# Algorithmic Aspects of WQO (Well-Quasi-Ordering) Theory

## Part IV: Ideals of WQOs and Their Algorithms

Sylvain Schmitz & Philippe Schnoebelen

LSV, CNRS & ENS Cachan

ESSLLI 2016, Bozen/Bolzano, Aug 22-26, 2016

Lecture notes & exercises available at

`http://www.lsv.fr/~schmitz/teach/2016_esslli`

# If you missed Part I

$(X, \leqslant)$ is a well-quasi-ordering (a wqo) if any infinite sequence $x_0, x_1, x_2 \ldots$ over $X$ contains an increasing pair $x_i \leqslant x_j$ (for some $i < j$)

**Examples.**

1. $(\mathbb{N}^k, \leqslant_\times)$ is a wqo (Dickson's Lemma)
   where, e.g., $(3,2,1) \leqslant_\times (5,2,2)$ but $(1,2,3) \not\leqslant_\times (5,2,2)$
2. $(\Sigma^*, \leqslant_*)$ is a wqo (Higman's Lemma)
   where, e.g., $abc \leqslant_* bacbc$ but $cba \not\leqslant_* bacbc$

Motivation for remaining two lectures:

- *WQO-based algorithms often have to handle/reason about infinite upward- or downward-closed sets*
- This is a non-trivial subtask
- But there exists a powerful & generic approach via ideals

# IF YOU MISSED PART I

$(X, \leqslant)$ is a well-quasi-ordering (a wqo) if any infinite sequence $x_0, x_1, x_2 \ldots$ over $X$ contains an increasing pair $x_i \leqslant x_j$ (for some $i < j$)

**Examples.**

1. $(\mathbb{N}^k, \leqslant_\times)$ is a wqo (Dickson's Lemma)
   where, e.g., $(3,2,1) \leqslant_\times (5,2,2)$ but $(1,2,3) \not\leqslant_\times (5,2,2)$
2. $(\Sigma^*, \leqslant_*)$ is a wqo (Higman's Lemma)
   where, e.g., $abc \leqslant_* bacbc$ but $cba \not\leqslant_* bacbc$

Motivation for remaining two lectures:

- *WQO-based algorithms often have to handle/reason about infinite upward- or downward-closed sets*
- This is a non-trivial subtask
- But there exists a powerful & generic approach via ideals

# Outline for Part IV

- The need for data structure and algorithms for closed subsets
- Ideals and filters : basics
- Effective ideals and filters
- The Valk-Jantzen-Goubault-Larrecq algorithm
- Building complex effective wqos from simpler ones : tuples, sequences, powersets, substructures, weakening, etc.

# Handling upward-closed subsets

When verifying safety for a WSTS we computed upward-closed subsets :

$$B \subseteq Pre^{\leqslant 1}(B) \subseteq Pre^{\leqslant 2}(B) \subseteq \cdots \subseteq \bigcup_m Pre^{\leqslant m}(B) = Pre^*(B)$$

How does one do this? Let's assume we are in $(\mathbb{N}^2, \leqslant_\times)$ and consider upward-closed subsets $U, U', V, ..$

There is the finite basis presentation:

$$U = \uparrow(a_1, b_1) \cup \cdots \cup \uparrow(a_\ell, b_\ell) \quad V = \uparrow(c_1, d_1) \cup \cdots \cup \uparrow(c_m, d_m)$$

How does one compute with this representation:

- How do we compare $U$ ? E.g. test whether $U \subseteq V$ ?
- How do we add to $U$ ? E.g. perform $U \leftarrow U \cup V$
- How do we remove from $U$ ? E.g. perform $U \leftarrow U \cap V$

# HANDLING UPWARD-CLOSED SUBSETS

When verifying safety for a WSTS we computed upward-closed subsets :

$$B \subseteq Pre^{\leqslant 1}(B) \subseteq Pre^{\leqslant 2}(B) \subseteq \cdots \subseteq \bigcup_m Pre^{\leqslant m}(B) = Pre^*(B)$$

How does one do this? Let's assume we are in $(\mathbb{N}^2, \leqslant_\times)$ and consider upward-closed subsets $U, U', V, ..$

There is the finite basis presentation:

$$U = \uparrow(a_1, b_1) \cup \cdots \cup \uparrow(a_\ell, b_\ell) \quad V = \uparrow(c_1, d_1) \cup \cdots \cup \uparrow(c_m, d_m)$$

How does one compute with this representation:

- How do we compare $U$ ? E.g. test whether $U \subseteq V$ ?
- How do we add to $U$ ? E.g. perform $U \leftarrow U \cup V$
- How do we remove from $U$ ? E.g. perform $U \leftarrow U \cap V$

# Handling upward-closed subsets

When verifying safety for a WSTS we computed upward-closed subsets :

$$B \subseteq Pre^{\leqslant 1}(B) \subseteq Pre^{\leqslant 2}(B) \subseteq \cdots \subseteq \bigcup_m Pre^{\leqslant m}(B) = Pre^*(B)$$

How does one do this? Let's assume we are in $(\mathbb{N}^2, \leqslant_\times)$ and consider upward-closed subsets $U, U', V, ..$

There is the finite basis presentation:

$$U = \uparrow(a_1, b_1) \cup \cdots \cup \uparrow(a_\ell, b_\ell) \quad V = \uparrow(c_1, d_1) \cup \cdots \cup \uparrow(c_m, d_m)$$

How does one compute with this representation:

- How do we compare $U$ ? E.g. test whether $U \subseteq V$ ?
- How do we add to $U$ ? E.g. perform $U \leftarrow U \cup V$
- How do we remove from $U$ ? E.g. perform $U \leftarrow U \cap V$

# HANDLING UPWARD-CLOSED SUBSETS

When verifying safety for a WSTS we computed upward-closed subsets :

$$B \subseteq Pre^{\leqslant 1}(B) \subseteq Pre^{\leqslant 2}(B) \subseteq \cdots \subseteq \bigcup_m Pre^{\leqslant m}(B) = Pre^*(B)$$

How does one do this? Let's assume we are in $(\mathbb{N}^2, \leqslant_\times)$ and consider upward-closed subsets $U, U', V, ..$

There is the finite basis presentation:

$$U = \uparrow(a_1, b_1) \cup \cdots \cup \uparrow(a_\ell, b_\ell) \quad V = \uparrow(c_1, d_1) \cup \cdots \cup \uparrow(c_m, d_m)$$

How does one compute with this representation:

- How do we compare $U$ ? E.g. test whether $U \subseteq V$ ?
- How do we add to $U$ ? E.g. perform $U \leftarrow U \cup V$
- How do we remove from $U$ ? E.g. perform $U \leftarrow U \cap V$

# HANDLING UPWARD-CLOSED SUBSETS

When verifying safety for a WSTS we computed upward-closed subsets :

$$B \subseteq Pre^{\leqslant 1}(B) \subseteq Pre^{\leqslant 2}(B) \subseteq \cdots \subseteq \bigcup_m Pre^{\leqslant m}(B) = Pre^*(B)$$

How does one do this? Let's assume we are in $(\mathbb{N}^2, \leqslant_\times)$ and consider upward-closed subsets $U, U', V, ..$

There is the finite basis presentation:

$$U = \uparrow(a_1, b_1) \cup \cdots \cup \uparrow(a_\ell, b_\ell) \quad V = \uparrow(c_1, d_1) \cup \cdots \cup \uparrow(c_m, d_m)$$

How does one compute with this representation:

- How do we compare $U$ ? E.g. test whether $U \subseteq V$ ?
- How do we add to $U$ ? E.g. perform $U \leftarrow U \cup V$
- How do we remove from $U$ ? E.g. perform $U \leftarrow U \cap V$

# Upward-Closed Subsets of $(\Sigma^*, \leqslant_*)$

Let us consider words with subword ordering, e.g., for lossy channel systems:

$$U = \uparrow abc \cup \cdots \cup \uparrow ddca \quad V = \uparrow bb \cup \cdots$$

How do we compare such sets?

How do we add to them ?

How do we remove from them ? E.g., how do we perform $U \leftarrow U \cap V$?

And how do we do $U \leftarrow U \smallsetminus \downarrow ba$ ?

**Bottom line:** These are feasible but not trivial !

Question 1: Can we handle $\mathbb{N}^k$ and $\Sigma^*$ efficiently ?

Question 2: And what about other WQOs? Recall the example with $(\mathbb{N}^2)^*$

# UPWARD-CLOSED SUBSETS OF $(\Sigma^*, \leqslant_*)$

Let us consider words with subword ordering, e.g., for lossy channel systems:

$$U = \uparrow abc \cup \cdots \cup \uparrow ddca \quad V = \uparrow bb \cup \cdots$$

How do we compare such sets?

How do we add to them ?

How do we remove from them ? E.g., how do we perform $U \leftarrow U \cap V$?

And how do we do $U \leftarrow U \setminus \downarrow ba$ ?

**Bottom line:** These are feasible but not trivial !

Question 1: Can we handle $\mathbb{N}^k$ and $\Sigma^*$ efficiently ?

Question 2: And what about other WQOs? Recall the example with $(\mathbb{N}^2)^*$

# UPWARD-CLOSED SUBSETS OF $(\Sigma^*, \leqslant_*)$

Let us consider words with subword ordering, e.g., for lossy channel systems:

$$U = \uparrow abc \cup \cdots \cup \uparrow ddca \quad V = \uparrow bb \cup \cdots$$

How do we compare such sets?

How do we add to them ?

How do we remove from them ? E.g., how do we perform $U \leftarrow U \cap V$?

And how do we do $U \leftarrow U \setminus \downarrow ba$ ?

**Bottom line:** These are feasible but not trivial !

Question 1: Can we handle $\mathbb{N}^k$ and $\Sigma^*$ efficiently ?

Question 2: And what about other WQOs? Recall the example with $(\mathbb{N}^2)^*$

# UPWARD-CLOSED SUBSETS OF $(\Sigma^*, \leqslant_*)$

Let us consider words with subword ordering, e.g., for lossy channel systems:

$$U = \uparrow abc \cup \cdots \cup \uparrow ddca \quad V = \uparrow bb \cup \cdots$$

How do we compare such sets?

How do we add to them ?

How do we remove from them ? E.g., how do we perform $U \leftarrow U \cap V$?

And how do we do $U \leftarrow U \setminus \downarrow ba$ ?

**Bottom line:** These are feasible but not trivial !

Question 1: Can we handle $\mathbb{N}^k$ and $\Sigma^*$ efficiently ?

Question 2: And what about other WQOs? Recall the example with $(\mathbb{N}^2)^*$

# UPWARD-CLOSED SUBSETS OF $(\Sigma^*, \leqslant_*)$

Let us consider words with subword ordering, e.g., for lossy channel systems:

$$U = \uparrow abc \cup \cdots \cup \uparrow ddca \quad V = \uparrow bb \cup \cdots$$

How do we compare such sets?

How do we add to them ?

How do we remove from them ? E.g., how do we perform $U \leftarrow U \cap V$?

And how do we do $U \leftarrow U \setminus \downarrow ba$ ?

**Bottom line:** These are feasible but not trivial !

Question 1: Can we handle $\mathbb{N}^k$ and $\Sigma^*$ efficiently ?

Question 2: And what about other WQOs? Recall the example with $(\mathbb{N}^2)^*$

# What about downward-closed subsets?

**Problem:** can't always be represented under the form
$D = \downarrow x_1 \cup \cdots \cup \downarrow x_\ell$

Recall: $D$ can always be represented by excluded minors:

$$D = X \smallsetminus \uparrow m_1 \smallsetminus \uparrow m_2 \cdots \smallsetminus \uparrow m_\ell$$

This amounts to $D = \neg U$ with $U = \uparrow m_1 \cup \cdots \cup \uparrow m_\ell$.

**Problem:** Not very convenient for simple sets:
— How do you represent $\downarrow(2,2)$ ? $\downarrow ab$ ?
— How do you compute $D \cup D'$ ?

There is a better solution: decompose into primes!

# WHAT ABOUT DOWNWARD-CLOSED SUBSETS?

**Problem:** can't always be represented under the form $D = \downarrow x_1 \cup \cdots \cup \downarrow x_\ell$

Recall: $D$ can always be represented by excluded minors:

$$D = X \smallsetminus \uparrow m_1 \smallsetminus \uparrow m_2 \cdots \smallsetminus \uparrow m_\ell$$

This amounts to $D = \neg U$ with $U = \uparrow m_1 \cup \cdots \cup \uparrow m_\ell$.

**Problem:** Not very convenient for simple sets:
— How do you represent $\downarrow(2,2)$ ? $\downarrow ab$ ?
— How do you compute $D \cup D'$ ?

There is a better solution: decompose into primes!

# What about downward-closed subsets?

**Problem:** can't always be represented under the form
$D = {\downarrow}x_1 \cup \cdots \cup {\downarrow}x_\ell$

Recall: $D$ can always be represented by excluded minors:

$$D = X \smallsetminus {\uparrow}m_1 \smallsetminus {\uparrow}m_2 \cdots \smallsetminus {\uparrow}m_\ell$$

This amounts to $D = \neg U$ with $U = {\uparrow}m_1 \cup \cdots \cup {\uparrow}m_\ell$.

**Problem:** Not very convenient for simple sets:
— How do you represent ${\downarrow}(2,2)$ ? ${\downarrow}ab$ ?
— How do you compute $D \cup D'$ ?

There is a better solution: decompose into primes!

# WHAT ABOUT DOWNWARD-CLOSED SUBSETS?

**Problem:** can't always be represented under the form
$D = \downarrow x_1 \cup \cdots \cup \downarrow x_\ell$

Recall: $D$ can always be represented by excluded minors:

$$D = X \smallsetminus \uparrow m_1 \smallsetminus \uparrow m_2 \cdots \smallsetminus \uparrow m_\ell$$

This amounts to $D = \neg U$ with $U = \uparrow m_1 \cup \cdots \cup \uparrow m_\ell$.

**Problem:** Not very convenient for simple sets:
— How do you represent $\downarrow(2,2)$ ? $\downarrow ab$ ?
— How do you compute $D \cup D'$ ?

There is a better solution: decompose into primes!

# Primes, Up and Down

Fix $(X,\leqslant)$ WQO and consider $Up(X) = \{U, U', \ldots\}$ and $Down(X) = \{D, D', \ldots\}$

**Def.** 1. $U$ ($\neq \emptyset$) is (up-) prime $\overset{\text{def}}{\Leftrightarrow}$ $U \subseteq (U_1 \cup U_2)$ implies $U \subseteq U_1$ or $U \subseteq U_2$.

2. $D$ ($\neq \emptyset$) is (down-) prime $\overset{\text{def}}{\Leftrightarrow}$ $D \subseteq (D_1 \cup D_2)$ implies $D \subseteq D_1$ or $D \subseteq D_2$.

**Examples:** for any $x \in X$, $\uparrow x$ is up-prime and $\downarrow x$ is down-prime

**Lem. (Irreducibility)**
1. $U$ is prime iff $U = U_1 \cup \cdots \cup U_n$ implies $U = U_i$ for some $i$
2. $D$ is prime iff $D = D_1 \cup \cdots \cup D_n$ implies $D = D_i$ for some $i$

**Lem. (Existence of Prime Decompositions, aka Completeness)**
1. Every $U \in Up$ is a finite union of up-primes
2. Every $D \in Down$ is a finite union of down-primes

# MINIMAL PRIME DECOMPOSITIONS

**Def.** A prime decomposition $U$ (or $D$) $= P_1 \cup \cdots \cup P_n$ is minimal $\overset{\text{def}}{\Leftrightarrow}$ $\forall i,j : P_i \subseteq P_j$ implies $i = j$.

**Thm.** Every $U$ (or $D$) has a unique minimal prime decomposition. It is called its canonical decomposition

**Thm. (Primes are Filters/Ideals)**
1. The up-primes of $X$ are exactly the $\uparrow x$ for $x \in X$ (the principal filters)
2. The down-primes of $X$ are exactly the ideals of $X$ (see below)

**Def.** An ideal $I$ of $X$ is a non-empty directed downward-closed subset

Recall: $I$ directed $\overset{\text{def}}{\Leftrightarrow}$ $x, y \in I \implies \exists z \in I : x \leqslant z \geqslant y$

Example: any $\downarrow x$ is an ideal (called a principal ideal)

Example: If $x_1 < x_2 < x_3 \ldots$ is an increasing sequence then $\bigcup_i \downarrow x_i$ is an ideal

Exercise: Take $D = \{(a,b) \mid \min(a,b) < 3 \vee \max(a,b) < 7\}$ in $\mathbb{N}^2$

# Minimal Prime Decompositions

**Def.** A prime decomposition $U$ (or $D$) $= P_1 \cup \cdots \cup P_n$ is minimal
$\overset{\text{def}}{\Leftrightarrow} \forall i,j : P_i \subseteq P_j$ implies $i = j$.

**Thm.** Every $U$ (or $D$) has a unique minimal prime decomposition. It is called its canonical decomposition

**Thm. (Primes are Filters/Ideals)**
1. The up-primes of $X$ are exactly the $\uparrow x$ for $x \in X$ (the principal filters)
2. The down-primes of $X$ are exactly the ideals of $X$ (see below)

**Def.** An ideal $I$ of $X$ is a non-empty directed downward-closed subset

Recall: $I$ directed $\overset{\text{def}}{\Leftrightarrow} x, y \in I \implies \exists z \in I : x \leqslant z \geqslant y$

Example: any $\downarrow x$ is an ideal (called a principal ideal)

Example: If $x_1 < x_2 < x_3 \ldots$ is an increasing sequence then $\bigcup_i \downarrow x_i$ is an ideal

Exercise: Take $D = \{(a,b) \mid \min(a,b) < 3 \lor \max(a,b) < 7\}$ in $\mathbb{N}^2$

# NAILING DOWN THE IDEALS

The ideals of $(\mathbb{N}, \leqslant)$ are exactly all $\downarrow n$ together with $\mathbb{N}$ itself

Hence $(Idl(\mathbb{N}), \subseteq) \equiv (\mathbb{N} \cup \{\omega\}, \leqslant)$, denoted $\mathbb{N}_\omega$ ($\equiv \omega + 1$)

**Thm.** The ideals of $(X_1 \times X_2, \leqslant_\times)$ are exactly the $J_1 \times J_2$ for $J_i$ an ideal of $X_i$ ($i = 1, 2$)

Hence $(Idl(X_1 \times X_2), \subseteq) \equiv Idl(X_1, \subseteq) \times Idl(X_2, \subseteq)$ Very nice !!!!

**Coro.** The ideals of $(\mathbb{N}^k, \leqslant_\times)$ are handled like $\mathbb{N}^k_\omega$

**Example:** Assume $U = \uparrow(2,2)$ and $D = \downarrow(4, \omega) \cup \downarrow(6, 3)$.
What is $U \smallsetminus D$ and $D \smallsetminus U$?

# Nailing down the ideals

The ideals of $(\mathbb{N}, \leqslant)$ are exactly all $\downarrow n$ together with $\mathbb{N}$ itself

Hence $(Idl(\mathbb{N}), \subseteq) \equiv (\mathbb{N} \cup \{\omega\}, \leqslant)$, denoted $\mathbb{N}_\omega$ $(\equiv \omega + 1)$

**Thm.** The ideals of $(X_1 \times X_2, \leqslant_\times)$ are exactly the $J_1 \times J_2$ for $J_i$ an ideal of $X_i$ $(i = 1, 2)$

Hence $(Idl(X_1 \times X_2), \subseteq) \equiv Idl(X_1, \subseteq) \times Idl(X_2, \subseteq)$ Very nice !!!!

**Coro.** The ideals of $(\mathbb{N}^k, \leqslant_\times)$ are handled like $\mathbb{N}_\omega^k$

**Example:** Assume $U = \uparrow(2,2)$ and $D = \downarrow(4,\omega) \cup \downarrow(6,3)$.
What is $U \setminus D$ and $D \setminus U$?

# Nailing down the ideals

The ideals of $(\mathbb{N},\leqslant)$ are exactly all $\downarrow n$ together with $\mathbb{N}$ itself

Hence $(\mathit{Idl}(\mathbb{N}),\subseteq) \equiv (\mathbb{N}\cup\{\omega\},\leqslant)$, denoted $\mathbb{N}_\omega$ ($\equiv \omega+1$)

**Thm.** The ideals of $(X_1\times X_2,\leqslant_\times)$ are exactly the $J_1\times J_2$ for $J_i$ an ideal of $X_i$ ($i=1,2$)

Hence $(\mathit{Idl}(X_1\times X_2),\subseteq) \equiv \mathit{Idl}(X_1,\subseteq)\times\mathit{Idl}(X_2,\subseteq)$ Very nice !!!!

**Coro.** The ideals of $(\mathbb{N}^k,\leqslant_\times)$ are handled like $\mathbb{N}_\omega^k$

**Example:** Assume $U=\uparrow(2,2)$ and $D=\downarrow(4,\omega)\cup\downarrow(6,3)$.
What is $U\setminus D$ and $D\setminus U$?

# IDEALS FOR $(\Sigma^*, \leqslant_*)$?

**Recall:** $\downarrow w$ is an ideal for any $w \in \Sigma^*$.

E.g. $\downarrow abc = \{abc, ab, ac, bc, a, b, c, \epsilon\}$

What else?

- $\Sigma^*$ ?
- $(ab)^* = \{\epsilon, ab, abab, ababab, \ldots\}$ ?
- $a^* + b^* = \{\epsilon, a, aa, aaa, \ldots, b, bb, bbb, \ldots\}$ ?
- $(a + b)^*$ ?

**Lem.** $I \cdot J \in \mathit{Idl}(\Sigma^*)$ for all $I, J \in \mathit{Idl}(\Sigma^*)$

**Thm.** The ideals of $\Sigma^*$ are exactly the concatenation products $P = A_1 \cdot A_2 \cdots A_n$ for atoms of the form $A = \downarrow a = \{a, \epsilon\}$ with $a \in \Sigma$ or $A = \Gamma^*$ with $\Gamma \subseteq \Sigma$.

**Exercise.** Use this to compute $\Sigma^* \setminus \uparrow bad$

# Ideals for $(\Sigma^*, \leqslant_*)$?

**Recall:** $\downarrow w$ is an ideal for any $w \in \Sigma^*$.

E.g. $\downarrow abc = \{abc, ab, ac, bc, a, b, c, \epsilon\}$

What else?

- $\Sigma^*$ ?
- $(ab)^* = \{\epsilon, ab, abab, ababab, \ldots\}$ ?
- $a^* + b^* = \{\epsilon, a, aa, aaa, \ldots, b, bb, bbb, \ldots\}$ ?
- $(a + b)^*$ ?

**Lem.** $I \cdot J \in \mathit{Idl}(\Sigma^*)$ for all $I, J \in \mathit{Idl}(\Sigma^*)$

**Thm.** The ideals of $\Sigma^*$ are exactly the concatenation products $P = A_1 \cdot A_2 \cdots A_n$ for atoms of the form $A = \downarrow a = \{a, \epsilon\}$ with $a \in \Sigma$ or $A = \Gamma^*$ with $\Gamma \subseteq \Sigma$.

**Exercise.** Use this to compute $\Sigma^* \setminus \uparrow bad$

# IDEALS FOR $(\Sigma^*, \leqslant_*)$?

**Recall:** $\downarrow w$ is an ideal for any $w \in \Sigma^*$.

E.g. $\downarrow abc = \{abc, ab, ac, bc, a, b, c, \epsilon\}$

What else?

- $\Sigma^*$ ?
- $(ab)^* = \{\epsilon, ab, abab, ababab, \ldots\}$ ?
- $a^* + b^* = \{\epsilon, a, aa, aaa, \ldots, b, bb, bbb, \ldots\}$ ?
- $(a+b)^*$ ?

**Lem.** $I \cdot J \in \mathit{Idl}(\Sigma^*)$ for all $I, J \in \mathit{Idl}(\Sigma^*)$

**Thm.** The ideals of $\Sigma^*$ are exactly the concatenation products $P = A_1 \cdot A_2 \cdots A_n$ for atoms of the form $A = \downarrow a = \{a, \epsilon\}$ with $a \in \Sigma$ or $A = \Gamma^*$ with $\Gamma \subseteq \Sigma$.

**Exercise.** Use this to compute $\Sigma^* \setminus \uparrow bad$

# IDEALS FOR $(\Sigma^*, \leqslant_*)$?

**Recall:** $\downarrow w$ is an ideal for any $w \in \Sigma^*$.

E.g. $\downarrow abc = \{abc, ab, ac, bc, a, b, c, \epsilon\}$

What else?

- $\Sigma^*$ ?
- $(ab)^* = \{\epsilon, ab, abab, ababab, \ldots\}$ ?
- $a^* + b^* = \{\epsilon, a, aa, aaa, \ldots, b, bb, bbb, \ldots\}$ ?
- $(a + b)^*$ ?

**Lem.** $I \cdot J \in \mathit{Idl}(\Sigma^*)$ for all $I, J \in \mathit{Idl}(\Sigma^*)$

**Thm.** The ideals of $\Sigma^*$ are exactly the concatenation products $P = A_1 \cdot A_2 \cdots A_n$ for atoms of the form $A = \downarrow a = \{a, \epsilon\}$ with $a \in \Sigma$ or $A = \Gamma^*$ with $\Gamma \subseteq \Sigma$.

**Exercise.** Use this to compute $\Sigma^* \setminus \uparrow bad$

# Ideals for $(\Sigma^*, \leqslant_*)$?

**Recall:** $\downarrow w$ is an ideal for any $w \in \Sigma^*$.

E.g. $\downarrow abc = \{abc, ab, ac, bc, a, b, c, \epsilon\}$

What else?

- $\Sigma^*$ ?
- $(ab)^* = \{\epsilon, ab, abab, ababab, \ldots\}$ ?
- $a^* + b^* = \{\epsilon, a, aa, aaa, \ldots, b, bb, bbb, \ldots\}$ ?
- $(a + b)^*$ ?

**Lem.** $I \cdot J \in \mathit{Idl}(\Sigma^*)$ for all $I, J \in \mathit{Idl}(\Sigma^*)$

**Thm.** The ideals of $\Sigma^*$ are exactly the concatenation products $P = A_1 \cdot A_2 \cdots A_n$ for atoms of the form $A = \downarrow a = \{a, \epsilon\}$ with $a \in \Sigma$ or $A = \Gamma^*$ with $\Gamma \subseteq \Sigma$.

**Exercise.** Use this to compute $\Sigma^* \smallsetminus \uparrow bad$

# WHAT DO WE WANT FOR HANDLING $(X,\leqslant)$?

**Def.** $X$ is ideally effective $\overset{\text{def}}{\Leftrightarrow}$

(XR): $X$ is recursive
(OD): $\leqslant$ is decidable over $X$
(IR): $Idl(X)$ is recursive
(ID): $\subseteq$ is decidable over $Idl(X)$
(CF): $F = \uparrow x \mapsto \neg F = X \smallsetminus F = I_1 \cup \cdots \cup I_n$ is recursive
(CI): $I \mapsto \neg I = \uparrow x_1 \cup \cdots \cup \uparrow x_n$ is recursive
(IF) & (II): $F_1, F_2 \mapsto F_1 \cap F_2 = \uparrow x_1 \cup \cdots$ and $I_1, I_2 \mapsto I_1 \cap I_2 = J_1 \cup \cdots$ are recursive
(IM): membership $x \in I$ is decidable over $X$ and $Idl(X)$
(XF) & (XI): $X = F_1 \cup \cdots F_n$ and $X = I_1 \cup \cdots I_m$ are effective
(PI): $x \mapsto \downarrow x$ is recursive

**Examples**: Is $(\mathbb{N},\leqslant)$ ideally effective?
What about $(\Sigma^*,\leqslant_*)$ ?

# What do we want for handling $(X, \leqslant)$?

**Def.** $X$ is ideally effective $\overset{\text{def}}{\Leftrightarrow}$

(XR): $X$ is recursive
(OD): $\leqslant$ is decidable over $X$
(IR): $\mathit{Idl}(X)$ is recursive
(ID): $\subseteq$ is decidable over $\mathit{Idl}(X)$

(CF): $F = \uparrow x \mapsto \neg F = X \setminus F = I_1 \cup \cdots \cup I_n$ is recursive
(CI): $I \mapsto \neg I = \uparrow x_1 \cup \cdots \cup \uparrow x_n$ is recursive
(IF) & (II): $F_1, F_2 \mapsto F_1 \cap F_2 = \uparrow x_1 \cup \cdots$ and $I_1, I_2 \mapsto I_1 \cap I_2 = J_1 \cup \cdots$ are recursive
(IM): membership $x \in I$ is decidable over $X$ and $\mathit{Idl}(X)$
(XF) & (XI): $X = F_1 \cup \cdots F_n$ and $X = I_1 \cup \cdots I_m$ are effective
(PI): $x \mapsto \downarrow x$ is recursive

**Examples**: Is $(\mathbb{N}, \leqslant)$ ideally effective?
What about $(\Sigma^*, \leqslant_*)$ ?

# Valk-Jantzen-Goubault-Larrecq Algorithm

**Thm.** If $(X, \leqslant)$ satisfies the first 4 axioms above and (CF), (II), (PI),(XI) then it is ideally effective.

(XR): $X$ is recursive
(OD): $\leqslant$ is decidable over $X$
(IR): $Idl(X)$ is recursive
(ID): $\subseteq$ is decidable over $Idl(X)$

(CF): $F = \uparrow x \mapsto \neg F = X \smallsetminus F = I_1 \cup \cdots \cup I_n$ is recursive
(CI): $I \mapsto \neg I = \uparrow x_1 \cup \cdots \cup \uparrow x_n$ is recursive
(IF) & (II): $F_1, F_2 \mapsto F_1 \cap F_2 = \uparrow x_1 \cup \cdots$ and $I_1, I_2 \mapsto I_1 \cap I_2 = J_1 \cup \cdots$ are recursive
(IM): membership $x \in I$ is decidable over $X$ and $Idl(X)$
(XF) & (XI): $X = F_1 \cup \cdots F_n$ and $X = I_1 \cup \cdots I_m$ are effective
(PI): $x \mapsto \downarrow x$ is recursive

# Valk-Jantzen-Goubault-Larrecq Algorithm

(XR): $X$ is recursive
(OD): $\leqslant$ is decidable over $X$
(IR): $\mathit{Idl}(X)$ is recursive
(ID): $\subseteq$ is decidable over $\mathit{Idl}(X)$

(CF): $F = \uparrow x \mapsto \neg F = X \smallsetminus F = I_1 \cup \cdots \cup I_n$ is recursive
(CI): $I \mapsto \neg I = \uparrow x_1 \cup \cdots \cup \uparrow x_n$ is recursive
(IF) & (II): $F_1, F_2 \mapsto F_1 \cap F_2 = \uparrow x_1 \cup \cdots$ and $I_1, I_2 \mapsto I_1 \cap I_2 = J_1 \cup \cdots$ are recursive
(IM): membership $x \in I$ is decidable over $X$ and $\mathit{Idl}(X)$
(XF) & (XI): $X = F_1 \cup \cdots F_n$ and $X = I_1 \cup \cdots I_m$ are effective
(PI): $x \mapsto \downarrow x$ is recursive

**Proof.** We first show (CD) $\overset{\text{def}}{\Leftrightarrow}$ that one can design a recursive $D = I_1 \cup \cdots I_n \mapsto \neg D = U = \uparrow x_1 \cup \uparrow x_2 \cup \cdots$
For this, set $U_0 = \emptyset$ and, as long as $\neg U_i \nsubseteq D$, we pick some $x \in \neg U_i \cap \neg D$ and set $U_{i+1} = U_i \cup \uparrow x$. Eventually $U_i = \neg D$ will happen

# VALK-JANTZEN-GOUBAULT-LARRECQ ALGORITHM

(XR): $X$ is recursive
(OD): $\leqslant$ is decidable over $X$
(IR): $\mathit{Idl}(X)$ is recursive
(ID): $\subseteq$ is decidable over $\mathit{Idl}(X)$

(CF): $F = \uparrow x \mapsto \neg F = X \setminus F = I_1 \cup \cdots \cup I_n$ is recursive
(CI): $I \mapsto \neg I = \uparrow x_1 \cup \cdots \cup \uparrow x_n$ is recursive
(IF) & (II): $F_1, F_2 \mapsto F_1 \cap F_2 = \uparrow x_1 \cup \cdots$ and $I_1, I_2 \mapsto I_1 \cap I_2 = J_1 \cup \cdots$ are recursive
(IM): membership $x \in I$ is decidable over $X$ and $\mathit{Idl}(X)$
(XF) & (XI): $X = F_1 \cup \cdots F_n$ and $X = I_1 \cup \cdots I_m$ are effective
(PI): $x \mapsto \downarrow x$ is recursive

**Proof.** Then we get (IF) from (CD) and (CI), by expressing intersection as dual of union, (IM) from (PI) and (ID), (XF) from (CD) by computing $\neg\varnothing$

# CONSTRUCTING IDEALLY EFFECTIVE WQOS

- $(X \times Y, \leqslant_\times)$ is ideally effective when $X$ and $Y$ are.

- $(X^*, \leqslant_*)$ is ideally effective when $X$ is. The ideals are the products of atoms $A = D^*$ for $D \in Down(X)$ and $A = \downarrow I$ for $I \in Idl(X)$

- $(X \sqcup Y, \leqslant_\sqcup)$ is ideally effective when $X$ and $Y$ are. $Idl(X \sqcup Y) \equiv Idl(X) \sqcup Idl(Y)$.

- $X \times_{\text{lex}} Y$ and $X \sqcup_{\text{lex}} Y$ are ideally effective when ..

- $\mathcal{P}_f(X)$ and $\mathcal{M}_f(X)$ are ideally ..

- $\mathcal{T}(X)$ is ideally effective when $X$ is but the ideals are highly complex

# Constructing ideally effective WQOs

- $(X \times Y, \leqslant_\times)$ is ideally effective when $X$ and $Y$ are.

- $(X^*, \leqslant_*)$ is ideally effective when $X$ is. The ideals are the products of atoms $A = D^*$ for $D \in \mathit{Down}(X)$ and $A = {\downarrow}I$ for $I \in \mathit{Idl}(X)$

- $(X \sqcup Y, \leqslant_\sqcup)$ is ideally effective when $X$ and $Y$ are. $\mathit{Idl}(X \sqcup Y) \equiv \mathit{Idl}(X) \sqcup \mathit{Idl}(Y)$.

- $X \times_{\text{lex}} Y$ and $X \sqcup_{\text{lex}} Y$ are ideally effective when ..

- $\mathcal{P}_f(X)$ and $\mathcal{M}_f(X)$ are ideally ..

- $\mathcal{T}(X)$ is ideally effective when $X$ is but the ideals are highly complex

# CONSTRUCTING IDEALLY EFFECTIVE WQOS

- $(X \times Y, \leqslant_\times)$ is ideally effective when $X$ and $Y$ are.

- $(X^*, \leqslant_*)$ is ideally effective when $X$ is. The ideals are the products of atoms $A = D^*$ for $D \in \mathit{Down}(X)$ and $A = \downarrow I$ for $I \in \mathit{Idl}(X)$

- $(X \sqcup Y, \leqslant_\sqcup)$ is ideally effective when $X$ and $Y$ are. $\mathit{Idl}(X \sqcup Y) \equiv \mathit{Idl}(X) \sqcup \mathit{Idl}(Y)$.

- $X \times_{\text{lex}} Y$ and $X \sqcup_{\text{lex}} Y$ are ideally effective when ..

- $\mathcal{P}_f(X)$ and $\mathcal{M}_f(X)$ are ideally ..

- $\mathcal{T}(X)$ is ideally effective when $X$ is but the ideals are highly complex

# Constructing ideally effective WQOs

- $(X \times Y, \leqslant_\times)$ is ideally effective when $X$ and $Y$ are.

- $(X^*, \leqslant_*)$ is ideally effective when $X$ is. The ideals are the products of atoms $A = D^*$ for $D \in \mathit{Down}(X)$ and $A = {\downarrow}I$ for $I \in \mathit{Idl}(X)$

- $(X \sqcup Y, \leqslant_\sqcup)$ is ideally effective when $X$ and $Y$ are. $\mathit{Idl}(X \sqcup Y) \equiv \mathit{Idl}(X) \sqcup \mathit{Idl}(Y)$.

- $X \times_{\mathsf{lex}} Y$ and $X \sqcup_{\mathsf{lex}} Y$ are ideally effective when ..

- $\mathcal{P}_f(X)$ and $\mathcal{M}_f(X)$ are ideally ..

- $\mathcal{T}(X)$ is ideally effective when $X$ is but the ideals are highly complex

# Constructing ideally effective WQOs

- $(X \times Y, \leqslant_\times)$ is ideally effective when $X$ and $Y$ are.

- $(X^*, \leqslant_*)$ is ideally effective when $X$ is. The ideals are the products of atoms $A = D^*$ for $D \in \mathit{Down}(X)$ and $A = \downarrow I$ for $I \in \mathit{Idl}(X)$

- $(X \sqcup Y, \leqslant_\sqcup)$ is ideally effective when $X$ and $Y$ are. $\mathit{Idl}(X \sqcup Y) \equiv \mathit{Idl}(X) \sqcup \mathit{Idl}(Y)$.

- $X \times_{\text{lex}} Y$ and $X \sqcup_{\text{lex}} Y$ are ideally effective when ..

- $\mathcal{P}_f(X)$ and $\mathcal{M}_f(X)$ are ideally ..

- $\mathcal{T}(X)$ is ideally effective when $X$ is but the ideals are highly complex

# CONSTRUCTING IDEALLY EFFECTIVE WQOS

- $(X \times Y, \leqslant_\times)$ is ideally effective when $X$ and $Y$ are.

- $(X^*, \leqslant_*)$ is ideally effective when $X$ is. The ideals are the products of atoms $A = D^*$ for $D \in \mathit{Down}(X)$ and $A = {\downarrow} I$ for $I \in \mathit{Idl}(X)$

- $(X \sqcup Y, \leqslant_\sqcup)$ is ideally effective when $X$ and $Y$ are. $\mathit{Idl}(X \sqcup Y) \equiv \mathit{Idl}(X) \sqcup \mathit{Idl}(Y)$.

- $X \times_{\text{lex}} Y$ and $X \sqcup_{\text{lex}} Y$ are ideally effective when ..

- $\mathcal{P}_f(X)$ and $\mathcal{M}_f(X)$ are ideally ..

- $\mathcal{T}(X)$ is ideally effective when $X$ is but the ideals are highly complex

# Constructing more ideally effective WQOs

- Assume $(X, \leqslant')$ is an extension of $(X, \leqslant)$, i.e., $\leqslant \subseteq \leqslant'$. Then $Idl(X, \leqslant') = \{\downarrow_{\leqslant'} I \mid I \in Idl(X, \leqslant)\}$.

Furthermore $(X, \leqslant')$ is ideally effective when $(X, \leqslant)$ is and the functions

$$I \mapsto \downarrow_{\leqslant'} I = I_1 \cup \cdots \cup I_\ell \quad \text{and} \quad \uparrow x \mapsto \uparrow_{\leqslant'} x = \uparrow x_1 \cup \cdots \cup \uparrow x_m$$

are recursive

**Example.** Subwords *cum* conjugacy:
$abcd \leqslant_{\circlearrowleft} acbadbbdbdbdbadbc$

Now a new research program :

- Characterize ideals
- Find algorithms for ideally effective wqos
- Find smarter algorithms and data structures

# Constructing more ideally effective WQOs

- Assume $(X, \leqslant')$ is an extension of $(X, \leqslant)$, i.e., $\leqslant \subseteq \leqslant'$. Then $Idl(X, \leqslant') = \{\downarrow_{\leqslant'} I \mid I \in Idl(X, \leqslant)\}$.

Furthermore $(X, \leqslant')$ is ideally effective when $(X, \leqslant)$ is and the functions

$$I \mapsto \downarrow_{\leqslant'} I = I_1 \cup \cdots \cup I_\ell \quad \text{and} \quad \uparrow x \mapsto \uparrow_{\leqslant'} x = \uparrow x_1 \cup \cdots \cup \uparrow x_m$$

are recursive

**Example.** Subwords *cum* conjugacy:
$\mathtt{abcd} \leqslant_{\circlearrowleft} \mathtt{acbadbbdbdbdbadbc}$

Now a new research program :

- Characterize ideals
- Find algorithms for ideally effective wqos
- Find smarter algorithms and data structures