsymbol{A}_1 \not\subseteq \boldsymbol{B}_1$, since $\boldsymbol{P} \subseteq \boldsymbol{Q}$, the inclusion test described in Lemma 4.13 gives us $\boldsymbol{P} \subseteq \boldsymbol{B}_2 \boldsymbol{Q}' \subseteq \boldsymbol{B}_1 \boldsymbol{B}_2 \boldsymbol{Q}' \subseteq \boldsymbol{P}$, so $\boldsymbol{P} = \boldsymbol{B}_2 \boldsymbol{Q}'$, and the induction hypothesis then yields $\mathtt{P} = \mathtt{B}_2 \mathtt{Q}'$. Similarly, if $\boldsymbol{B}_1 \not\subseteq \boldsymbol{A}_1$, then $\mathtt{Q} = \mathtt{A}_2 \mathtt{P}'$. In particular, $\boldsymbol{A}_1$ and $\boldsymbol{B}_1$ cannot be incomparable, otherwise $\mathtt{Q} = \mathtt{B}_1 \mathtt{B}_2 \mathtt{Q}' = \mathtt{B}_1 \mathtt{P} = \mathtt{B}_1 \mathtt{A}_1 \mathtt{A}_2 \mathtt{P}' = \mathtt{B}_1 \mathtt{A}_1 \mathtt{Q}$, which is impossible since $\mathtt{Q}$ and $\mathtt{B}_1 \mathtt{A}_1 \mathtt{Q}$ do not even have the same length. Hence, $\boldsymbol{B}_1$ must contain or be contained in $\boldsymbol{A}_1$. Without loss of generality, we assume $\boldsymbol{B}_1 \subseteq \boldsymbol{A}_1$. For the sake of contradiction, let us assume that $\boldsymbol{A}_1 \not\subseteq \boldsymbol{B}_1$. In that case, we have seen that $\mathtt{P} = \mathtt{B}_2 \mathtt{Q}'$, which in particular implies $\boldsymbol{A}_1 = \boldsymbol{B}_2$. Since $\boldsymbol{B}_1 \subseteq \boldsymbol{A}_1 = \boldsymbol{B}_2$, and $\mathtt{Q}$ is reduced, $\mathtt{B}_2$ cannot be of the form D^*, hence is equal to $I_2 + \epsilon$ for some ideal I_2 of X. Since $\boldsymbol{B}_1 \subseteq \boldsymbol{B}_2$, $\mathtt{B}_1$ cannot be of the form D^* either, and must therefore be equal to $I_1 + \epsilon$ for some ideal I_1 of X. Moreover, $I_1 \subseteq I_2$. We apply Lemma 4.13 to the inclusion of $\boldsymbol{Q} = \boldsymbol{B}_1 \boldsymbol{B}_2 \boldsymbol{Q}'$ in $\boldsymbol{P} = \boldsymbol{A}_1 \boldsymbol{A}_2 \boldsymbol{P}' = \boldsymbol{B}_2 \boldsymbol{A}_2 \boldsymbol{P}'$ and we obtain that $\boldsymbol{B}_2 \boldsymbol{Q}' \subseteq \boldsymbol{A}_2 \boldsymbol{P}'$. From the inclusions $\boldsymbol{B}_2 \boldsymbol{Q}' \subseteq \boldsymbol{A}_2 \boldsymbol{P}' \subseteq \boldsymbol{P} = \boldsymbol{B}_2 \boldsymbol{Q}'$, we deduce that $\boldsymbol{A}_2 \boldsymbol{P}' = \boldsymbol{B}_2 \boldsymbol{Q}'$, hence by induction hypothesis $\mathtt{A}_2 \mathtt{P}' = \mathtt{B}_2 \mathtt{Q}'$. This is impossible since $\mathtt{P} = \mathtt{B}_2 \mathtt{Q}'$, which implies that $\mathtt{P} = \mathtt{A}_2 \mathtt{P}'$, which contradicts $\mathtt{P} = \mathtt{A}_1 \mathtt{A}_2 \mathtt{P}'$.

We now want to show that $\boldsymbol{A}_2 \boldsymbol{P}' = \boldsymbol{B}_2 \boldsymbol{Q}'$. Since the situation is symmetric, we only prove one inclusion. We distinguish two cases.

1. If $\boldsymbol{A}_1$ is of the form $I + \epsilon$, then so is $\boldsymbol{B}_1$ and Lemma 4.13 directly implies $\boldsymbol{A}_2 \boldsymbol{P}' \subseteq \boldsymbol{B}_2 \boldsymbol{Q}'$.
2. Otherwise, $\boldsymbol{A}_1$ is of the form D^*, in which case $\boldsymbol{A}_2 \not\subseteq \boldsymbol{A}_1 = \boldsymbol{B}_1$, because $\mathtt{P}$ is reduced. Therefore, by Lemma 4.13, $\boldsymbol{A}_2 \boldsymbol{P}' \subseteq \boldsymbol{B}_2 \boldsymbol{Q}'$.

By symmetry, we obtain $\boldsymbol{A}_2 \boldsymbol{P}' = \boldsymbol{B}_2 \boldsymbol{Q}'$ and conclude the proof with the induction hypothesis. $\square$

4.5   Finitary powersets

Given a QO $(X, \leq)$, we write $\mathcal{P}(X)$ to denote its powerset, with typical elements $S, T,$ ... A usual way of extending the quasi-ordering between elements of X into a quasi-ordering between sets of such elements is the *Hoare quasi-ordering* (also called *domination quasi-ordering*), denoted $\sqsubseteq_H$, and defined by

$$S \sqsubseteq_H T \overset{\text{def}}{\Leftrightarrow} \forall x \in S : \exists y \in T : x \leq y.$$

A convenient characterization of this ordering is the following: $S \sqsubseteq_H T$ iff $S \subseteq \downarrow_X T$. Note that $(\mathcal{P}(X), \sqsubseteq_H)$ is in general not antisymmetric even when $(X, \leq)$ is. For example, and writing $\equiv_H$ to denote $\sqsubseteq_H \cap \sqsupseteq_H$, the above characterization implies that $S \equiv_H \downarrow_X S$ for any $S \subseteq X$. In particular, this shows that the quotient $\mathcal{P}(X)/\equiv_H$ is isomorphic to $(Down(X), \subseteq)$. While $(Down(X), \subseteq)$ is well-founded if, and only if, $(X, \leq)$ is a WQO (cf. Lemma 2.3), this does not guarantee that $(Down(X), \subseteq)$ is a WQO, as famously shown by Rado [49].[8] In other words, powerset is not a WQO-preserving construction.

However, the *finitary powerset* construction is WQO-preserving. Let $\mathcal{P}_f(X)$, sometimes also written $[X]^{<\omega}$, denote the set of all *finite subsets* of X.

Theorem 4.23. $(\mathcal{P}_f(X), \sqsubseteq_H)$ *is WQO if, and only if,* $(X, \leq)$ *is WQO.*

The if direction is an easy consequence of Higman's Lemma: the function that maps each word in X^* to its set of letters, in $\mathcal{P}_f(X)$, is monotonic and surjective, and the image of a WQO by any monotonic map is WQO. We shall see another proof in Section 5.2.

Proposition 4.24 (Ideals of $\mathcal{P}_f(X)$). *Given a WQO $(X, \leq)$, the ideals of $(\mathcal{P}_f(X), \sqsubseteq_H)$ are exactly the sets $\mathcal{J}$ of the form $\mathcal{J} = \mathcal{P}_f(D)$ for $D \in Down(X)$.*

Proof. $(\Longleftarrow)$: $\emptyset \in \mathcal{P}_f(D)$, so $\mathcal{P}_f(D)$ is non-empty. It is downwards-closed, since if $S \sqsubseteq_H T \in \mathcal{P}_f(D)$, then $S \subseteq \downarrow_X T \subseteq \downarrow_X D = D$. It is directed, since if $S, T \in \mathcal{P}_f(D)$, then $S \cup T \in \mathcal{P}_f(D)$, and $S, T \sqsubseteq_H S \cup T$.

$(\Longrightarrow)$: Let $\mathcal{J}$ be an ideal of $\mathcal{P}_f(X)$ and let $D = \bigcup_{S \in \mathcal{J}} S$, so that $\mathcal{J} \subseteq \mathcal{P}_f(D)$. Since $\mathcal{J}$ is downwards-closed under $\sqsubseteq_H$, D is downwards-closed under $\leq$ and $\{x\} \in \mathcal{J}$ for all $x \in D$. Since $\mathcal{J}$, being an ideal, is non-empty, $\emptyset \in \mathcal{J}$. Finally, if $S, T \in \mathcal{J}$, then there is some $U \in \mathcal{J}$ such that $S, T \sqsubseteq_H U$. Thus $S \cup T \sqsubseteq_H U$, and therefore $S \cup T \in \mathcal{J}$. Therefore, $\mathcal{J}$ contains the empty set, all the singletons included in D, is closed under finite unions, and so is equal to $\mathcal{P}_f(D)$. $\qquad\square$

When $(X, \leq)$ is ideally effective, finite subsets of X can be represented using any of the usual data structures and Proposition 4.24 directly leads to a data structure for $Idl(\mathcal{P}_f(X))$ inherited from the representation of X's ideals and downwards-closed sets.

Theorem 4.25. *With the above representations, the finitary powerset with Hoare's ordering is an ideally effective constructor.*

---

[8] In fact $(\mathcal{P}(X), \sqsubseteq_H)$ is a WQO iff X is an ω^2-WQO [47,33].

Proof. Let $(X, \leq)$ be an ideally effective WQO. In the following we use shorthand notations such as $\downarrow_H$ for $\downarrow_{\mathcal{P}_f(X)}$, etc., with the obvious meaning.

(OD): The sets we consider being finite, the definition of $\sqsubseteq_H$ leads to an obvious implementation.

(ID): Testing inclusion in $Idl(\mathcal{P}_f(X))$ reduces to testing inclusion in $Down(X)$ by
$$\mathcal{J}_1 = \mathcal{P}_f(D_1) \subseteq \mathcal{J}_2 = \mathcal{P}_f(D_2) \iff D_1 \subseteq D_2.$$

(PI): Given S a finite subset of X, the principal ideal $\downarrow_H S$ is $\mathcal{P}_f(\downarrow_X S)$ so we just need to compute the downwards-closed $\downarrow_X S = \bigcup_{x \in S} \downarrow x$ in X's representation.

(CF): Given $S \in \mathcal{P}_f(X)$, the complement of $\uparrow_H S$ can be given an ideal decomposition via

$$\mathcal{P}_f(X) \smallsetminus \uparrow_H S = \bigcup_{x \in S} \mathcal{P}_f(X) \smallsetminus \uparrow_H \{x\} = \bigcup_{x \in S} \mathcal{P}_f(X \smallsetminus \uparrow x)$$

This can now be computed using (CF) for X.

(II): We have $\mathcal{P}_f(D_1) \cap \mathcal{P}_f(D_2) = \mathcal{P}_f(D_1 \cap D_2)$.

(IF): Filters may be intersected using $\uparrow S \cap \uparrow T = \uparrow(S \cup T)$.

(CI): Given an ideal $\mathcal{J} = \mathcal{P}_f(D)$, $\mathcal{P}_f(X) \smallsetminus \mathcal{J}$ consists of the sets that contain at least one element not in D. That is:

$$\neg \mathcal{J} = \uparrow_H \{x_1\} \cup \cdots \cup \uparrow_H \{x_n\} \text{ if } X \smallsetminus D = \uparrow_X x_1 \cup \cdots \cup \uparrow_X x_n,$$

which is computable using (CI) for X.

The above proves that the powerset constructor is an ideally effective constructor. Once again, the computability of the presentation described above from a presentation of $(X, \leq)$ is clear. For (XI) and (XF), observe that $\mathcal{P}_f(X)$ is its own ideal decomposition since $X \in Down(X)$, and the filter decomposition of $\mathcal{P}_f(X)$ is $\uparrow_H \emptyset$. □

5   More constructions on ideally effective WQOs

In this section we describe more constructions that yield new ideally effective WQOs from previously defined ones. By contrast with the constructors of Section 4.1, these constructions take some extra parameters that are not WQOs—for example, an equivalence relation in order to build quotient WQOs (see Section 5.2). Showing that the quotient WQO is ideally effective will need some effectiveness assumptions on the equivalence at hand, in the spirit of what we did with the one-sorted constructors.

5.1   Order extension

Let $(X, \leq)$ be a WQO and let $\leq'$ be an extension of $\leq$ (i.e., $\leq \subseteq \leq'$). Then $(X, \leq')$ is also a WQO. In this subsection, we investigate the ideals of $(X, \leq')$ and present some sufficient condition for $(X, \leq')$ to be ideally effective, assuming $(X, \leq)$ is. In the next subsections, we will present natural applications of this framework.

Proposition 5.1. *Given a WQO $(X, \leq)$ and an extension $\leq'$ of $\leq$, the ideals of $(X, \leq')$ are exactly the downward closures under $\leq'$ of the ideals of $(X, \leq)$. That is,*

$$Idl(X, \leq') = \{\downarrow_{\leq'} I \mid I \in Idl(X, \leq)\}.$$

Proof. ($\supseteq$) Let $I \in Idl(X, \leq)$. Even though I may not be downwards-closed in $(X, \leq')$, it is still directed. It is easy to see that $\downarrow_{\leq'} I$ is directed, non-empty, and downwards-closed for $\leq'$. Thus it is an ideal of $(X, \leq')$.

($\subseteq$) Let J be an ideal of $(X, \leq')$. J may not be directed in $(X, \leq)$, but it is still downwards-closed under $\leq$. As a consequence, it can be decomposed as a finite union of ideals of $(X, \leq)$: $J = I_1 \cup \cdots \cup I_n$. Then $J = \downarrow_{\leq'} J = \downarrow_{\leq'} I_1 \cup \cdots \cup \downarrow_{\leq'} I_n$. Now applying Lemma 2.5 to $(X, \leq')$, we have $J = \downarrow_{\leq'} I_i$ for some i. □

Assume that $(X, \leq)$ is an ideally effective WQO for which we have a presentation at hand, in particular data structures for X and $Idl(X)$. Let $\leq'$ be an extension of $\leq$. To represent elements of $(X, \leq')$, it is natural to use the same data structure for X than the one used for $(X, \leq)$. For ideals, Proposition 5.1 suggests to also use the same data structure as the one for ideals of X. That is, an ideal $J \in Idl(X, \leq')$ will actually be represented by any $I \in Idl(X)$ such that $J = \downarrow_{\leq'} I$.

Using these representations for $(X, \leq')$ does not always lend itself to algorithms that would witness ideal effectiveness, even under the assumptions that $(X, \leq)$ is ideally effective and that $\leq'$ is decidable. There is even is a "natural" counter example: the lexicographic ordering over $X \times X$ (see Section 6.2). This fact justifies that we make further assumptions. More precisely, we show that $(X, \leq')$ is ideally effective if we can compute downward closures under $\leq'$:

Theorem 5.2. *Let $(X, \leq)$ be an ideally effective WQO and $\leq'$ an extension of $\leq$. Then, $(X, \leq')$ is ideally effective for the aforementioned data structures of X and $Idl(X, \leq')$, whenever the following functions are computable:*

$$\mathcal{C}l_\mathrm{I} : Idl(X, \leq) \rightarrow Down(X, \leq) \qquad \mathcal{C}l_\mathrm{F} : Fil(X, \leq) \rightarrow Up(X, \leq)$$
$$I \mapsto \downarrow_{\leq'} I \qquad\qquad\qquad \uparrow x \mapsto \uparrow_{\leq'}(\uparrow x) = \uparrow_{\leq'} x$$

Moreover, under these assumptions, a presentation of $(X, \leq')$ can be computed uniformly from a presentation of $(X, \leq)$ and algorithms realizing $\mathcal{C}l_\mathrm{I}$ and $\mathcal{C}l_\mathrm{F}$.

Note that if $I \in Idl(X, \leq)$, then $\downarrow_{\leq'} I$ is also downwards-closed for $\leq$ and thus can be represented as a downwards-closed set of $(X, \leq)$. This is precisely this representation that the function $\mathcal{C}l_\mathrm{I}$ outputs. Same goes for $\mathcal{C}l_\mathrm{F}$. Note that using functions $\mathcal{C}l_\mathrm{I}$ and $\mathcal{C}l_\mathrm{F}$, it is possible to compute the downward and upward closure under $\leq'$ of arbitrary downwards- and upwards-closed sets for $\leq$ using the canonical decompositions: $\downarrow_{\leq'}(I_1 \cup \cdots \cup I_n) = (\downarrow_{\leq'} I_1) \cup \cdots \cup (\downarrow_{\leq'} I_n)$ and $\uparrow_{\leq'}(\uparrow x_1 \cup \cdots \cup \uparrow x_n) = \uparrow_{\leq'} x \cup \cdots \cup \uparrow_{\leq'} x_n$.

Proof. We proceed to show that $(X, \leq')$ is ideally effective.

(OD): One can tests $x \leq' y$, since this is equivalent to $y \in \mathcal{C}l_\mathrm{F}(\uparrow_\leq x)$.

(ID): Ideal inclusion can be decided using $\mathcal{C}l_\mathrm{I}$ and the inclusion test for downwards-closed sets of $(X, \leq)$: $\downarrow_{\leq'} I_1 \subseteq \downarrow_{\leq'} I_2 \iff I_1 \subseteq \mathcal{C}l_\mathrm{I} I_2$.

(PI): The principal ideal $\downarrow_{\leq'} x$ of $(X, \leq')$ is represented by $\downarrow_{\leq} x$, since $\downarrow_{\leq'}(\downarrow_{\leq} x) = \downarrow_{\leq'} x$.

(CF): For $x \in X$, the filter complement $X \smallsetminus \uparrow_{\leq'} x$ is $X \smallsetminus \mathcal{C}l_F(\uparrow_{\leq} x)$ which can be computed, using (CF) and (II) for $(X, \leq)$, as a downwards-closed set in $(X, \leq)$. This is represented by an ideal decomposition $D = \bigcup_{i<n} I_i$ which is canonical in $(X, \leq)$ but not necessarily in $(X, \leq')$ since one may have $\downarrow_{\leq'} I_i \subseteq \downarrow_{\leq'} I_j$ for $i \neq j$. However, extracting the canonical ideal decomposition wrt. $\leq'$ can be done using (ID) for $(X, \leq')$.

(II): Intersection of ideals is computed with $\downarrow_{\leq'} I_1 \cap \downarrow_{\leq'} I_2 = \mathcal{C}l_I(I_1) \cap \mathcal{C}l_I(I_2)$. Here again, this result in an ideal decomposition that is canonical for $\leq$ but not for $\leq'$ until we process it as done for (CF).

(CI), (CF)   : these operations are obtained similarly.

With algorithms for the closure functions $\mathcal{C}l_I$ and $\mathcal{C}l_F$, the presentation above is computable from a presentation of $(X, \leq)$. Regarding (XF) and (XI), we note that filter and ideal decompositions of X for $\leq$ are also valid decompositions for $\leq'$. However, these decompositions might not be canonical for $\leq'$ even if they are for $\leq$, in which case the canonical decompositions can be obtained using (OD) and (ID), as usual. $\square$

5.1.1 Sequences under stuttering. In this subsection, we apply Theorem 5.2 to an extension of Higman's ordering $\leq_*$ on finite sequences (from Section 4.4).

Given a QO $(X, \leq)$, we define the *stuttering ordering* $\leq_{st}$ over X^* by $\boldsymbol{x} = x_1 \cdots x_n \leq_{st} \boldsymbol{y} = y_1 \cdots y_m \overset{\text{def}}{\Leftrightarrow}$ there are n indices $1 \leq p_1 \leq p_2 \leq \cdots \leq p_n \leq m$ such that $x_i \leq y_{p_i}$ for all $i = 1, \ldots, n$. Compared with Higman's ordering, the sequence of positions $(p_i)_{i=1,\ldots,n}$ in $\boldsymbol{y}$ need not be *strictly* increasing: repetitions are allowed. For instance, if $X = \{a, b\}$ is a finite alphabet, then $aabbaa \leq_{st} aba \leq_{st} aabbaa$ but $aabbaa \nleq_{st} ab$. Or with $X = \mathbb{N}$, $(1, 1, 1) \leq_{st} (2)$. Note that even when $(X, \leq)$ is antisymmetric, $(X^*, \leq_{st})$ need not be.

Remark 5.3. There is another way to define the stuttering ordering: define the stuttering equivalence relation $\sim_{st}$ on X^* as the smallest equivalence relation such that for all $\boldsymbol{x}, \boldsymbol{y} \in X^*$ and $a \in X$, $\boldsymbol{x}a\boldsymbol{y} \sim_{st} \boldsymbol{x}aa\boldsymbol{y}$. Informally, this equivalence does not distinguish between a single and several consecutive occurrences of a same element. Then, $\leq_{st} = \leq_* \circ \sim_{st}$, where $\circ$ denotes the composition of relations. Observe that $\sim_{st}$ is *not* the same as the equivalence relation $\equiv_{st} = \leq_{st} \cap \geq_{st}$ induced by the ordering, even if $(X, \leq)$ is a partial order $\leq$. For instance, if $a \leq b$ in X, then $ab \equiv_{st} b$ in X^*, but $ab \sim_{st} b$ does not hold. However the inclusion $\sim_{st} \subseteq \equiv_{st}$ is always valid. $\square$

Obviously, $\leq_{st}$ is an order extension of $\leq_*$, thus $(X^*, \leq_{st})$ is a WQO when $(X, \leq)$ is, and we can apply Theorem 5.2.

Theorem 5.4. *The stuttering extension of a WQO $(X, \leq)$ is an ideally effective constructor.*

Proof. In the light of Theorems 4.12 and 5.2, it suffices to show that the following closure functions are computable:

$$\mathcal{C}l_I : Idl(X^*, \leq_*) \to Down(X^*, \leq_*) \qquad \mathcal{C}l_F : Fil(X^*, \leq_*) \to Up(X^*, \leq_*)$$
$$I \mapsto \downarrow_{st} I \qquad\qquad\qquad \uparrow\boldsymbol{u} \mapsto \uparrow_{st}(\uparrow\boldsymbol{u}) = \uparrow_{st}\boldsymbol{u}$$

Recall from Section 5.1 that the ideals of $(X^*, \leq_*)$ are the (concatenation) products of atoms, where atoms are either of the form D^* for some $D \in Down(X)$ or $I + \epsilon$ for some $I \in Idl(X)$. It is quite immediate to see that $Cl_I(D^*) = D^*$ and $Cl_I(I + \epsilon) = I^*$, and that given two products of atoms $\boldsymbol{P}_1, \boldsymbol{P}_2, Cl_I(\boldsymbol{P}_1 \cdot \boldsymbol{P}_2) = Cl_I(\boldsymbol{P}_1) \cdot Cl_I(\boldsymbol{P}_2)$. From these equations, it is simple to write an inductive algorithm computing Cl_I.

Function Cl_F is computable as well, although less straightforward. We provide an expression for Cl_F in Lemma 5.5 which is clearly computable. $\qquad \square$

Lemma 5.5. *Given* $\boldsymbol{u} = x_1 \cdots x_n \in X^*$,

$$
Cl_F(\boldsymbol{u}) = \uparrow_{st} \boldsymbol{u} = \uparrow_* \left\{ y_1 \cdots y_k \;\middle|\; \begin{array}{l} 0 \leq k \leq n \\ 0 = \ell_0 < \ell_1 < \cdots < \ell_k = n \\ \forall j = 1, \ldots, k : y_j \in \min(\bigcap_{\ell_{j-1} < \ell \leq \ell_j} \uparrow_X x_\ell) \end{array} \right\},
$$

where $\min(A)$ *denotes a finite basis of the upwards-closed subset* A.

(Intuitively, the set ranges over all ways to cut $\boldsymbol{u}$ *in* k *consecutive pieces, and embeds all elements of the* j-*th piece into the same element* y_j. *It has long sequences, the longest being* $\boldsymbol{u}$, *and shorter ones with potentially larger elements.)*

This is the fully generic formula to describe the function Cl_F for any X. However, in simple cases, $Cl_F(\boldsymbol{w})$ takes a much simpler form. For instance, for $X = \mathbb{N}$, we have $Cl_F(x_1 \cdots x_n) = \uparrow_* \max(x_1, \cdots, x_n)$, and for $X = \Sigma$ a finite alphabet, $Cl_F(\boldsymbol{w}) = \uparrow_* \boldsymbol{v}$ where $\boldsymbol{v}$ is the shortest member of the equivalence class $[\boldsymbol{w}]_{\sim_{st}}$ (that is, $\boldsymbol{v}$ is obtained from $\boldsymbol{w}$ by fusing consecutive equal letters).

Proof (Of Lemma 5.5). The "$\supseteq$" direction is obvious.
($\subseteq$) Given $\boldsymbol{w} \geq_{st} x_1 \cdots x_n$, there exist a monotonic mapping p from $\{1, \cdots, n\}$ to $\{1, \cdots, m\}$, where $m = |\boldsymbol{w}|$, and such that each x_i is associated to an element in $\boldsymbol{w}$ that is larger than or equal to x_i. Denoting the image of p by $\{y_1, \ldots, y_k\}$, this entails a decomposition of $\boldsymbol{w} = \boldsymbol{w}_0 y_1 \boldsymbol{w}_1 y_2 \cdots y_k \boldsymbol{w}_k$ where the y_i's are in X and the $\boldsymbol{w}_i$'s in X^*. Further define i_j to be the largest i such that $p(i) = j$ (i.e., the index of the right-most symbol of $x_1 \cdots x_n$ to be mapped to y_j). It follows that $0 = i_0 < i_1 < \cdots < i_k = n$, and for all $\ell \in [n]$ and $j \in [k]$, $i_{j-1} < \ell \leq i_j \Rightarrow x_\ell \leq y_j$. Then $\boldsymbol{w} \geq_* y_1 \cdots y_k$ which is indeed an element of the set described in the proposition. $\qquad \square$

5.2 Quotienting under a compatible equivalence

In this subsection, we apply the results of Section 5.1 to the most commonly encountered case of order-extension: quotient under an equivalence relation.

Let $(X, \leq)$ be a WQO and let E be an equivalence relation on X which is *compatible* with $\leq$ in the sense that $\leq \circ E = E \circ \leq$, where $\circ$ denotes the composition of relations. Define the relation $\leq_E$ on X to be $\leq \circ E$. Then $\leq_E$ is clearly reflexive, and is transitive since

$$
\leq_E \circ \leq_E = (\leq \circ E) \circ (\leq \circ E) = \leq \circ (E \circ \leq) \circ E = \leq \circ (\leq \circ E) \circ E = \leq \circ E = \leq_E
$$

In this subsection, we give sufficient conditions for $(X, \leq_E)$ to be ideally effective, provided $(X, \leq)$ is.

Remark 5.6. Note that stuttering from Section 5.1.1 is *not* an example: Although $\leq_{\text{st}} = \leq_* \circ \sim_{\text{st}}$, the other condition does not hold: $\leq_{\text{st}} \neq \sim_{\text{st}} \circ \leq_*$. For instance, consider $X = \mathbb{N}^2$ where $\langle 1, 2 \rangle \langle 2, 1 \rangle \leq_{\text{st}} \langle 2, 2 \rangle$. However, if X is a finite alphabet, the equality $\leq_{\text{st}} = \sim_{\text{st}} \circ \leq_*$ holds and $(X^*, \leq_{\text{st}})$ can be treated as a quotient. As another example, the finitary powerset $\mathcal{P}_f(X)$ from Section 4.5 can be obtained as a quotient of $(X^*, \leq_*)$, and could be shown ideally effective using Theorem 5.7 below. However, and because operations in $\mathcal{P}_f(X)$ are quite simple, and because powerset is a fundamental constructor, we decided to provide a direct, more concrete, construction.   □

Observe that $\leq_E$ is an extension of $\leq$, and thus results on quotients can be seen as an application of Section 5.1. However, since quotients are of such importance in computer science (and used more often than mere extensions), we reformulate Theorem 5.2 in this specific context: functions $\mathcal{C}l_{\text{I}}$ and $\mathcal{C}l_{\text{F}}$ take an interesting form. As in the case of extensions, elements and ideals of $(X, \leq_E)$ will be represented using the data structures coming from a presentation of $(X, \leq)$.

Theorem 5.7. *Let $(X, \leq)$ be an ideally effective WQO and E be an equivalence relation on X compatible with $\leq$. Then, $(X, \leq_E)$ is ideally effective for the aforementioned data structures of X and $Idl(X, \leq_E)$, whenever the following functions are computable:*

$$\mathcal{C}l_{\text{I}} : Idl(X, \leq) \to Down(X, \leq) \qquad \mathcal{C}l_{\text{F}} : Fil(X, \leq) \to Up(X, \leq)$$
$$I \mapsto \overline{I} \qquad\qquad\qquad \uparrow x \mapsto \uparrow \overline{x}$$

where, given $S \subseteq X$, $\overline{S}$ denotes the closure under E of S, i.e., $\overline{S} \overset{def}{=} \{y \mid \exists x \in S : xEy\}$. In particular, $\overline{x}$ simply is the equivalence class of x.

Moreover, under these assumptions, we can compute a presentation of $(X, \leq_E)$ from a presentation of $(X, \leq)$.

Proof. In the light of Theorem 5.2, it suffices to show that $\downarrow_{\leq_E} F = \overline{F}$ and $\downarrow_{\leq_E} I = \overline{I}$ for any filter F and any ideal I of $(X, \leq)$. The first equality follows from $\leq_E = \leq \circ E$ while the second comes from $\leq_E = E \circ \leq$. This is why we introduced the compatibility condition $\leq \circ E = E \circ \leq$.   □

In particular, we see that the ideals of $(X, \leq_E)$ are exactly the closures under E of the ideals of $(X, \leq)$. That is, $Idl(X, \leq_E) = \left\{ \overline{I} : I \in Idl(X, \leq) \right\}$.

We conclude this section with two results that are specific to WQOs obtained by quotienting, and which lead to simplifications in several algorithms.

Proposition 5.8. *Let J be an ideal under $\leq_E$, and let $J = I_1 \cup \cdots \cup I_k$ be the canonical ideal decomposition of J under $\leq$. Then $J = \overline{I_i}$ for every i.*

Proof. Recall from the proof of Proposition 5.1 that $J = \overline{I_i}$ for some i. Without loss of generality, we can assume $i = 1$. For the sake of contradiction, suppose that there exists some i such that $J \neq \overline{I_i}$. Again without loss of generality, we can assume $i = k$. From $\overline{I_1} \neq \overline{I_k}$, we deduce that there exists $x \in I_1$ which has no E-equivalent in I_k.

We will now show that $J \subseteq I_1 \cup \cdots \cup I_{k-1}$, which will be a contradiction since we assumed that we started from a canonical ideal decomposition. Let $y \in J$. Then there

exists a $y' \in I_1$ such that $y \mathrel{E} y'$. Since I_1 is an ideal under $\leq$, there is a $z \in I_1$ such that $x \leq z$ and $y' \leq z$. We have $y \mathrel{E} y' \leq z$, thus there exists z' such that $y \leq z' \mathrel{E} z$. Since J is closed under E-equivalence, $z' \in J$, hence $z' \in I_i$ for some i. However, z' cannot belong to I_k, since $x \leq_E z'$ and the E-equivalence class of x is disjoint from I_k. So $z' \in I_1 \cup \cdots \cup I_{k-1}$, and hence $y \in I_1 \cup \cdots \cup I_{k-1}$. Thus $J = I_1 \cup \cdots \cup I_{k-1}$, and we have a contradiction. □

Proposition 5.9. *For any two ideals $I_1, I_2 \in Idl(X, \leq)$, $\overline{I_1} \cap \overline{I_2} = \overline{I_1 \cap \overline{I_2}} = \overline{\overline{I_1} \cap I_2}$. For any two filters $F_1, F_2 \in Fil(X, \leq)$, $\overline{F_1} \cap \overline{F_2} = \overline{F_1 \cap \overline{F_2}} = \overline{\overline{F_1} \cap F_2}$.*

Thus, computing intersections for (II) (resp., (IF)) only requires one invocation of $\mathcal{Cl}_I$ (resp., $\mathcal{Cl}_F$) instead of two in the algorithm described in the proof of Theorem 5.2.

Proof. We show $\overline{I_1} \cap \overline{I_2} = \overline{I_1 \cap \overline{I_2}}$, the other equality is symmetric. For the right-to-left inclusion, we have $I_1 \cap \overline{I_2} \subseteq \overline{I_1} \cap \overline{I_2}$, and closing both sides under E gives the required result. For the left-to-right inclusion, let $x \in \overline{I_1} \cap \overline{I_2}$. Then there exist $x_1 \in I_1$ and $x_2 \in I_2$ such that $x_1 \mathrel{E} x \mathrel{E} x_2$. Then $x_1 \in I_1 \cap \overline{I_2}$, and thus $x \in \overline{I_1 \cap \overline{I_2}}$.

The same proof applies to filters. □

5.2.1 Sequences under conjugacy.

Consider a WQO $(X, \leq)$, and define an equivalence relation $\sim_{\mathrm{cj}}$ on X^* as follows: $\boldsymbol{w} \sim_{\mathrm{cj}} \boldsymbol{v}$ iff there exist $\boldsymbol{x}, \boldsymbol{y}$ such that $\boldsymbol{w} = \boldsymbol{xy}$ and $\boldsymbol{v} = \boldsymbol{yx}$. One can imagine an equivalence class of $\sim_{\mathrm{cj}}$ as a sequence written on an (oriented) circle instead of a line. We can now define a notion of *subwords under conjugacy* via $\leq_{\mathrm{cj}} \stackrel{\mathrm{def}}{=} \sim_{\mathrm{cj}} \circ \leq_*$, which is exactly the relation denoted $\preceq_c$ in [3, p. 49].

Since $\sim_{\mathrm{cj}}$ is compatible with $\leq_*$, that is $\leq_* \circ \sim_{\mathrm{cj}} = \sim_{\mathrm{cj}} \circ \leq_*$, our results over quotients apply to $(X^*, \leq_{\mathrm{cj}})$.

Theorem 5.10. *Sequence extension with conjugacy is an ideally effective constructor.*

Proof. Note that the data structures used for elements and ideals of $(X^*, \leq_{\mathrm{cj}})$ are obtained from data structures for $(X^*, \leq_*)$ as done with Theorem 5.7.

In the light of Theorem 5.7, it suffices to show that we can compute closures under $\sim_{\mathrm{cj}}$ of elements and ideals of $(X^*, \leq_*)$. Given $\boldsymbol{w} \in X^*$, the equivalence class of $\boldsymbol{w}$ under $\sim_{\mathrm{cj}}$ is equal to $\overline{\boldsymbol{w}} = \{c^{(i)}(\boldsymbol{w}) \mid 0 \leq i < \max(1, |\boldsymbol{w}|)\}$, where $c^{(i)}$ denotes the i-th iterate of the cycle operator $c(w_1 \cdots w_n) = w_2 \cdots w_n w_1$, which corresponds to rotating the sequence i times. This expression is obviously computable.

Computing the closure under $\sim_{\mathrm{cj}}$ of ideals is quite similar. Remember that ideals of $(X^*, \leq_*)$ are products of atoms, where atoms are either of the form D^* for some $D \in Down(X)$, or of the form $I + \epsilon$, for some $I \in Idl(X)$. Then, given $\boldsymbol{P} = \boldsymbol{A}_1 \cdots \boldsymbol{A}_k$ an ideal of $(X^*, \leq_*)$:

$$\overline{\boldsymbol{P}} = \bigcup_{i=1}^{k} c^{(i)}(\boldsymbol{P}) \cdot e(\boldsymbol{A}_i)$$

where $e(D^*) = D^*$ and $e(I + \epsilon) = \epsilon$. The presence of the extra $e(A_i)$ in the above expression might become clearer when considering a simple example as $\boldsymbol{P} = \{a\}^* \{b\}^*$ where $\overline{\boldsymbol{P}} = \{a\}^* \{b\}^* \{a\}^* \cup \{b\}^* \{a\}^* \{b\}^*$. Indeed, $abba \sim_{\mathrm{cj}} baab \sim_{\mathrm{cj}} aabb \in \boldsymbol{P}$. □

5.2.2   Multisets under the embedding ordering. Given a WQO $(X, \leq)$, we consider the set $X^{\circledast}$ of finite multisets over X. Intuitively, multisets are sets where an element might occur multiple times. Formally, a multiset $M \in X^{\circledast}$ is a function from X to $\mathbb{N}$: $M(x)$ denotes the number of occurrences of x in M. The support of a multiset M denoted $Supp(M)$ is the set $\{x \in X \mid M(x) \neq 0\}$. A multiset is said to be finite if its support is.

A natural algorithmic representation for these objects are lists of elements of X, keeping in mind that a permutation of a list represents the same multiset. Formally, this means that $X^{\circledast}$ is the quotient of X^* by the equivalence relation $\sim$ defined by

$$\boldsymbol{u} = u_1 \cdots u_n \sim \boldsymbol{v} = v_1 \cdots v_m \overset{\text{def}}{\Leftrightarrow} n = m \wedge \exists \sigma \in S_n : u_i = v_{\sigma(i)} \text{ for all } i = 1, \ldots, n,$$

where S_n denotes the group of permutations over $\{1, \cdots, n\}$.

Once again, the equivalence relation $\sim$ is compatible with $\leq_*$. We denote by $\leq_{\text{emb}}$ the composition $\sim \circ \leq_* = \leq_* \circ \sim$, often called multiset embedding. (There exist other classical quasi-orderings on finite multisets, such as the domination quasi-ordering, aka the Dershowitz-Manna quasi-ordering [13]: see [31] for a proof that it is an ideally effective constructor.) For this section, we focus on $(X^{\circledast}, \leq_{\text{emb}})$, which is an application of our results on quotients.

Theorem 5.11. *Finite multisets with multiset embedding is an ideally effective constructor.*

Proof (Sketch). Note that the data structures used for elements and ideals of $(X^{\circledast}, \leq_{\text{emb}})$ are obtained from data structures for $(X^*, \leq_*)$ as done with Theorem 5.7.

In the light of Theorem 5.7, it suffices to show that we can compute closures under $\sim$ of elements and ideals of $(X^*, \leq_*)$. Given $\boldsymbol{w} = x_1 \cdots x_n \in X^*$, the equivalence class of $\boldsymbol{w}$ under $\sim$ simply consists of all the possible permutation of the word $\boldsymbol{w}$:

$$\overline{\boldsymbol{w}} = \bigcup_{\sigma \in S_n} x_{\sigma(1)} \cdots x_{\sigma(n)}$$

Closures of ideals is a little more complex. Let $\boldsymbol{P} = \boldsymbol{A}_1 \cdots \boldsymbol{A}_n$ be an ideal of $(X^*, \leq_*)$. Define:

$$D \overset{\text{def}}{=} \bigcup \{E \in Down(X) \mid \exists i \in \{1, \cdots, n\} : E^* = \boldsymbol{A}_i\}.$$

In other words, $D \in Down(X)$ is obtained from $\boldsymbol{P}$ by picking the atoms $\boldsymbol{A}_i$ that are of the second kind, $\boldsymbol{A}_i = E^*$, and taking the union of their generators. Similarly, let $I_1, \ldots, I_p$ be the ideals of $(X, \leq)$ that appear as $I_i + \epsilon$ in $\boldsymbol{P}$, with repetitions, and in order of occurrence. Then:

$$\overline{\boldsymbol{P}} = \bigcup_{\sigma \in S_p} D^* I_{\sigma(1)} D^* \cdots D^* I_{\sigma(p)} D^* .$$

$\square$

5.3   Induced WQOs

Let $(X, \leq)$ be a WQO. A subset Y of X (not necessarily finite) induces a quasi-ordering $(Y, \leq)$ which is also WQO.

Any subset $S \subseteq X$ induces a subset $Y \cap S$ in Y. Obviously, if S is upwards-closed (or downwards-closed) in X, then it induces an upwards-closed (resp., downwards-closed) subset in Y. However an ideal I or a filter F in X does not always induce an ideal or a filter in Y. In the other direction though, if $J \in Idl(Y)$, the downward closure $\downarrow_X J$ is an ideal of X. Therefore, to describe the ideals of Y, we need to identify those ideals of X that are of the form $\downarrow_X J$ for some ideal J of Y. This is captured by the following notion:

Definition 5.12. *Given a WQO $(X, \leq)$ and a subset Y of X, we say that an ideal $I \in Idl(X)$ is* in the adherence *of Y if $I = \downarrow_X (I \cap Y)$.*

In particular this implies that $I \subseteq \downarrow_X Y$ (we say that I is "below Y") and $I \cap Y \neq \emptyset$ (we say that I is "crossing Y"). The converse implication does not hold, as witnessed by $X = \mathbb{N}$, $Y = [1, 3] \cup [5, 7]$ and $I = \downarrow 4$.

We now show that the ideals of Y are exactly the subsets induced by ideals of X that are in the adherence of Y.

Theorem 5.13. *Let $(X, \leq)$ be a WQO and Y be a subset of X. A subset J of Y is an ideal of Y if and only if $J = I \cap Y$ for some $I \in Idl(X)$ in the adherence of Y. In this case, $I = \downarrow_X J$, and is thus uniquely determined from J.*

Proof. $(\Longrightarrow)$: If $J \in Idl(Y)$ then $I \stackrel{\text{def}}{=} \downarrow_X J$ is directed hence is an ideal of X. Clearly, $J = I \cap Y$, so I is in the adherence of Y.
$(\Longleftarrow)$: If $I \in Idl(X)$ is in the adherence of Y then $J \stackrel{\text{def}}{=} I \cap Y$ is non-empty (since I is crossing Y) and it is directed since for any $x, y \in J$ there is $z \in I$ above x and y, and $z \leq z'$ for some $z' \in J$ since I is below Y.

Uniqueness is clear since the compatibility assumption "$I = \downarrow_X (I \cap Y)$" completely determines I from the ideal $J = I \cap Y$ it induces.   □

An alternative definition of *adherence* often found in the literature (*e.g.* in [42]) is the following one: an ideal $I \in Idl(X)$ is in the adherence of Y if and only if there exists a directed subset $\Delta \subseteq Y$ such that $I = \downarrow_X \Delta$. The two definitions are equivalent, as we now argue, so that, notably, Theorem 5.13 extends Lemma 4.6 from [42].

Proof (that the two notions of adherence coincide).
$(\Longrightarrow)$: Assume $I = \downarrow_X (I \cap Y)$. We show that $\Delta = I \cap Y$ is directed: let $x, y \in \Delta \subseteq I$, since I is directed, there exists $z \in I$ such that $z \geq x, y$. But since $I = \downarrow_X \Delta$, there exists $z' \in \Delta$ such that $z' \geq z \geq x, y$, which proves that Δ is directed.
$(\Longleftarrow)$: Assume that there exists a directed subset $\Delta \subseteq Y$ such that $I = \downarrow_X \Delta$. Then $\downarrow_X (I \cap Y) = \downarrow_X (\downarrow_X \Delta \cap Y) = \downarrow_X (\Delta \cap Y) = \downarrow_X \Delta = I$.   □

Similarly, we can define a notion of adherence for filters. However, in this case, the condition $F = \uparrow_X (F \cap Y)$ simplifies: writing F as $\uparrow_X x$, this means that $x' \equiv_X x$ for some $x' \in Y$, in which case $F = \uparrow_X x'$. This is not surprising: $(Y, \leq)$ is a WQO, hence all its filters are principal.

5.3.1   Ideal effectiveness. Assuming that $(X, \leq)$ is an ideally effective WQO, and given $Y \subseteq X$, we can simply represent elements of Y by restricting the data structure for X to Y. This requires that Y is a recursive set. Alternatively, Theorem 5.13 suggests that we represent ideals of Y as ideals of X that are in the adherence of Y. This requires that we can decide membership in the adherence of Y. As in the case of extensions, the ideal effectiveness of $(Y, \leq)$ does not always follow from the ideal effectiveness of $(X, \leq)$ (see [31] for an example). We therefore have to introduce extra assumptions.

Theorem 5.14. *Let $(X, \leq)$ be a WQO and $Y \subseteq X$. Then $(Y, \leq)$ is ideally effective (for the aforementioned representations) provided:*

– *membership in Y is decidable over (the representation for) X,*
– *the following functions are computable:*

$$\mathcal{S}_\mathrm{I} : Idl(X, \leq) \to Down(X, \leq) \qquad \mathcal{S}_\mathrm{F} : Fil(X, \leq) \to Up(X, \leq)$$
$$I \mapsto \downarrow_X (I \cap Y) \qquad\qquad F \mapsto \uparrow_X (F \cap Y)$$

Moreover, in this case, a presentation of $(Y, \leq)$ can be computed from a presentation of $(X, \leq)$.

The rest of this subsection is dedicated to the proof of this theorem.

First, let us mention that our first assumption implies that we have a data structure for elements of Y and that thanks to function $\mathcal{S}_\mathrm{I}$, we can decide whether an ideal I of X is in the adherence of Y: it suffices to check that $\mathcal{S}_\mathrm{I}(I) = I$.

Let us prove that $(Y, \leq)$ is ideally effective.

(OD): since $\leq$ is decidable on X, its restriction to Y is still decidable.
(ID): Given two ideals I_1, I_2 that are in the adherence of Y, $I_1 \cap Y \subseteq I_2 \cap Y \iff I_1 \subseteq I_2$. The left-to-right implication uses that $I_i = \downarrow_X(I_i \cap Y)$. Therefore, inclusion for ideals of Y can be implemented by relying on (ID) for X.
(PI): if $y \in Y$, then $\downarrow_X y$ is adherent to Y and $\downarrow_X y \cap Y = \downarrow_Y y$.

For the four remaining operations, we need to be able to compute a representation of $D \cap Y$ and $U \cap Y$ for $D \in Down(X)$ and $U \in Up(X)$.

Lemma 5.15. *Let $D \in Down(X)$. The canonical representation of $D \cap Y$ (as a downwards-closed set of Y) is exactly the canonical representation of $\downarrow_X(D \cap Y)$ (as a downwards-closed set of X).*

Proof. Let $\bigcup_i I_i$ be the *canonical* decomposition of $\downarrow_X(D \cap Y)$. Remember that an ideal J of Y is represented by the unique ideal I of X which is in the adherence of Y such that $J = I \cap Y$. Thus, stating that $\bigcup_i I_i$ is the canonical representation of $D \cap Y$ means that:

1. $D \cap Y = \bigcup_i (I_i \cap Y)$;
2. for every i, $I_i \cap Y$ is an ideal of Y;
3. $I_i \cap Y$ and $I_j \cap Y$ are incomparable for inclusion, for $i \neq j$.

For the first point, $\bigcup_i (I_i \cap Y) = (\bigcup_i I_i) \cap Y = (\downarrow_X(D \cap Y)) \cap Y = D \cap Y$.

We now argue that each $I_i \cap Y$ is indeed an ideal of Y, i.e., all I_i's are in the adherence of Y. One inclusion being trivial, we need to show that $I_i \subseteq \downarrow_X(I_i \cap Y)$, for any i. Let $x_i \in I_i$. Since the ideals I_j are incomparable for inclusion, there exists $x_i' \in I_i$ such that $x_i \leq x_i'$ and for any $j \neq i$, $x_i' \notin I_j$ (I_i is directed). Besides, $x_i' \in I_i \subseteq \downarrow_X(D \cap Y)$ and thus there is an element x_i'' such that $x_i' \leq x_i'' \in D \cap Y$. As the sets I_j are downwards-closed, x_i'' cannot belong to any I_j with $j \neq 0$, hence x_i'' is in $I_i \cap Y$. Therefore, $x_i \in \downarrow_X(I_i \cap Y)$.

Finally, the ideal decomposition $D \cap Y = \bigcup_j (I_j \cap Y)$ is canonical since the I_j's are incomparable in X (recall the above criterion for inclusion of ideals of Y). □

Observe that if $D = \bigcup_i I_i$ then $\downarrow_X(D \cap Y) = \bigcup_i \downarrow_X(I_i \cap Y) = \bigcup_i \mathcal{S}_I(I)$. Thus the canonical representation of $D \cap Y$ is indeed computable from $D \in Down(X)$.

We now present the dual of the previous lemma:

Lemma 5.16. *Given $U \in Up(X)$, a canonical representation of $U \cap Y$ (as an upwards-closed set of Y) can be computed from a canonical representation of $\uparrow_X(U \cap Y)$ (as an upwards-closed set of X).*

Proof. Let $\bigcup_i \uparrow x_i$ be a canonical filter decomposition (in X) of the upwards-closed set $\uparrow_X(U \cap Y)$. We first prove that for every i, x_i is equivalent to some element of Y. Indeed, since $\uparrow_X x_i \subseteq \uparrow_X(U \cap Y)$, there exists $y \in U \cap Y$ with $y \leq x_i$. But then, y must be in some $\uparrow_X x_j$. Since the decomposition is canonical, the x_j's are incomparable, hence we cannot have $x_j \leq y \leq x_i$ for $j \neq i$. Thus, $x_i \equiv y \in Y$.

Moreover, we can compute a canonical filter decomposition of $\uparrow_X(U \cap Y)$ using only elements in Y: for each x_i, it is decidable whether $x_i \in Y$ (our first assumption on Y). If not, we can enumerate elements of Y until we find some $y_i \equiv x_i$. Such an element exists, and thus the enumeration terminates.

We thus obtain a canonical filter decomposition $\bigcup_i \uparrow y_i$ of $\downarrow_X(U \cap Y)$ with $y_i \in Y$. The rest of the proof is similar to the proof of Lemma 5.15. □

Here also, a canonical representation of $\uparrow_X(U \cap Y)$ is computable from U, using the function $\mathcal{S}_F$.

We can now describe procedures for the four remaining operations:

(CF): Given $y \in Y$, the complement of $\uparrow_Y y$ is computed by using the equality $Y \smallsetminus \uparrow_Y y = (X \smallsetminus \uparrow_X y) \cap Y$. The downwards-closed set $(X \smallsetminus \uparrow_X y)$ is computable using (CF) for X, and its intersection with Y is computable using Lemma 5.15.

(II): Given two ideals I and I' in the adherence of Y, the intersection of the ideals they induce is $(I \cap Y) \cap (I' \cap Y) = (I \cap I') \cap Y$, which is computable using (II) for X and Lemma 5.15.

(IF): Computing the intersection of filters is similar to computing the intersection of ideals: given $y_1, y_2 \in Y$, $(\uparrow_Y y_1) \cap (\uparrow_Y y_2) = (\uparrow_X y_1 \cap \uparrow_X y_2) \cap Y$, which is computable using (IF) for X and Lemma 5.16.

(CI): Given an ideal I in the adherence of Y, $Y \smallsetminus (I \cap Y) = (X \smallsetminus I) \cap Y$, which is computable using (CI) for X and Lemma 5.16.

Finally, and as always, the above presentation can be computed from a presentation of $(X, \leq)$, thanks to the functions $\mathcal{S}_{\mathrm{I}}$ and $\mathcal{S}_{\mathrm{F}}$. Notably, the ideal decomposition of Y can be computed as the subset induced by the downwards-closed set X of X, using Lemma 5.15, and the filter decomposition of Y can be computed as the subset induced by the upwards-closed set X of X, using Lemma 5.16.

Remark 5.17. If Y is a downwards-closed subset of X, then I is adherent to Y if and only if $I \subseteq Y$, and therefore $Idl(Y) = Idl(X) \cap \mathcal{P}(Y)$. Moreover, $\mathcal{S}_{\mathrm{I}}$ is computable thanks to (II), and $\mathcal{S}_{\mathrm{F}}(\uparrow x) = \uparrow x$ if $x \in Y$, $\mathcal{S}_{\mathrm{F}}(\uparrow x) = \emptyset$ otherwise. Indeed, if $x \notin Y$, then $\uparrow x \cap Y = \emptyset$.

Similarly, if Y is upwards-closed, $\mathcal{S}_{\mathrm{F}}$ can be computed with (II), and $\mathcal{S}_{\mathrm{I}}(I) = I$ if $Y \cap I \neq \emptyset$, $\mathcal{S}_{\mathrm{I}}(I) = \emptyset$ otherwise. Again, $Y \cap I \neq \emptyset$ if and only if $\exists x \in \min(Y) : x \in I$. Given such an x, then $\forall y \in I : \exists z \in I : z \geq x, y$ by directedness. Therefore, $I \subseteq \downarrow(I \cap \uparrow x) \subseteq \downarrow(I \cap Y)$. $\qquad\square$

6   Towards a richer theory of ideally effective WQOs

6.1   A minimal definition

As we mentioned in the remarks following Definition 3.1, our definition contains redundancies: some of the requirements are implied by the others. Here is the same definition in which we removed redundancies:

Definition 6.1 (Simply effective WQOs). *A WQO* $(X, \leq)$ *further equipped with data structures for X and $Idl(X)$ is* simply effective *if:*

(ID) *ideal inclusion $\subseteq$ is decidable on $Idl(X)$;*
(PI) *principal ideals are computable, that is, $x \mapsto \downarrow x$ is computable;*
(CF) *complementation of filters, denoted $\neg : Fil(X) \to Down(X)$, is computable;*
(II) *intersection of ideals, denoted $\cap : Idl(X) \times Idl(X) \to Down(X)$, is computable.*

A short presentation *of $(X, \leq)$ is a list of: data structures for X and $Idl(X)$, procedures for the above operations, the ideal decomposition of X.*

Note that a short presentation of $(X, \leq)$ is obtained from a presentation of $(X, \leq)$ by dropping procedures for $(\mathrm{OD}), (\mathrm{CI}), (\mathrm{IF})$ and by dropping (XF). Surprisingly, short presentations carry enough information:

Theorem 6.2. *There exists an algorithm that given a short presentation of $(X, \leq)$ outputs a presentation of $(X, \leq)$.*

Corollary 6.3. *A WQO $(X, \leq)$ (with data structures for X and $Idl(X)$) is ideally effective if and only if it is simply effective.*

Before we proceed to proving Theorem 6.2, why did we bother to display full presentations of WQOs in previous sections? Our proofs of ideal effectiveness would indeed have been shorter.

Our choice is motivated by practical reasons: the algorithms we have given until now are much more efficient than the ones deduced from Theorem 6.2, which is simply

impractical. (Indeed, most of these algorithms have been implemented, at the highest level of generality, by the second author.) Theorem 6.2 is more conceptual, and if one only needs computability results, then Theorem 6.2 provides a simpler path to this goal.

As practice goes, we will refine the notion of ideally effective WQOs to "efficient" ideally effective WQOs in Section 6.3. Most of the WQOs we have seen earlier are efficient in that sense. By contrast, the presentation of $(X, \leq)$ built from Section 6.3 is not *polynomial-time* (see Section 6.3 for a definition).

Proof (of Theorem 6.2). We explain how to obtain the missing procedures:

(OD): Given $x, y \in X$, $x \leq y \iff {\downarrow}x \subseteq {\downarrow}y$. The latter can be tested using (PI) and (ID).

(CI): We show a stronger statement, denoted (CD), that complementing an arbitrary downwards-closed set is computable. This strengthening is necessary for (IF).
Let D be an arbitrary downwards-closed set. We compute $\neg D$ as follows:

   1. Initialize $U := \emptyset$;
   2. While $\neg U \not\subseteq D$ do
      (a) pick some $x \in \neg U \cap \neg D$;
      (b) set $U := U \cup {\uparrow}x$

Every step of this high-level algorithm is effective. The complement $\neg U$ is computed using the description above: $\neg \bigcup_{i=1}^{n} {\uparrow}x_i = \bigcap_{i=1}^{n} \neg {\uparrow}x_i$ which is computed with (CF) and (II) (or with (XI) in case $n = 0$, i.e., for $U = \emptyset$). Then, inclusion $\neg U \subseteq D$ is tested with (ID). If this test fails, then we know $\neg U \cap \neg D$ is not empty, and thus we can enumerate elements $x \in X$ by brute force, and test membership in U and in D. Eventually, we will find some $x \in \neg U \cap \neg D$.
To prove partial correctness we use the following loop invariant: U is upwards-closed and $U \subseteq \neg D$. The invariant holds at initialization and is preserved by the loop's body since if ${\uparrow}x$ is upwards-closed and since $x \notin D$ and D downwards-closed imply ${\uparrow}X \subseteq \neg D$. Thus when/if the loop terminates, one has both $\neg U \subseteq D$ and the invariant $U \subseteq \neg D$, i.e., $U = \neg D$.
Finally, the algorithm terminates since it builds a strictly increasing sequence of upwards-closed sets, which must be finite by Lemma 2.3.

(IF): This follows from (CF) and (CD), by expressing intersection in terms of complement and union.

Lastly, we need to show that we can retrieve the filter decomposition of X. It suffices to use (CD) to compute $X = \neg\emptyset$.   □

The algorithm for (CD) computes an upwards-closed set U from an oracle answering queries of the form "Is $U \cap I$ empty?" for ideals I. It is an instance of the generalized Valk-Jantzen Lemma [25], an important tool for showing that some upwards-closed sets are computable. This algorithm was originally developed by Valk and Jantzen [58] in the specific case of $(\mathbb{N}^k, \leq_\times)$.

As seen in the above proof, the fact that (ID), (CF), (II) and (PI) entail (CI) is non-trivial. The existence of such a non-trivial redundancy in our definition raises the question of whether there are other hidden redundancies. The following theorem answers the question in the negative.

Theorem 6.4. *For each operation A among* (ID)*,* (CF)*,* (II) *and* (PI)*, there exists a WQO $(X_A, \leq_A)$ equipped with data structures for X and $Idl(X)$ for which operation A is not computable, while the other three are.*

This theorem means that short presentations are the shortest possible to capture the information we want. Technically, we should also argue that the ideal decomposition of X cannot be retrieved from procedures for operations (ID), (CF), (II), (PI).

The proofs of Theorem 6.4 and this last statement are rather technical and will not be included here: we refer to [31] for full proofs. Interestingly, all four WQOs $(X_A, \leq_A)$ presented in the proof of Theorem 6.4 are obtained as extensions of ideally effective WQOs. Notably, this proves that extensions of ideally effective WQOs are not always ideally effective, which justifies the two extra assumptions we used in Theorem 5.2. More precisely, it justifies that at least one of these assumptions is necessary, and indeed, we can always compute the closure function $\mathcal{C}l_\mathrm{F}$ from the closure function $\mathcal{C}l_\mathrm{I}$ (but not the converse!), and this in a uniform manner. The latter result relies on an algorithm that is very similar to the generalized Valk and Jantzen Lemma.

6.2 On alternative effectiveness assumptions

The set of effectiveness assumptions collected in Definition 3.1 or Definition 4.1 is motivated by the need to perform boolean operations on (downwards-, upwards-) closed subsets, as illustrated in our motivating examples from Section 2.1. Other choices are possible, and we illustrate a possible variant here.

6.2.1   A natural but not ideally effective constructor. Given two QOs $(X, \leq_X)$ and $(Y, \leq_Y)$, we can define the lexicographic quasi-ordering $\leq_\mathrm{lex}$ on $X \times Y$ by:

$$\langle x_1, y_1 \rangle \leq_\mathrm{lex} \langle x_2, y_2 \rangle \overset{\text{def}}{\Leftrightarrow} x_1 <_X x_2 \vee (x_1 \equiv_X x_2 \wedge y_1 \leq_Y y_2),$$

where classically, $\equiv_X$ denotes the equivalence relation $\leq_X \cap \geq_X$ and $<_X$ denotes the strict ordering associated to X, defined as $\leq_X \smallsetminus \equiv_X$.

Since $\leq_\mathrm{lex}$ is coarser than the product ordering $\leq_\times$ from Section 4.3, $(X \times Y, \leq_\mathrm{lex})$ is a WQO as soon as $\leq_X$ and $\leq_Y$ are. Besides, when $(X, \leq_X)$ and $(Y, \leq_Y)$ are ordinals, the lexicographic product corresponds to ordinal multiplication.

This WQO is simple and natural, but it is not always ideally effective in the sense of Definition 3.1 (at least for the natural representation of elements of $X \times Y$). The fact that our definition misses such a simple WQO is disturbing and will be discussed in the next subsection. For now, let us show why lexicographic product is not an ideally effective constructor.

Proposition 6.5. *There exists ideally effective WQOs $(X, \leq_X)$ and $(Y, \leq_Y)$ such that, representing elements of $X \times Y$ as pairs, $(X \times Y, \leq_\mathrm{lex})$ is not ideally effective. Notably, the lexicographic product constructor does not preserve ideal effectiveness. Note that this holds for any representation of ideals of $(X \times Y, \leq_\mathrm{lex})$.*

Proof. In line with the results mentioned in Section 6.1, we can show that there exists an ideally effective WQO $(X, \leq)$ such that we cannot decide whether $I \in Idl(X)$ is

principal (see [31] for a proof). Recall that an ideal is principal if $I = \downarrow x$ for some element $x \in X$. On the other side, define $Y = \{a, b\}$ ordered with equality. What matters here is that Y is not an ideal, and the two-symbol alphabet is the simplest WQO with this property. The WQO $(Y, \leq_Y)$ is obviously ideally effective.

Now we claim the following: given $I \in Idl(X)$, the downwards-closed set $I \times Y \subseteq X \times Y$ is an ideal of $(X \times Y, \leq_{\text{lex}})$ if and only if I is not a principal ideal of X.

Indeed, by contraposition, if $I = \downarrow x$ for some $x \in X$, then $\langle x, a \rangle$ and $\langle x, b \rangle$ do not have a common upper bound in $I \times Y$ with respect to $\leq_{\text{lex}}$, hence $I \times Y$ is not directed. Conversely, assume I is not principal and let $\langle x, c \rangle, \langle y, d \rangle \in I \times Y$. Since I is not principal, it is possible to find $z \in I$ such that $z > x$ and $z > y$. The element $\langle z, a \rangle$ is a suitable common upper bound, and $I \times Y$ is directed.

Now assume that $(X \times Y, \leq_{\text{lex}})$ is ideally effective. Given any $I \in Idl(X)$, we can compute $U = X \smallsetminus I$ since $(X, \leq_X)$ is ideally effective. The upwards-closed set U is represented by a filter decomposition $U = \bigcup_i \uparrow x_i$. It is thus possible to compute the upwards-closed set

$$V \stackrel{\text{def}}{=} U \times \{a\} \cup U \times \{b\} = \bigcup_i (\uparrow_{\text{lex}} \langle x_i, a \rangle \cup \uparrow_{\text{lex}} \langle x_i, b \rangle) \in Up(X \times Y, \leq_{\text{lex}})$$

Since $(X \times Y, \leq_{\text{lex}})$ is ideally effective, we can compute the complement of V (using axiom (CF)). But $(X \times Y) \smallsetminus V = I \times Y$, and we can compute a canonical decomposition of this downwards-closed set. The number of ideals of this canonical decomposition indicates whether $I \times Y$ is an ideal or not, and therefore whether the ideal I is principal or not.

In conclusion, assuming that $(X \times Y, \leq_{\text{lex}})$ is ideally effective, we can build an algorithm to decide whether an ideal of $(X, \leq_X)$ is principal: contradiction.   □

6.2.2   Deciding principality. In the previous subsection, we have shown that a very natural constructor, the lexicographic product, is not ideally effective. However, in practice $(X \times Y, \leq_{\text{lex}})$ is usually ideally effective, that is, the lexicographic product of two "actually used" WQOs $(X, \leq_X)$ and $(Y, \leq_Y)$ is ideally effective.

Thus, the problem seems to come from the fact that our definition allows "exotic" WQOs. Indeed, we can show (see [31]) that the lexicographic product of two ideal effective WQOs for which we can decide whether an ideal is principal, is ideally effective. All WQOs used in practice trivially meet this extra condition, to the point that we could argue that we should not accept as "effective" any WQO that would not meet this requirement.

Hence imagine we changed our definition of ideally effective WQOs to include the condition that principality of ideals is decidable. Lexicographic product is now an ideally effective constructor, and most of the constructors described in this chapter would remaing ideally effective, to the exception of extensions and quotients: Theorem 5.2 and Theorem 5.7 fail with this new definition (see again [31] for details).

6.2.3   Directions for future work. We would like to mention three directions in which our work can be extended.

The first one was carried out in [18], relying on the topological notion of notion called *Noetherian space* to generalize WQOs, in the following sense. Given a quasi-ordered set $(X, \leq)$, the Alexandroff topology has as open sets exactly the upwards-closed sets for the quasi-ordering $\leq$. It turns out that the Alexandroff topology associated to $\leq$ is Noetherian if and only if $\leq$ is a WQO on X. There are also Noetherian topologies that do not arise as Alexandroff topologies, for example the cofinite topology on an infinite set, or the Zariski topology on a Noetherian ring. One advantage of Noetherian spaces is that they are preserved under more constructors than WQOs, e.g., the full powerset of a Noetherian space (with the so-called lower Vietoris topology) is again Noetherian. In [18], the authors define a notion of effectiveness very similar to ours for Noetherian spaces, which however excludes complements and filters, which do not make sense there. Similarly, this notion of effectiveness is preserved under many constructors.

A second extension of this work was carried out in [31]. The motivation is close to the one above: handling more constructors. As mentioned in Section 4.5, the infinite powerset $\mathcal{P}(X)$ of a WQO, ordered with the Hoare ordering is not a WQO in general. However, the class of WQOs for which $(\mathcal{P}(X), \sqsubseteq_H)$ is a WQO is well-known: these WQOs are called ω^2-WQO (*e.g.* see [47,33]). The second author [31] proposes a generalization of our notion of ideal effective WQOs which he calls ideal effective ω^2-WQOs (also Idl^2-effective WQOs). He then shows that the constructors presented in this chapter also preserve this stronger notion of Idl^2-effectiveness, and also prove that *e.g.* the powerset of an Idl^2-effective WQO, ordered with the Hoare quasi-ordering, is an ideally effective WQO. The notion of ω^2-WQO can be generalized to the notion of α-WQO for any indecomposable ordinal α, eventually leading to the notion of *better quasi-ordering* (α-WQO for every countable α). In [31], the author raises the question on how to generalize ideal effectiveness to these classes of quasi-orderings.

Finally, one might challenge our own decision of representing upwards- and downwards-closed sets as their filter/ideal decompositions. Its main advantage is genericity: as proved in Section 2, this decomposition is possible in any WQO. It is also very convenient. In the simple cases of $(\mathbb{N}^k, \leq_\times)$ and $(A^*, \leq_*)$, the representations and algorithms we illustrated in Section 2.1 have been used for years by researchers who were not aware that they were manipulating ideals. This suggests that the idea is somehow natural.

This does not rule our the existence of better ad-hoc solutions when considering a specific WQO, notably in terms of efficiency. As will be seen in Section 6.3, the procedures we have presented in Section 4.4 have an exponential-time worst-case complexity. This exponential blow-up essentially occurs when one has to distribute the unions over the products in order to retrieve an actual filter/ideal decomposition. We are not sure this can be averted, but when one only needs to represent certain particular closed subsets of $(X^*, \leq_*)$, better representations do exist: see for instance [22].

6.3   On computational complexity

In [31], the second author provides a complexity analysis of the algorithms we have described in this chapter. Let us briefly summarize the complexity of the WQO constructors we have considered.

Formally, let us define a *polynomial-time* ideally effective WQO to be an ideally effective WQO for which there exist *polynomial-time* procedures for (OD), (ID), (CF), (IF), (CI), (II), (PI). A presentation of an ideally effective WQO is said to be *polynomial-time* if all the procedures it is composed of run in polynomial time. For instance, $\mathbb{N}$ is a polynomial-time ideally effective WQO, and the presentation we gave for it is polynomial-time. However, a WQO as simple as $(A^*, \leq_*)$, where $A = \{a, b\}$, is not polynomial-time, at least for our choice of data structure for A^* and $Idl(A^*)$. Indeed, observe that the upwards-closed set $U_n = \uparrow a^n \cap \uparrow b^n$ has at least exponentially many (in n) minimal elements: any word with n a's and n b's is a minimal element of U_n. Therefore, the filter decomposition of U_n is of exponential size in n, and thus requires exponential-time to compute.

However, for instance, the Cartesian product $(X \times Y, \leq_\times)$ of polynomial-time ideally effective WQOs is polynomial-time. (That would fail if X or Y were not polynomial-time: for instance, if $(X, \leq_X) = (A^*, \leq_*)$, then the upwards-closed set $\uparrow(a^n, y_1) \cap \uparrow(b^n, y_2)$ has at least exponentially minimal elements, independently of the filter decomposition of $\uparrow y_1 \cap \uparrow y_2$.) Furthermore, from polynomial-time presentations of $(X, \leq_X)$ and $(Y, \leq_Y)$, the presentation of $(X \times Y, \leq_\times)$ we compute in Section 4.3 is polynomial-time as well. This motivates the following definition: an ideally effective constructor C is *polynomial-time* if it is possible to compute a polynomial-time presentation for $C[(X_1, \leq_1), \ldots, (X_n, \leq_n)]$ given polynomial-time presentations of $(X_1, \leq_1), \ldots, (X_n, \leq_n)$. Note that we require that the procedures of the presentation for $C[(X_1, \leq_1), \ldots, (X_n, \leq_n)]$ are polynomial-time, but we do not make any assumption on the complexity of the procedure that builds the new presentation from presentations for each $(X_i, \leq_i)$.

With this definition in mind, here is a summary of the complexity results from [31]:

- Both disjoint sum and lexicographic sum are polynomial-time ideally effective constructors—this is a trivial analysis of the presentation of Section 4.2.
- Cartesian product is a polynomial-time ideally effective constructor; that again follows easily from an analysis of Section 4.3.
- Higman's sequence extension QO is *not* a polynomial-time ideally effective constructor. As we have seen above, already in the simple case of finite sequences over a finite alphabet, some operations require exponential time. It is not difficult to see that the presentation we gave in Section 4.4 consists of exponential time procedures.
- The finite powerset constructor (under the Hoare quasi-ordering) is a polynomial-time ideally effective constructor. This again follows from an easy analysis of Section 4.5. This justifies implementing $\mathcal{P}_f(X)$ directly, and not as a quotient of X^*.
- The finite multiset constructor, under multiset embedding, is an exponential-time ideally effective constructor, and already $(\mathbb{N}^{2^\circledast}, \leq_{\text{emb}})$ is not a polynomial-time ideally effective WQO. However, $(A^\circledast, \leq_{\text{emb}})$ and $(\mathbb{N}^\circledast, \leq_{\text{emb}})$ are polynomial-time effective WQOs when A is a finite alphabet under equality.

7   Concluding remarks

We have proposed a set of effectiveness assumptions that allow one to compute with upwards-closed and downwards-closed subsets of WQOs, represented as their canonical filter and ideal decompositions respectively. These effectiveness assumptions are

fulfilled in the main WQOs that appear in practical computer applications, which are built using constructors that we have shown to be ideally effective. Our algorithms unify and generalize some algorithms that have been used for many years in simple settings, such as $\mathbb{N}^k$ or the set of finite words ordered by embedding.

We have not considered any WQO constructor more complex than sequence extension, and this is an obvious direction for extending this work. How does one compute with closed subsets of finite labeled trees ordered by Kruskal's homeomorphic embedding? Or of some class of finite graphs well-quasi-ordered by some notion of embedding? The case of finite trees has already been tackled by the first author, see [18,26]. The technicalities are daunting, well beyond the ambitions of this chapter, however.

Acknowledgments

References

1. P. A. Abdulla, M. F. Atig, and J. Cederberg. Timed lossy channel systems. In *FST&TCS 2012, LIPIcs 18*, pages 374–386. Leibniz-Zentrum für Informatik, 2012.
2. P. A. Abdulla, K. Čerāns, B. Jonsson, and Y.-K. Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160(1/2):109–127, 2000.
3. P. A. Abdulla, A. Collomb-Annichini, A. Bouajjani, and B. Jonsson. Using forward reachability analysis for verification of lossy channel systems. *Formal Methods in System Design*, 25(1):39–65, 2004.
4. Y. Akama. Set systems: Order types, continuous nondeterministic deformations, and quasi-orders. *Theoretical Comput. Sci.*, 412(45):6235–6251, 2011.
5. G. Bachmeier, M. Luttenberger, and M. Schlund. Finite automata for the sub- and superword closure of CFLs: Descriptional and computational complexity. In *LATA 2015, LNCS 8977*, pages 473–485. Springer, 2015.
6. N. Bertrand and Ph. Schnoebelen. Computable fixpoints in well-structured symbolic model checking. *Formal Methods in System Design*, 43(2):233–267, 2013.
7. G. Birkhoff. *Lattice Theory*, volume XXV of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, R.I., corrected reprint of the third edition (1967) edition, 1979.
8. M. Blondin, A. Finkel, and P. McKenzie. Well behaved transition systems. *Logical Methods in Comp. Science*, 13(3):1–19, 2017.
9. M. Blondin, A. Finkel, and P. McKenzie. Handling infinitely branching well-structured transition systems. *Information and Computation*, 258:28–49, 2018.
10. K. T. Chen, R. H. Fox, and R. C. Lyndon. Free differential calculus, IV. the quotient groups of the lower central series. *Annals of Mathematics*, 68(1):81–95, 1958.
11. F. D'Alessandro and S. Varricchio. Well quasi-orders in formal language theory. In *DLT 2008, LNCS 5257*, pages 84–95. Springer, 2008.
12. N. Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3(1–2):69–115, 1987.
13. N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Commun. ACM*, 22(8):465–476, 1979.
14. L. P. Deutsch. An interactive program verifier. Technical Report CSL-73-1, Xerox Palo Alto Research Center, 1973.
15. A. Finkel. The ideal theory for WSTS. In *RP 2016, LNCS 9899*, pages 1–22. Springer, 2016.

16. A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part I: Completions. In *STACS 2009, LIPIcs 3*, pages 433–444. Leibniz-Zentrum für Informatik, 2009.

17. A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part II: Complete WSTS. In *ICALP 2009, LNCS 5556*, pages 188–199. Springer, 2009.

18. A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, Part I: Completions. *Mathematical Structures in Computer Science*, 2019. Submitted. Journal version of [16].

19. A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Comput. Sci.*, 256(1–2):63–92, 2001.

20. J. H. Gallier. What's so special about Kruskal's theorem and the ordinal γ_0? a survey of some results in proof theory. *Annals of Pure and Applied Logic*, 53(3):199–260, 1991.

21. G. Geeraerts, J.-F. Raskin, and L. Van Begin. Expand, enlarge and check: New algorithms for the coverability problem of WSTS. *J. Comput. Syst. Sci.*, 72(1):180–203, 2006.

22. T. Geffroy, J. Leroux, and G. Sutre. Backward coverability with pruning for lossy channel systems. In *SPIN 2017*, pages 132–141. ACM Press, 2017.

23. E. Goto. Monocopy and associative algorithms in an extended Lisp. Technical report, University of Tokyo, Japan, May 1974.

24. J. Goubault. HimML: Standard ML with fast sets and maps. In *5th ACM SIGPLAN Workshop on ML and its Applications*, 1994. doi:10.1.1.40.4967.

25. J. Goubault-Larrecq. On a generalization of a result by Valk and Jantzen. Research Report LSV-09-09, Laboratoire Spécification et Vérification, ENS Cachan, France, May 2009.

26. J. Goubault-Larrecq and S. Schmitz. Deciding piecewise testable separability for regular tree languages. In *ICALP 2016, LIPIcs 55*, pages 97:1–97:15. Leibniz-Zentrum für Informatik, 2016.

27. Ch. Haase, S. Schmitz, and Ph. Schnoebelen. The power of priority channel systems. *Logical Methods in Comp. Science*, 10(4:4), 2014.

28. P. Habermehl, R. Meyer, and H. Wimmel. The downward-closure of Petri net languages. In *ICALP 2010, LNCS 6199*, pages 466–477. Springer, 2010.

29. S. Haddad, S. Schmitz, and Ph. Schnoebelen. The ordinal-recursive complexity of timed-arc Petri nets, data nets, and other enriched nets. In *LICS 2012*, pages 355–364. IEEE Comp. Soc. Press, 2012.

30. L. H. Haines. On free monoids partially ordered by embedding. *Journal of Combinatorial Theory*, 6(1):94–98, 1969.

31. S. Halfon. *On Effective Representations of Well Quasi-Orderings*. Thèse de doctorat, École Normale Supérieure Paris-Saclay, France, June 2018.

32. G. Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, 2(7):326–336, 1952.

33. P. Jančar. A note on well quasi-orderings for powersets. *Information Processing Letters*, 72(5–6):155–161, 1999.

34. M. Kabil and M. Pouzet. Une extension d'un théorème de P. Jullien sur les âges de mots. *RAIRO Theoretical Informatics and Applications*, 26(5):449–482, 1992.

35. R. M. Karp and R. E. Miller. Parallel program schemata. *J. Comput. Syst. Sci.*, 3(2):147–195, 1969.

36. I. Kříž and R. Thomas. On well-quasi-ordering finite structures with labels. *Graphs and Combinatorics*, 6(1):41–49, 1990.

37. J. B. Kruskal. Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture. *Transactions of the American Mathematical Society*, 95(2):210–225, 1960.

38. J. B. Kruskal. The theory of well-quasi-ordering: A frequently discovered concept. *Journal of Combinatorial Theory, Series A*, 13(3):297–305, 1972.

39. R. Lazić, T. Newcomb, J. Ouaknine, A. W. Roscoe, and J. Worrell. Nets with tokens which carry data. *Fundamenta Informaticae*, 88(3):251–274, 2008.

40. R. Lazić and S. Schmitz. The ideal view on Rackoff's coverability technique. In *RP 2015, LNCS 9328*, pages 76–88. Springer, 2015.
41. R. Lazić and S. Schmitz. The complexity of coverability in ν-Petri nets. In *LICS 2016*, pages 467–476. ACM Press, 2016.
42. J. Leroux and S. Schmitz. Demystifying reachability in vector addition systems. In *LICS 2015*, pages 56–67. IEEE Comp. Soc. Press, 2015.
43. J. Leroux and S. Schmitz. Ideal decompositions for vector addition systems. In *STACS 2016, LIPIcs 47*, pages 1:1–1:13. Leibniz-Zentrum für Informatik, 2016.
44. M. Leuschel. Homeomorphic embedding for online termination of symbolic methods. In *The Essence of Computation, Complexity, Analysis, Transformation, LNCS 2566*, pages 379–403. Springer, 2002.
45. L. Lovász. Graph minor theory. *Bull. Amer. Math. Soc.*, 43(1):75–86, 2006.
46. P. Manolios and D. Vroon. Ordinal arithmetic: Algorithms and mechanization. *Journal of Automated Reasoning*, 34(4):387–423, 2005.
47. A. Marcone. Fine analysis of the quasi-orderings on the power set. *Order*, 18(4):339–347, 2001.
48. J.-É. Pin. Logic, semigroups and automata on words. *Annals of Mathematics and Artificial Intelligence*, 16(1):343–384, 1996.
49. R. Rado. Partial well-ordering of sets of vectors. *Mathematika*, 1(2):89–95, 1954.
50. N. Robertson and P. D. Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.
51. G. E. Sacks. Constructive ordinals and Π_1^1 sets. In *Higher Recursion Theory*, volume 2 of *Perspectives in Mathematical Logic*, chapter 1, pages 3–21. Springer, 1990.
52. J. Sakarovitch and I. Simon. Subwords. In M. Lothaire, editor, *Combinatorics on Words*, volume 17 of *Encyclopedia of Mathematics and Its Applications*, chapter 6, pages 105–142. Cambridge Univ. Press, 1983.
53. S. Schmitz. Complexity bounds for ordinal-based termination. In *RP 2014, LNCS 8762*, pages 1–19. Springer, 2014.
54. S. Schmitz. Complexity hierarchies beyond Elementary. *ACM Trans. Computation Theory*, 8(1), 2016.
55. S. Schmitz and Ph. Schnoebelen. Multiply-recursive upper bounds with Higman's lemma. In *ICALP 2011, LNCS 6756*, pages 441–452. Springer, 2011.
56. S. Schmitz and Ph. Schnoebelen. Algorithmic aspects of WQO theory. Lecture notes, 2012.
57. S. Schmitz and Ph. Schnoebelen. The power of well-structured systems. In *CONCUR 2013, LNCS 8052*, pages 5–24. Springer, 2013.
58. R. Valk and M. Jantzen. The residue of vector sets with applications to decidability problems in Petri nets. *Acta Informatica*, 21(6):643–674, 1985.
59. G. Zetzsche. Computing downward closures for stacked counter automata. In *STACS 2015, LIPIcs 30*, pages 743–756. Leibniz-Zentrum für Informatik, 2015.
60. G. Zetzsche. Separability by piecewise testable languages and downward closures beyond subwords. In *LICS 2018*, pages 929–938. ACM Press, 2018.