

Ideally Effective WQOs

Simon Halfon

Joint work with J. Goubault-Larrecq, P. Karandikar, K. Narayan Kumar
and P. Schnoebelen

LSV, ENS Cachan

BRAVAS

Motivation

- Up-sets:

$$U = F_1 \cup F_2 \cup \dots \cup F_n$$

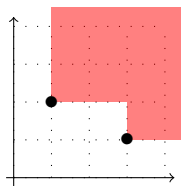
Motivation

- Up-sets:

$$U = F_1 \cup F_2 \cup \dots \cup F_n$$

WQO \Rightarrow filters = principal

filters = $\uparrow x$



$$U = \uparrow(1, 2) \cup \uparrow(3, 1)$$

Motivation

- Up-sets:

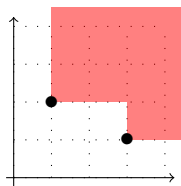
$$U = F_1 \cup F_2 \cup \dots \cup F_n$$

WQO \Rightarrow filters = principal

filters = $\uparrow x$

- Down-sets:

$$D = I_1 \cup I_2 \cup \dots \cup I_n$$



$$U = \uparrow(1, 2) \cup \uparrow(3, 1)$$

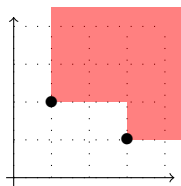
Motivation

- Up-sets:

$$U = F_1 \cup F_2 \cup \dots \cup F_n$$

WQO \Rightarrow filters = principal

filters = $\uparrow x$

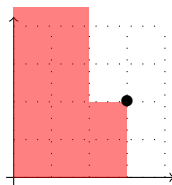


$$U = \uparrow(1, 2) \cup \uparrow(3, 1)$$

- Down-sets:

$$D = I_1 \cup I_2 \cup \dots \cup I_n$$

Principal ideals are ideals: $\downarrow x$,
but not only.



$$D = \downarrow(2, \omega) \cup \downarrow(3, 2)$$

Definition

A WQO (X, \leq) is Ideally Effective if:

- (X, \leq) is effective
- $(Idl(X), \subseteq)$ is effective
- Complements and Intersections of filters and ideals are computable

Definition

A WQO (X, \leq) is Ideally Effective if:

- (X, \leq) is effective
- $(Idl(X), \subseteq)$ is effective
- Complements and Intersections of filters and ideals are computable
- Up-sets = lists of elements of X .

Definition

A WQO (X, \leq) is Ideally Effective if:

- (X, \leq) is effective
 - $(Idl(X), \subseteq)$ is effective
 - Complements and Intersections of filters and ideals are computable
-
- Up-sets = lists of elements of X .
 - Down-sets = lists of ideals.

Definition

A WQO (X, \leq) is Ideally Effective if:

- (X, \leq) is effective
 - $(\text{Idl}(X), \subseteq)$ is effective
 - Complements and Intersections of filters and ideals are computable
-
- Up-sets = lists of elements of X .
 - Down-sets = lists of ideals.
 - Compute minimal representations:
 $\uparrow(1, 2) \cup \uparrow(2, 1) \cup \uparrow(2, 2) = \uparrow(1, 2) \cup \uparrow(2, 1)$
 - Lift all set-theoretic operations to Up- and Down-sets.

Examples

(VASS):

\mathbb{N}^d

\Rightarrow

ω -vectors

Examples

(VASS): \mathbb{N}^d \Rightarrow ω -vectors

(Channel Systems): A^* \Rightarrow SRE

Examples

(VASS): \mathbb{N}^d \Rightarrow ω -vectors

(Channel Systems): A^* \Rightarrow SRE

(Universality Problem): $\mathcal{P}_f(\mathbb{N}^d)$ $\Rightarrow \mathcal{P}_f(\text{Down}(\mathbb{N}^d)) \equiv \mathcal{P}_f(\mathbb{A}(\mathbb{N}_\omega^d))$

Examples

(VASS): \mathbb{N}^d \Rightarrow ω -vectors

(Channel Systems): A^* \Rightarrow SRE

(Universality Problem): $\mathcal{P}_f(\mathbb{N}^d)$ $\Rightarrow \mathcal{P}_f(\text{Down}(\mathbb{N}^d)) \equiv \mathcal{P}_f(\mathbb{A}(\mathbb{N}_\omega^d))$

(Timed Petri Nets): $(\mathbb{N}^d)^*$ \Rightarrow SRE of ω vectors

Examples

(VASS): \mathbb{N}^d \Rightarrow ω -vectors

(Channel Systems): A^* \Rightarrow SRE

(Universality Problem): $\mathcal{P}_f(\mathbb{N}^d)$ $\Rightarrow \mathcal{P}_f(\text{Down}(\mathbb{N}^d)) \equiv \mathcal{P}_f(\mathbb{A}(\mathbb{N}_\omega^d))$

(Timed Petri Nets): $(\mathbb{N}^d)^*$ \Rightarrow SRE of ω vectors

(Priority Channel): $\mathbf{0} \left. \begin{matrix} * \cdots * \\ \vdots \\ * \end{matrix} \right\} d$ \Rightarrow SRE of SRE

Examples

(VASS): \mathbb{N}^d \Rightarrow ω -vectors

(Channel Systems): A^* \Rightarrow SRE

(Universality Problem): $\mathcal{P}_f(\mathbb{N}^d)$ $\Rightarrow \mathcal{P}_f(\text{Down}(\mathbb{N}^d)) \equiv \mathcal{P}_f(\mathbb{A}(\mathbb{N}_\omega^d))$

(Timed Petri Nets): $(\mathbb{N}^d)^*$ \Rightarrow SRE of ω vectors

(Priority Channel): $\mathbf{0} \left. \begin{matrix} * \cdots * \\ \vdots \\ * \end{matrix} \right\} d$ \Rightarrow SRE of SRE

(Universality Problem): $\text{Down}(\mathbb{N}^d)$
 $\equiv \mathbb{A}(\mathbb{N}_\omega^d)$ \Rightarrow $\mathbb{A}(\mathbb{A}(\omega + 2))$

“Natural” WQOs are ideally effective

Base Cases:

Finite orderings and Linear orderings

Inductive Cases:

(X, \leq_X) , (Y, \leq_Y) ideally effective WQO. The following are ideally effective:

- $(X \times Y, \leq_X \times \leq_Y)$
- (X^*, \leq_*)
- $(\mathcal{P}_f(X), \leq_H)$
- $(X^{\otimes}, \leq_{\text{emb}})$
- $(X^{\otimes}, \leq_{\text{ms}})$
- $(X \times Y, \leq_{\text{lex}})$
- $(X \sqcup Y, \leq_X \sqcup \leq_Y)$
- $(X \oplus Y, \leq_X \oplus \leq_Y)$

And algorithms for set-theoretic operations can be computed from algorithms for set-theoretic operations in X and Y (OCaml Functor).

“Semi-natural” WQOs are ideally effective as well

(X, \leq) is an ideally effective WQO:

- $\leq \subseteq \leq' \Rightarrow (X, \leq')$ is a WQO.

Under extra assumptions, it is ideally effective.

“Semi-natural” WQOs are ideally effective as well

(X, \leq) is an ideally effective WQO:

- $\leq \subseteq \leq' \Rightarrow (X, \leq')$ is a WQO.
Under extra assumptions, it is ideally effective.
- Particular case: Quotients.
E.g. $(X^{\otimes}, \leq_{\text{emb}}) = (X^*/\simeq, \leq_* \circ \simeq)$.

“Semi-natural” WQOs are ideally effective as well

(X, \leq) is an ideally effective WQO:

- $\leq \subseteq \leq' \Rightarrow (X, \leq')$ is a WQO.
Under extra assumptions, it is ideally effective.
- Particular case: Quotients.
E.g. $(X^{\otimes}, \leq_{\text{emb}}) = (X^* / \simeq, \leq_* \circ \simeq)$.
- $Y \subseteq X \Rightarrow (Y, \leq)$ is a WQO.
Under extra assumptions, it is ideally effective.

“Semi-natural” WQOs are ideally effective as well

(X, \leq) is an ideally effective WQO:

- $\leq \subseteq \leq' \Rightarrow (X, \leq')$ is a WQO.
Under extra assumptions, it is ideally effective.
- Particular case: Quotients.
E.g. $(X^{\otimes}, \leq_{\text{emb}}) = (X^*/\simeq, \leq_* \circ \simeq)$.
- $Y \subseteq X \Rightarrow (Y, \leq)$ is a WQO.
Under extra assumptions, it is ideally effective.

Applications:

- Words with stuttering: $aaddaaabba \leq_{\text{st}} \text{abracadabra}$
- Words on a circle: $baaada \leq_{\text{st}} \text{abracadabra}$

- Inclusion is always decidable in polynomial time.
- Sums, Products, Finite Powerset preserve polynomial time.
- Finite Sequences and Multisets don't.

Conclusion

- Effective toolbox for manipulating up/down sets of many WQOs, encountered in particular in verification
- Next step: Trees and Graphs.
- Effective FAC ?