MPRI 2-9-1
“Algorithmic Aspects of WQO Theory”
Nov. 19th, 2020: Ideals & Generic algorithms for downward-closed sets
Some recalls from previous weeks

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

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

**Verification of WSTS.** It is possible to decide Safety, Termination, etc., for systems with well-quasi-ordered states and monotonic (aka compatible) steps.

**Today's class:** WQO-based algorithms often have to handle/reason about/infinite upward- or downward-closed sets

- This is a non-trivial problem
- But there exists a powerful & generic approach via ideals
**Some calls from previous weeks**

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

**Verification of WSTS.** It is possible to decide Safety, Termination, etc., for systems with well-quasi-ordered states and monotonic (aka compatible) steps.

**Today’s class:** *WQO-based algorithms often have to handle/reason about/infinite upward-or downward-closed sets*

- This is a non-trivial problem
- But there exists a powerful & generic approach via ideals
OUTLINE FOR TODAY

- 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**

Verifying safety for a WSTS is usually done by computing upward-closed subsets

\[ B \subseteq Pre^{\leq 1}(B) \subseteq Pre^{\leq 2}(B) \subseteq \cdots \subseteq \bigcup_{m} Pre^{\leq m}(B) = Pre^*(B) \]

How is this implemented in practice?

Consider \((\mathbb{N}^2, \leq x)\) and upward-closed subsets \(U, U', V, \ldots\)

There is the finite basis presentation:

\[ U = \begin{array}{c}
\end{array} \]
HANDLING UPWARD-CLOSED SUBSETS

Verifying safety for a WSTS is usually done by computing upward-closed subsets.

How is this implemented in practice?

Consider \((\mathbb{N}^2, \preceq)\) and upward-closed subsets \(U, U', V, \ldots\)

There is the finite basis presentation:

\[
U = \uparrow(2,6) \cup \uparrow(4,5) \cup \uparrow(6,1) \cup \uparrow(10,0)
\]

We also need algorithms for computing with this representation:
- E.g., testing whether \(U \subseteq V\)
- E.g., performing \(U \cap U' \cap V\)
Handling upward-closed subsets

How is this implemented in practice?

Consider \((\mathbb{N}^2, \leq_x)\) and upward-closed subsets \(U, U', V, \ldots\)

There is the finite basis presentation:

\[
U = \uparrow(2,6) \cup \uparrow(4,5) \cup \uparrow(6,1) \cup \uparrow(10,0)
\]

We also need algorithms for computing with this representation:

- E.g., testing whether \(U \subseteq V\)
- E.g., performing \(U \leftarrow U \cup V\) or \(U \leftarrow U \cap V\)
How is this implemented in practice?

Consider \((\mathbb{N}^2, \leq_x)\) and upward-closed subsets \(U, U', V, \ldots\)

There is the finite basis presentation:

\[
U = \uparrow(2,6) \cup \uparrow(4,5) \cup \uparrow(6,1) \cup \uparrow(10,0)
\]

We also need algorithms for computing with this representation:

- E.g., testing whether \(U \subseteq V\)
- E.g., performing \(U \leftarrow U \cup V\) or \(U \leftarrow U \cap V\)
**Upward-Closed Subsets of \((\Sigma^*, \leq^*)\)**

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

\[
U \leftarrow U \cap \uparrow cbab \text{ or } U \leftarrow U \setminus \downarrow baccbab
\]

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

- Can we handle \(\mathbb{N}^k\) and \(\Sigma^*\) efficiently?
- What about other WQOs? E.g. over \((\mathbb{N}^2)^*\):
  \[
  \uparrow (\begin{pmatrix} 2 \\ 0 \end{pmatrix}) \cap \uparrow (\begin{pmatrix} 1 \\ 1 \end{pmatrix})
  \]
Upward-Closed Subsets of \((\Sigma^*, \preceq^*)\)

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

\[ U \leftarrow U \cap \uparrow cbab \quad \text{or} \quad U \leftarrow U \setminus \downarrow bacbab \]

Bottom line: These are feasible but not trivial!

- Can we handle \(\mathbb{N}^k\) and \(\Sigma^*\) efficiently?
- What about other WQOs? E.g. over \((\mathbb{N}^2)^*\): \(\uparrow(\begin{pmatrix} 2 \\ 0 \end{pmatrix}) \cap \uparrow(\begin{pmatrix} 1 \\ 1 \end{pmatrix})\)
UPWARD-CLOSED SUBSETS OF \((\Sigma^* , \leq^*)\)

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

\[
U = \uparrow abc \cup \cdots \cup \uparrow ddc \quad V = \uparrow bb \cup \cdots
\]

How do we compare such sets?
How do we add to them?
How do we remove from them? E.g., how do we perform
\[
U \leftarrow U \cap \uparrow cbab \text{ or } U \leftarrow U \setminus \downarrow baccbab
\]?

Bottom line: These are feasible but not trivial!

- Can we handle \(\mathbb{N}^k\) and \(\Sigma^*\) efficiently?
- What about other WQOs? E.g. over \((\mathbb{N}^2)^*\): \(\uparrow (|2,0|) \cap \uparrow (|1,1|)\)
Let us consider words with subword ordering, e.g., for lossy channel systems:

\[ \mathcal{U} = \uparrow abc \cup \cdots \cup \uparrow ddca \quad \mathcal{V} = \uparrow bb \cup \cdots \]

How do we compare such sets?

How do we add to them?

How do we remove from them? E.g., how do we perform \( \mathcal{U} \leftarrow \mathcal{U} \cap \uparrow cbab \) or \( \mathcal{U} \leftarrow \mathcal{U} \setminus \downarrow baccbab \)?

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

- Can we handle \( \mathbb{N}^k \) and \( \Sigma^* \) efficiently?
- What about other WQOs? E.g. over \( (\mathbb{N}^2)^* \): \( \uparrow (|2 \ 0 | \ 0 \ 2) \cap \uparrow (|1 \ 1 | \ 0 \ 0) \)
**Now what about downward-closed subsets?**

**Problem:** downward-closed $D$ can’t always be represented under the form $D = \downarrow x_1 \cup \cdots \cup \downarrow x_\ell$, take e.g. $D = \mathbb{N}^2$.

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

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

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 $\downarrow (2,2)$ in $(\mathbb{N}^2, \leq_X)$? And $\downarrow ab$ in $(\Sigma^*, \leq_*)$?

$$\downarrow (2,2) = \neg [\uparrow (0,3) \cup \uparrow (3,0)] \quad \downarrow ab = \neg [\uparrow ba \cup \uparrow c \cup \cdots]$$

— How do you compute $D \cup D'$?

There is a better solution: decompose into primes!
Now what about downward-closed subsets?

**Problem:** downward-closed $D$ can’t always be represented under the form $D = \downarrow x_1 \cup \cdots \cup \downarrow x_\ell$, take e.g. $D = \mathbb{N}^2$.

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

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

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 $\downarrow (2, 2)$ in $(\mathbb{N}^2, \leq_x)$? And $\downarrow ab$ in $(\Sigma^*, \leq_*)$?

$$\downarrow (2, 2) = \neg [\uparrow (0, 3) \cup \uparrow (3, 0)] \\
\downarrow ab = \neg [\uparrow ba \cup \uparrow c \cup \cdots]$$

— How do you compute $D \cup D'$?

There is a better solution: decompose into primes!
**Now what about downward-closed subsets?**

**Problem:** downward-closed $D$ can’t always be represented under the form $D = \downarrow x_1 \cup \cdots \cup \downarrow x_\ell$, take e.g. $D = \mathbb{N}^2$.

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

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

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 $\downarrow (2, 2)$ in $(\mathbb{N}^2, \leq_x)$? And $\downarrow ab$ in $(\Sigma^*, \leq_*)$?

$$\downarrow (2, 2) = \neg [\uparrow (0, 3) \cup \uparrow (3, 0)] \quad \downarrow ab = \neg [\uparrow ba \cup \uparrow c \cup \cdots]$$

— How do you compute $D \cup D'$?

There is a better solution: decompose into primes!
**NOW WHAT ABOUT DOWNWARD-CLOSED SUBSETS?**

**Problem:** downward-closed $D$ *can’t always* be represented under the form $D = \downarrow x_1 \cup \cdots \cup \downarrow x_\ell$, take e.g. $D = \mathbb{N}^2$.

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

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

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 $\downarrow (2, 2)$ in $(\mathbb{N}^2, \leq_x)$? And $\downarrow ab$ in $(\Sigma^*, \leq_*)$?

$$\downarrow (2, 2) = \neg [\uparrow (0, 3) \cup \uparrow (3, 0)]$$

$$\downarrow ab = \neg [\uparrow ba \cup \uparrow c \cup \cdots]$$

— How do you compute $D \cup D'$?

There is a better solution: decompose into primes!
PRIMES, UP AND DOWN

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

**Def. 4.1.** 1. $U (\neq \emptyset)$ is (up-) prime $\iff U \subseteq (U_1 \cup U_2)$ implies $U \subseteq U_1$ or $U \subseteq U_2$.
2. $D (\neq \emptyset)$ is (down-) prime $\iff D \subseteq (D_1 \cup D_2)$ 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. 4.2. (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. 4.3. (Completeness: Prime Decompositions Exist)**
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 $\iff \forall i, j : P_i \subseteq P_j$ implies $i = j$.

**Thm. 4.4.** Every $U$ (or $D$) has a unique minimal prime decomposition. It is called its **canonical decomposition**

**Prop. 4.8.** (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 $\iff x, y \in I \implies \exists z \in I : x \leq z \geq y$

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: Let us look at $\neg U$ for our earlier $U \subseteq \mathbb{N}^2$
**Minimal Prime Decompositions**

**Def.** A prime decomposition \( U \) (or \( D \)) = \( P_1 \cup \cdots \cup P_n \) is **minimal**

\[ \iff \forall i, j : P_i \subseteq P_j \implies i = j. \]

**Thm. 4.4.** Every \( U \) (or \( D \)) has a unique minimal prime decomposition. It is called its **canonical decomposition**

**Prop. 4.8.** (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** \( \iff x, y \in I \implies \exists z \in I : x \leq z \geq y \)

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: Let us look at \( \neg U \) for our earlier \( U \subseteq \mathbb{N}^2 \)
A DOWNWARD-CLOSED SUBSET OF $\mathbb{N}^2$

$u =$
A DOWNWARD-CLOSED SUBSET OF $\mathbb{N}^2$

\[ D = \neg \cup = \]
A downward-closed subset of $\mathbb{N}^2$

\[ D = I_1 \cup \cdots \cup I_4 \]
NAILING DOWN THE IDEALS

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

Hence \((\text{Idl}(\mathbb{N}), \subseteq) \equiv (\mathbb{N} \cup \{\omega\}, \leq)\), denoted \(\mathbb{N}_\omega (\equiv \omega + 1)\).

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

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

**Coro.** The ideals of \((\mathbb{N}^k, \leq_x)\) 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 \((\mathbb{N}, \leq)\) are exactly all \(\downarrow n\) together with \(\mathbb{N}\) itself

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

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

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

**Coro.** The ideals of \((\mathbb{N}^k, \leq_X)\) 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 \((\mathbb{N}, \leq)\) are exactly all \(\downarrow n\) together with \(\mathbb{N}\) itself

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

**Thm.** The ideals of \((X_1 \times X_2, \leq_x)\) 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 \((\mathbb{N}^k, \leq_x)\) 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 \((\Sigma^*, \leq^*)\)?

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

*E.g.* \(\downarrow abc = \{abc, ab, ac, bc, a, b, c, \varepsilon\}\)

What else?

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

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

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

**Exercise.** Use this to compute \(\Sigma^* \setminus \uparrow \text{bad}\)
IDEALS FOR \((\Sigma^*, \leq *)\)?

**Recall:** \(\downarrow w\) is an ideal for any \(w \in \Sigma^*\).
E.g. \(\downarrow abc = \{abc, ab, ac, bc, a, b, c, \varepsilon\}\)

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

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

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

**Exercise.** Use this to compute \(\Sigma^* \setminus \uparrow bad\)
IDEALS FOR \((\Sigma^*, \leq^*)\)?

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

E.g. \(\downarrow abc = \{abc, ab, ac, bc, a, b, c, \varepsilon\}\)

What else?

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

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

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

**Exercise.** Use this to compute \(\Sigma^* \setminus \uparrow bad\)
IDEALS FOR \((\Sigma^*, \leq^*)\)?

Recall: \(\downarrow w\) is an ideal for any \(w \in \Sigma^*\).
E.g. \(\downarrow abc = \{abc, ab, ac, bc, a, b, c, \varepsilon\}\)

What else?

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

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

Thm. The ideals of \(\Sigma^*\) are exactly the concatenation products \(P = \Lambda_1 \cdot \Lambda_2 \cdots \Lambda_n\) for atoms of the form \(\Lambda = \downarrow a = \{a, \varepsilon\}\) with \(a \in \Sigma\) or \(\Lambda = \Gamma^*\) with \(\Gamma \subseteq \Sigma\).

Exercise. Use this to compute \(\Sigma^* \setminus \uparrow \text{bad}\)
IDEALS FOR \((\Sigma^*, \leq^*)\)?

Recall: \(\downarrow w\) is an ideal for any \(w \in \Sigma^*\).
E.g. \(\downarrow abc = \{abc, ab, ac, bc, a, b, c, \varepsilon\}\)

What else?

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

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

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

Exercise. Use this to compute \(\Sigma^* \setminus \uparrow bad\)
WHAT IS REQUIRED FOR HANDLING \((X, \leq)\)?

**Def.** \(X\) is ideally effective \(\overset{\text{def}}{\iff}\)

(XR): \(X\) is recursive
(OR): \(\leq\) is decidable over \(X\)
(IR): \(\text{Idl}(X)\) is recursive
(II): \(\subseteq\) is decidable over \(\text{Idl}(X)\)

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

**Examples:** Is \((\mathbb{N}, \leq)\) ideally effective?
What about \((\Sigma^*, \leq^*)\)?
What is required for handling \((X, \preceq)\)?

**Def.** \(X\) is ideally effective \(\text{def} \iff \)

(XR): \(X\) is recursive  
(OR): \(\preceq\) is decidable over \(X\)  
(IR): \(\text{Idl}(X)\) is recursive  
(II): \(\subseteq\) is decidable over \(\text{Idl}(X)\)

(CF): \(F = \uparrow x \iff \neg F = X \setminus F = I_1 \cup \cdots \cup I_n\) is recursive  
(CI): \(I \iff \neg I = \uparrow x_1 \cup \cdots \cup \uparrow x_n\) is recursive

(IF) & (II): \(F_1, F_2 \iff F_1 \cap F_2 = \uparrow x_1 \cup \cdots\) and \(I_1, I_2 \iff I_1 \cap I_2 = J_1 \cup \cdots\) are recursive

(IM): membership \(x \in I\) is decidable over \(X\) and \(\text{Idl}(X)\)

(XF) & (XI): \(X = F_1 \cup \cdots F_n\) and \(X = I_1 \cup \cdots I_m\) are effective  
(PI): \(x \iff \downarrow x\) is recursive

**Examples:** Is \((\mathbb{N}, \preceq)\) ideally effective?  
What about \((\Sigma^*, \preceq_*)\)?
**Thm.** If \((X, \leq)\) satisfies the first 4 axioms above and (CF), (II), (PI), (XI) then it is ideally effective.

(XR): \(X\) is recursive
(OR): \(\leq\) is decidable over \(X\)
(IR): \(Idl(X)\) is recursive
(II): \(\subseteq\) is decidable over \(Idl(X)\)

(CF): \(F = \uparrow x \iff \neg F = X \setminus F = I_1 \cup \cdots \cup I_n\) is recursive
(CI): \(I \iff \neg I = \uparrow x_1 \cup \cdots \cup \uparrow x_n\) is recursive
(IF) & (II): \(F_1, F_2 \iff F_1 \cap F_2 = \uparrow x_1 \cup \cdots\) and \(I_1, I_2 \iff I_1 \cap I_2 = J_1 \cup \cdots\) 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): \(x \iff \downarrow x\) is recursive
\textbf{VALK-JANTZEN-GOUBAULT-LARRECQ ALGORITHM}

(XR): $X$ is recursive
(OR): $\leq$ is decidable over $X$
(IR): $\text{Idl}(X)$ is recursive
(II): $\subseteq$ is decidable over $\text{Idl}(X)$

(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 $\text{Idl}(X)$
(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

\textbf{Proof.} We first show (CD) $\overset{\text{def}}{\iff}$ 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 $D \not\subseteq \neg U_i$, we pick some $x$ s.t.

$D \not\ni x \not\ni U_i$ 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
(OR): ≤ is decidable over X
(IR): $\text{Idl}(X)$ is recursive
(II): ≤ is decidable over $\text{Idl}(X)$

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

Proof. Then we get (IF) from (CD) and (CI), by expressing intersection as dual of union, (IM) from (PI) and (II), (XF) from (CD) by computing $\neg \emptyset$
Valk-Jantzen-Goubault-Larrecq Algorithm

(XR): \( X \) is recursive
(OR): \( \leq \) is decidable over \( X \)
(IR): \( \text{Idl}(X) \) is recursive
(II): \( \subseteq \) is decidable over \( \text{Idl}(X) \)

(CF): \( F = \uparrow x \iff \neg F = X \setminus F = I_1 \cup \cdots \cup I_n \) 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 \( \text{Idl}(X) \)
(XF) & (XI): \( X = F_1 \cup \cdots \cup F_n \) and \( X = I_1 \cup \cdots \cup I_m \) are effective
(PI): \( x \mapsto \downarrow x \) is recursive

Thm [Halfon]. There are no more redundancies in the blue axioms
**CONSTRUCTING IDEALLY EFFECTIVE WQOs**

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

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

- \((X \sqcup Y, \leq_{\sqcup})\) is ideally effective when \(X\) and \(Y\) are.\[\text{Idl}(X \sqcup Y) \equiv \text{Idl}(X) \sqcup \text{Idl}(Y)\].

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

- \(P_f(X)\) and \(M_f(X)\) and \((X^*, \leq_{\text{st}})\) and \(\cdots\) are ideally ..

- \(\mathcal{T}(X)\) is ideally effective when \(X\) is but the ideals are more complex (see Goubault-Larrecq & Schmitz, ICALP 2016)
CONSTRUCTING IDEALLY EFFECTIVE WQOS

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

- \((X^*, \leq_*)\) 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 \(I \in Idl(X)\).

- \((X \sqcup Y, \leq_\sqcup)\) is ideally effective when \(X\) and \(Y\) are. 
  \(Idl(X \sqcup Y) = Idl(X) \sqcup Idl(Y)\).

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

- \(\mathcal{P}_f(X)\) and \(\mathcal{M}_f(X)\) and \((X^*, \leq_{st})\) and \(\cdots\) are ideally ..

- \(\mathcal{T}(X)\) is ideally effective when \(X\) is but the ideals are more complex (see Goubault-Larrecq & Schmitz, ICALP 2016)
**CONSTRUCTING IDEALLY EFFECTIVE WQOs**

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

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

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

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

- $P_f(X)$ and $M_f(X)$ and $(X^*, \leq_{st})$ and ⋯ are ideally ..

- $\mathcal{I}(X)$ is ideally effective when $X$ is but the ideals are more complex (see Goubault-Larrecq & Schmitz, ICALP 2016)
CONSTRUCTING IDEALLY EFFECTIVE WQOS

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

- \((X^*, \leq^*_*)\) 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 \(I \in Idl(X)\)

- \((X \sqcup Y, \leq \sqcup)\) 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 ..

- \(P_f(X)\) and \(M_f(X)\) and \((X^*, \leq_{st})\) and \(\cdots\) are ideally ..

- \(\mathcal{T}(X)\) is ideally effective when \(X\) is but the ideals are more complex (see Goubault-Larrecq & Schmitz, ICALP 2016)
CONSTRUCTING IDEALLY EFFECTIVE WQOs

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

- \((X^*, \leq_*)\) 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 \(I \in Idl(X)\).

- \((X \sqcup Y, \leq_\sqcup)\) 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 ..

- \(P_f(X)\) and \(M_f(X)\) and \((X^*, \leq_{st})\) and \(\cdots\) are ideally..

- \(\mathcal{I}(X)\) is ideally effective when \(X\) is but the ideals are more complex (see Goubault-Larrecq & Schmitz, ICALP 2016).
CONSTRUCTING IDEALLY EFFECTIVE WQOs

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

- \((X^*, \leq_st)\) 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 \(I \in Idl(X)\).

- \((X \sqcup Y, \leq_{\sqcup})\) 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 ..

- \(P_f(X)\) and \(M_f(X)\) and \((X^*, \leq_st)\) and \(\cdots\) are ideally ..

- \(\mathcal{I}(X)\) is ideally effective when \(X\) is but the ideals are more complex (see Goubault-Larrecq & Schmitz, ICALP 2016).
CONSTRUCTING MORE IDEALLY EFFECTIVE WQOS

1. Assume \((X, \leq')\) is an extension of \((X, \leq)\), i.e., \(\leq \subseteq \leq'\).
Then \(Idl(X, \leq') = \{\downarrow \leq, I \mid I \in Idl(X, \leq)\}\).

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

\[ I \mapsto \downarrow \leq', I = I_1 \cup \cdots \cup I_\ell \quad \text{and} \quad \uparrow x = F \mapsto \uparrow \leq', F = \uparrow x_1 \cup \cdots \cup \uparrow x_m \]

are recursive.

**Example.** Subwords *cum* conjugacy:

\[ \text{abcd} \leq_{\circ} \text{acbadbdbhdbdbadb} \]

**Example.** Quotienting \((X, \leq)\) by some equivalence \(\approx\) such that \(\approx \circ \leq = \leq \circ \approx\)
1. Assume \((X, \leq')\) is an extension of \((X, \leq)\), i.e., \(\leq \subseteq \leq'\).

Then \(\text{Idl}(X, \leq') = \{\downarrow \leq', I \mid I \in \text{Idl}(X, \leq)\}\).

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

\[
I \mapsto \downarrow \leq', I = I_1 \cup \cdots \cup I_\ell \quad \text{and} \quad \uparrow x = F \mapsto \uparrow \leq', F = \uparrow x_1 \cup \cdots \cup \uparrow x_m
\]

are recursive.

**Example.** Subwords *cum* conjugacy:

\[
abcd \leq Q \quad acbadbbdbdbdbadbac
\]

**Example.** Quotienting \((X, \leq)\) by some equivalence \(\approx\) such that

\[
\approx \circ \leq = \leq \circ \approx
\]
CONSTRUCTING MORE IDEALLY EFFECTIVE WQOS

2. Assume \((Y, \leq_Y)\) is a subwqo of \((X, \leq_X)\), i.e., \(Y \subseteq X\) and 
\[ \leq_Y = \leq_X \cap Y \times Y. \]

Then \(Idl(Y, \leq) = \{I \cap Y \mid I \in Idl(X)\ \text{st.} \ I \subseteq \downarrow_X Y \land I \cap Y \neq \emptyset\}\).

Furthermore \((Y, \leq)\) is ideally effective when \((X, \leq)\) is and when \(Y\) and the functions

\[
Idl(X) \rightarrow \text{Down}(X) \\
I \mapsto \downarrow_X (I \cap Y) = I_1 \cup \cdots I_\ell \\
\text{and} \\
Fil(X) \rightarrow \text{Up}(X) \\
\uparrow x = F \mapsto \uparrow_X (F \cap Y) = \uparrow x_1 \cup \cdots \uparrow x_m
\]

are recursive.

Example. \((L, \leq_\ast)\) for a context-free \(L \subseteq \Sigma^\ast\).

Example. Decreasing sequences in \(\mathbb{N}^\ast\) with the subsequence ordering.
2. Assume \((Y, \leq_Y)\) is a subwqo of \((X, \leq_X)\), i.e., \(Y \subseteq X\) and 
\[\leq_Y = \leq_X \cap Y \times Y.\]

Then \(Idl(Y, \leq) = \{I \cap Y \mid I \in Idl(X) \text{ st. } I \subseteq \downarrow_X Y \land I \cap Y \neq \emptyset\}\).

Furthermore, \((Y, \leq)\) is ideally effective when \((X, \leq)\) is and when \(Y\) and the functions 
\[Idl(X) \rightarrow Down(X) \quad \text{and} \quad Fil(X) \rightarrow Up(X)\]

are recursive.

**Example.** \((L, \leq_*)\) for a context-free \(L \subseteq \Sigma^*\).

**Example.** Decreasing sequences in \(\mathbb{N}^*\) with the subsequence ordering.
CONCLUSION FOR PART V

Ideal-based algorithms already have several applications.

Handling WQO’s raise many interesting algorithmic questions:

- Best algorithms for \((\Sigma^*, \leq_*)\)? (Karandikar et al., TCS 2016)
- Best algorithms for \((\mathbb{N}^k)^*\)?
- Fully generic library of data structures and algorithms?
- Separating the polynomial and the exponential cases?
- More constructions .. Beyond WQOs ..
- …