Some Properties of Inclusions of Multisets and Strictly Increasing Boolean Functions

Pierre Hyvernat

Some Properties of Inclusions of Multisets and Strictly Increasing Boolean Functions

Spring 2011

*Pierre Hyvernat** pierre.hyvernat@univ-savoie.fr
http://lama.univ-savoie.fr/~hyvernat/

Abstract. This short paper contains the result of a small investigation into a curious puzzle: call an n-tuple of sets $\overline{X} = (X_1, \dots, X_n)$ smaller than $\overline{Y}$ if it has less *unordered sections*. We show that equivalence classes up-to this preorder are very easy to describe (read on to get the answer!) and characterize the preorder in terms of the simpler pointwise inclusion. This gives a simple algorithm for comparing n-tuples of sets. An interesting point is that part of this work relies on the notion of *strictly increasing boolean function*, which doesn't appear in the traditional literature (??). As an extra, we also show that contrary to plain boolean functions, strictly increasing boolean functions aren't finitely generated.

0. Introduction: a puzzle. Choose a (fixed) set N and a (fixed) natural number n. Pointwise inclusion makes $\mathcal{P}_*(N)^n$, the n-tuples of *non-empty* subsets of N, into a partial order:

$$(X_1, \dots, X_n) \subseteq (Y_1, \dots, Y_n) \quad \stackrel{\text{def}}{=} \quad \forall 1 \leq i \leq n,\ X_i \subseteq Y_i \ .$$

Because we restrict to non-empty subsets, we have

$$\overline{X} \subseteq \overline{Y} \qquad \Leftrightarrow \qquad \prod_{1\leq i\leq n} X_i \subseteq \prod_{1\leq i\leq n} Y_i \ .$$

Consider now a commutative version of the cartesian product where instead of the usual ordered n-tuples, we take "unordered n-tuples".

Definition 1. *If $\overline{X} = (X_1, \dots, X_n)$ is an n-tuple of non-empty subsets of N, define $\mathcal{S}(\overline{X})$, the set of unordered sections of $\overline{X}$, as*

$$\mathcal{S}(\overline{X}) \qquad \stackrel{\text{def}}{=} \qquad \left(\prod_{1\leq i\leq n} X_i \right) / S_n \ , \tag{1}$$

where $\_/S_n$ denotes quotienting by the action of the symmetric group S_n.

From now on, we will drop the adjective "unordered" and refer to an element of $\mathcal{S}(\overline{X})$ simply as a section of $\overline{X}$. We use this notion to define a new preorder on $\mathcal{P}_*(N)^n$:

Definition 2. *If $\overline{X}$ and $\overline{Y}$ are in $\mathcal{P}_*(N)^n$, put:*

$$\overline{X} \sqsubseteq \overline{Y} \qquad \stackrel{\text{def}}{=} \qquad \mathcal{S}(\overline{X}) \subseteq \mathcal{S}(\overline{Y}) \ . \tag{2}$$

We write $\overline{X} \approx \overline{Y}$ for "$\overline{X} \sqsubseteq \overline{Y}$ and $\overline{Y} \sqsubseteq \overline{X}$".

This is not antisymmetric as $\left(X_{\sigma(1)}, \dots, X_{\sigma(n)}\right) \approx (X_1, \dots, X_n)$ for any permutation $\sigma \in S_n$.

The aim of this note is to answer the following:

Questions.

- *When do we have $\overline{X} \approx \overline{Y}$?*
- *What is the relation between $\overline{X} \sqsubseteq \overline{Y}$ and $\overline{X} \subseteq \overline{Y}$?*

The problem is subtler than it appears and the first question makes for an interesting puzzle: while elementary, the proof is more complex than what most people initially think. Readers are thus encouraged to spend a couple of minutes playing with the problem before reading on.

* Laboratoire de Mathématiques, Université de Savoie, 73376 Le Bourget-du-Lac Cedex, France

Notation: to make formulas less verbose, we will abuse the vector notation by lifting "$\in$" pointwise: just as $\overline{X} \subseteq \overline{Y}$ means "$\forall 1 \leq i \leq n,\ X_i \subseteq Y_i$", the notation $\overline{a} \in \overline{X}$ is a synonym for "$\forall 1 \leq i \leq n,\ a_i \in X_i$". The action of S_n on n-tuples is written with a dot and is defined as $\sigma{\cdot}\overline{a} \stackrel{\text{def}}{=} \big(a_{\sigma(1)}, \ldots, a_{\sigma(n)}\big)$. When talking about n-multisets (orbits for the action of S_n), we identify an n-tuple with its orbit. In particular, $\overline{a} \in \mathcal{S}\big(\overline{X}\big)$ means that $\sigma{\cdot}\overline{a} \in \overline{X}$ for some permutation σ.

1. The equivalence relation. The first question has a simple answer: equivalence is just equality up-to a permutation of the sets.

Proposition 1. *Given any $\overline{X}$ and $\overline{Y}$ in $\mathcal{P}_*(N)^n$, we have*

$$\overline{X} \approx \overline{Y} \quad \Leftrightarrow \quad \exists \sigma \in S_n,\ \sigma{\cdot}\overline{X} = \overline{Y}\ . \tag{3}$$

In other words, the equivalence class of $\overline{X}$ consists of all the $\sigma{\cdot}\overline{X}$.

This is slightly surprising because the left-hand side is definitionally equal to

$$\forall \overline{a} \in \overline{X},\ \exists \sigma \in S_n,\ \sigma{\cdot}\overline{a} \in \overline{Y} \quad \text{and} \quad \forall \overline{a} \in \overline{Y},\ \exists \sigma \in S_n,\ \sigma{\cdot}\overline{a} \in \overline{X}\ ,$$

while the right-hand side is definitionally equal to

$$\exists \sigma \in S_n, \quad \Big(\forall \overline{a} \in \overline{X},\ \sigma{\cdot}\overline{a} \in \overline{Y} \text{ and } \forall \overline{a} \in \overline{Y},\ \sigma^{-1}{\cdot}\overline{a} \in \overline{X}\Big)\ .$$

That the latter implies the former is trivial. Proposition 1 asserts the converse, i.e., that we can choose the permutation uniformly for all the $\overline{a} \in \overline{X}$ and $\overline{a} \in \overline{Y}$!

Lemma 1.

- *If $\overline{X} \sqsubseteq \overline{Y}$ then, for all $1 \leq j \leq n$, there is some $1 \leq i \leq n$ s.t. $X_i \subseteq Y_j$.*
- *If $\overline{X} \approx \overline{Y}$ then $X_{i_0} = Y_{j_0}$ for some pair i_0, j_0.*

Proof. The first point is done by contradiction. Suppose that $\exists j_0, \forall i, X_i \not\subseteq Y_{j_0}$, this implies that there is an $\overline{a} \in \overline{X}$ s.t. $\forall i, a_i \notin Y_{j_0}$. This $\overline{a}$ cannot be a section of $\overline{Y}$. Contradiction!

The second point follows easily: starting from Y_1 and repeatedly using the first point, we can construct an infinite chain of reverse inclusions:

$$Y_1 \ \supseteq\ X_{i_1} \ \supseteq\ Y_{j_2} \ \supseteq\ X_{i_2} \ \supseteq\ Y_{j_2} \ \supseteq\ \ldots$$

Because there are finitely many X_i's and Y_j's, the chain must contain cycles. This gives an equality between some X_i and Y_j. □

This solves the beginning of Proposition 1: if $\overline{X} \approx \overline{Y}$, then at least one of the sets appears on both sides. What we now need is to show that

$$(Z, X_2, \ldots, X_n) \approx (Z, Y_2, \ldots, Y_n) \quad \Rightarrow \quad (X_2, \ldots, X_n) \approx (Y_2, \ldots, Y_n)\ .$$

This will allow to finish the proof of Proposition 1 by induction on n. If unordered sections are seen as a "commutative cartesian product", the next definition would be the corresponding "division":

Definition 3. *Let T be a collection of n-multisets and $Z \in \mathcal{P}_*(N)$ we put*

$$T \div Z \quad \stackrel{\text{def}}{=} \quad \Big\{(a_2, \ldots, a_n) \;\Big|\; \forall a \in Z,\ (a, a_2, \ldots, a_n) \in T\Big\}\ .$$

Lemma 2. *We have:*

$$\mathcal{S}\big(X_1, X_2, \ldots, X_n\big) \div X_1 \quad = \quad \mathcal{S}\big(X_2, \ldots, X_n\big)\ . \tag{4}$$

Proof. The "$\supseteq$" inclusion follows from the definition.

For the "$\subseteq$" inclusion, suppose $(a_2, \dots, a_n) \in \mathcal{S}\big(\overline{X}\big) \div X_1$. Let $b \in X_1$: we necessarily have that $(b, a_2, \dots, a_n) \in \mathcal{S}\big(\overline{X}\big)$, i.e., there is a permutation τ s.t. $(b, a_2, \dots, a_n) \in (X_{\tau(1)}, \dots, X_{\tau(n)})$.

If $\tau(1) = 1$, we have $(a_2, \dots, a_n) \in (X_{\tau(2)}, \dots, X_{\tau(n)})$ and we can conclude directly.

If $\tau(1) \neq 1$, up-to choosing an appropriate element in the orbit of $(a_2, \dots a_n)$, we can assume that $\tau(1) = 2$, $\tau(2) = 1$ and $\tau(i) = i$ when $2 < i \leq n$, i.e., that $b \in X_2$, $a_2 \in X_1$, and $a_i \in X_i$ whenever $2 < i \leq n$.

Put $a_1 \stackrel{\text{def}}{=} a_2$. Because $a_1 = a_2 \in X_1$, we have $(a_1, a_2, \dots, a_n) \in \mathcal{S}\big(\overline{X}\big)$, i.e., $\sigma{\cdot}\overline{a} \in \overline{X}$ for some permutation σ. Let $k \stackrel{\text{def}}{=} \min\{i \mid i > 0, a_{\sigma^i(1)} = a_1\}$. This k exists and is at most the length of the cycle of σ containing 1. We put $I \stackrel{\text{def}}{=} \big\{1, \sigma(1), \dots, \sigma^{k-1}(1)\big\}$ and $I^c = \{1, \dots, n\} \setminus I$.

Rearrange the columns of

$$\begin{array}{ccccccccccc} a_{\sigma(1)} & & \cdots & & a_1 = a_2 & & \cdots & & a_2 & & a_{\sigma(n)} \\ \bullet & & & & \bullet & & & & \bullet & & \bullet \\ \big|{\in} & & & & \big|{\in} & & & & \big|{\in} & & \big|{\in} \\ \bullet & & & & \bullet & & & & \bullet & & \bullet \\ X_1 & & \cdots & & X_? & & \cdots & & X_? & & X_n \end{array}$$

into two parts:

$$\underbrace{\begin{array}{ccccc} a_{\sigma(1)} & a_{\sigma^2(1)} & \cdots & a_{\sigma^k(1)} = a_1 & \\ \bullet & \bullet & & \bullet & \\ \bullet & \bullet & & \bullet & \\ X_1 & X_{\sigma(1)} & \cdots & X_{\sigma^{k-1}(1)} & \end{array}}_{I} \quad \underbrace{\begin{array}{ccccccc} \cdots & & a_2 & \cdots & a_{\sigma(i)} & & \cdots \\ & & \bullet & & \bullet & & \\ & & \big|{\in} & & \big|{\in} & & \\ & & \bullet & & \bullet & & \\ \cdots & & X_? & \cdots & X_i & & \cdots \end{array}}_{I^c} .$$

The indices of $\overline{X}$ on the left are exactly those in I, and so are the indices of $\overline{a}$. Thus, the indices of $\overline{X}$ and $\overline{a}$ on the right are exactly those in I^c. This shows that $(a_i)_{i \in I^c}$ is a section of $(X_i)_{i \in I^c}$. Moreover, because each of $\sigma(1)$, …, $\sigma^{k-1}(1)$ is strictly more than 2 (by the definition of k), we also have $a_{\sigma^i(1)} \in X_{\sigma^i(1)}$ for each $1 \leq i \leq k-1$ by a previous hypothesis. This shows that the permutation

$$\rho : \{2, \dots, n\} \to \{2, \dots, n\} \qquad \rho(i) \stackrel{\text{def}}{=} \begin{cases} i & \text{if } i \text{ is one of } \sigma(1), \dots, \sigma^{k-1}(1) \\ \sigma(i) & \text{otherwise} \end{cases}$$

satisfies $\rho{\cdot}(a_2, \dots, a_n) \in (X_2, \dots, X_n)$, and this finishes the proof that $(a_2, \dots, a_n)$ is indeed a section of $(X_2, \dots, X_n)$. □

2. The preorder. The initial question wasn't very formal and read as: "*What is the relation between $\overline{X} \sqsubseteq \overline{Y}$ and $\overline{X} \subseteq \overline{Y}$?*" We already know that $\sigma{\cdot}\overline{X} \subseteq \overline{Y}$ implies $\overline{X} \sqsubseteq \overline{Y}$. A corollary to Proposition 1 gives a slightly more precise result:

$$\sigma{\cdot}\overline{X} \subseteq \overline{Y} \text{ and } \overline{X} \not\approx \overline{Y} \quad \Rightarrow \quad \overline{X} \sqsubsetneq \overline{Y} .$$

We cannot hope for the converse: $\big(\{3\}, \{1,2,3\}\big) \sqsubseteq \big(\{2,3\}, \{1,3\}\big)$ but $\{1,2,3\} \not\subseteq \{2,3\}$ and $\{1,2,3\} \not\subseteq \{1,3\}$! What happens is that some of the elements of the Y_i's (like "2" above) can move around in a restricted manner without adding new sections. We will characterize exactly what constraints are involved. First, a definition:

Definition 4.

- *Let $\mathbf{B} \stackrel{\text{def}}{=} \{0,1\}$ equipped with the order $0 \leq 1$. This is a complete lattice with operations written $\vee$ and $\wedge$.*
- *The lattice structure is lifted pointwise to $\mathbf{B}^n$.*
- *If $u \in \mathbf{B}^n$, the weight of u is the number of 1's in u. It is written $|u|$.*

We write elements of $\mathbf{B}^n$ without parenthesis as in "$011101 \in \mathbf{B}^6$".

Definition 5. *If $a \in N$ and $\overline{X} \in \mathcal{P}_*(N)^n$, $\chi_{\overline{X}}(a) \in \mathbf{B}^n$ is defined by $\big(\chi_{\overline{X}}(a)\big)_i \stackrel{\text{def}}{=} 1$ iff $a \in X_i$.*

Thus, $\chi_{\overline{X}}(a)$ describes in which X_i's the element a appears. The next lemma is a first step to answer our question: it gives a necessary and sufficient condition for $\overline{X} \sqsubseteq \overline{Y}$ by looking where each $a \in N$ appears in $\overline{X}$ and $\overline{Y}$. Note that is doesn't quantify over permutations.

Lemma 3. *We have $\overline{X} \sqsubseteq \overline{Y}$ iff $f(u) \stackrel{\text{def}}{=} \bigvee_{\chi_{\overline{Y}}(a) \leq u} \chi_{\overline{X}}(a)$ satisfies $|f(u)| \leq |u|$ for all u's.*

For example, for $\big(\{3\},\{1,2,3\}\big) \sqsubseteq \big(\{2,3\},\{1,3\}\big)$, the function f is $(b_1,b_2) \mapsto (b_1 \wedge b_2, b_1 \vee b_2)$:

$$\begin{array}{lllll} a: & 1 & 2 & 3 & \\ \chi_{\overline{Y}}(a): & 01 & 10 & 11 & \\ \chi_{\overline{X}}(a): & 01 & 01 & 11 & . \end{array}$$

The function f is the least (for the extensional order) increasing boolean function satisfying $f\big(\chi_{\overline{Y}(a)}\big) \geq \chi_{\overline{X}}(a)$. When, like in the example above, the function $a \mapsto \chi_{\overline{Y}}(a)$ is bijective from N to $\mathbf{B}^n \setminus \{0\cdots0\}$ and the function $\chi_{\overline{Y}}(a) \mapsto \chi_{\overline{X}}(a)$ is increasing (i.e., "when a appears in more Y_i's than b, then a appears in more X_i's than b"), the "global" condition of the lemma simplifies to the "local" condition that $\big|\chi_{\overline{X}}(a)\big| \leq \big|\chi_{\overline{Y}}(a)\big|$ for each a. Because this is a simple necessary condition for $\overline{X} \sqsubseteq \overline{Y}$, the interesting part of the lemma is that it is also sufficient.

Proof. For the "$\Leftarrow$" implication, suppose that $\overline{a} \in \overline{X}$. We want to show that $\sigma{\cdot}\overline{a} \in \overline{Y}$ for some permutation σ. If we look at the bipartite graph

$$\begin{array}{ccccc} a_1 & a_2 & & \cdots & a_n \\ \bullet & \bullet & & & \bullet \\ & & & & \\ \bullet & \bullet & & & \bullet \\ Y_1 & Y_2 & & \cdots & Y_n \end{array} \quad ,$$

with an edge between a_i and Y_j when $a_i \in Y_j$, finding a σ s.t. $\sigma{\cdot}\overline{a} \in \overline{Y}$ is equivalent to finding a perfect matching in the bipartite graph. By Hall's "marriage theorem" ([3]), this is equivalent to "every subset of the a_i's of cardinality c has at least c neighbors among the Y_j's".

Take some subset $U \subseteq \{a_1,\ldots,a_n\}$ of cardinality c. Because $\overline{a}$ is a section of $\overline{X}$, by the marriage theorem, this set has at least c neighbors in the corresponding $\overline{a}/\overline{X}$ bipartite graph. By the construction, we have

$$c \quad \leq \quad \underbrace{\left|\bigvee_{a\in U} \chi_{\overline{X}}(a)\right|}_{\substack{\#\text{ of neighboors}\\ \text{of } U \text{ in } \overline{a}/\overline{X}.}} \quad \leq \quad \left| f\left(\bigvee_{a\in U} \chi_{\overline{Y}}(a)\right)\right| \quad \leq \quad \underbrace{\left|\bigvee_{a\in U} \chi_{\overline{Y}}(a)\right|}_{\substack{\#\text{ of neighboors}\\ \text{of } U \text{ in } \overline{a}/\overline{Y}.}}$$

where the second inequality follows from the definition of f, and the third inequality follows from the hypothesis. This concludes the "$\Leftarrow$" implication.

For the "$\Rightarrow$" implication, let $\overline{X} \sqsubseteq \overline{Y}$ and $c = |u| < |f(u)|$. We can find $\{a_1,\ldots,a_k\} \subseteq N$ which satisfies $\bigvee_{i\leq k} \chi_{\overline{X}}(a_i) \leq u$ and $\big|\bigvee_{i\leq k} \chi_{\overline{X}}(a_i)\big| > c$. In particular, we have

$$\left|\bigvee_{1\leq i\leq k} \chi_{\overline{Y}}(a_i)\right| \quad \leq \quad c \quad < \quad \left|\bigvee_{1\leq i\leq k} \chi_{\overline{X}}(a_i)\right| .$$

For each 1 in $\bigvee_{i\leq k} \chi_{\overline{X}}(a_i)$ take one of the a_i's which has a 1 in the same position. Call the resulting tuple $\overline{a}$. Note that this tuple has length strictly greater than c and may contain repetitions. This is a partial section of $\overline{X}$: take those X_j's s.t. $\bigvee_{i\leq k} \chi_{\overline{X}}(a_i)$ contains a 1 on the j-th coordinate. To complete this tuple into a section of the whole $\overline{X}$, add one element from each of the remaining sets to obtain a section $(\overline{a},\overline{a}')$ of $\overline{X}$. This is also a section of $\overline{Y}$ and in particular, each element of $\overline{a}$ needs to find its place into one of the Y_i's. This is impossible because there are at most c sets Y_i that can contain the elements of the tuple $\overline{a}$. Contradiction!

□

Lemma 3 does characterize the $\sqsubseteq$ preorder but still looks a little ad-hoc. To go further, we will use the concept of *strictly increasing boolean function*.

3. Interlude: strictly increasing boolean functions. The next notion is quite natural, but the literature is scarce on the subject. For example, neither the big book [7] on boolean functions nor the survey [2] on monotone boolean functions seems to mention them.

Ultra Simple Lemma. *If $f : \mathbf{B}^n \to \mathbf{B}^m$ is strictly increasing, then $n \leq m$.*

Proof. Just note that chains in $\mathbf{B}^n$ have length at most n and that a strictly increasing function preserves the cardinality of chains. □

This remark might be the reason why the notion doesn't appear much in the literature on boolean functions: it is not possible to decompose a strictly increasing function $\mathbf{B}^n \to \mathbf{B}^m$ into an m-tuple of strictly increasing functions $\mathbf{B}^n \to \mathbf{B}$. In other words, the outputs of a strictly increasing boolean function cannot be treated independently.

Simple Lemma. *An increasing function $f : \mathbf{B}^n \to \mathbf{B}^n$ is strictly increasing iff it is weight preserving: $|f(u)| = |u|$ for all $u \in \mathbf{B}^n$.*

In particular, the values of $f(0\cdots0)$ and $f(1\cdots1)$ are fixed to $0\cdots0$ and $1\cdots1$.

Examples of such functions are the permutations $u \mapsto \sigma{\cdot}u$: they are exactly the invertible strictly increasing boolean functions. Less trivial and more interesting is the "and/or" function: $(b_1, b_2) \mapsto (b_1 \wedge b_2, b_1 \vee b_2)$ whose full graph is

$$\left\{\begin{array}{lcl} 00 & \mapsto & 00 \\ 10 & \mapsto & 01 \\ 01 & \mapsto & 01 \\ 11 & \mapsto & 11 \end{array}\right. .$$

Applying this "and/or" function repeatedly yields many more examples, including the function pushing all the 1's to the right:

$$u \quad \mapsto \quad 00\cdots00\ \underbrace{11\ldots11}_{|u|} .$$

Unfortunately, strictly increasing boolean functions aren't generated by those two families. For example, it can be shown that the next function cannot be obtained by composing permutations with "and/or" functions:

$$\left\{\begin{array}{lcl} 0000 & \mapsto & 0000 \\ 0001, 0010, 0100, 1000 & \mapsto & 0001 \\ 1100, 0110, 0011 & \mapsto & 0011 \\ 1010, 0101, 1001 & \mapsto & 0101 \\ 1110, 1101, 1011, 0111 & \mapsto & 0111 \\ 1111 & \mapsto & 1111 \end{array}\right. .$$

See the appendix for a proof that strictly increasing boolean functions aren't even finitely generated.

Because of global constraints, not every partial weight preserving function can be promoted to a total one. For example,

$$\left\{\begin{array}{lcl} 1100 & \mapsto & 1100 \\ 1010 & \mapsto & 0011 \end{array}\right.$$

cannot appear in the graph of a strictly increasing: it would imply $1110 \mapsto 1111$, which contradicts the "simple lemma". In general, to check if a partial weight-preserving function f can be extended to a total one, compute $f' : u \mapsto \bigvee_{v \leq u} f(v)$, the least increasing total function agreeing with f. For f to be part of a total strictly increasing function, it is necessary (by the "simple lemma") to have $|f'(u)| \leq |u|$ for all $u \in \mathbf{B}^n$. The next lemma shows that this is also sufficient.

Lemma 4. *An increasing function $f : \mathbf{B}^n \to \mathbf{B}^n$ is extensionally smaller than a strictly increasing function if and only if $|f(u)| \leq |u|$ for all $u \in \mathbf{B}^n$.*

Proof. The implication "⇒" follows directly from the "simple lemma". For the converse, we define a new function f' from f in such a way that:

- $f \leq f'$,
- f' is increasing and weight preserving.

We define $f'(u)$ inductively by looking at u's with decreasing weights. Whenever $|f(u)| = |u|$, we put $f'(u) \stackrel{\text{def}}{=} f(u)$. Otherwise, we look at $\bigwedge_{u<v} f'(v)$. Note that because we start with the "heavy" u's, this is well defined. There are two possibilities:

- $\left|\bigwedge_{u<v} f'(v)\right| = |u|$, in this case we use $\bigwedge_{u<v} f'(v)$ for the value of $f'(u)$,
- $\left|\bigwedge_{u<v} f'(v)\right| = 1 + |u|$, in that case we choose a "1" in $\bigwedge_{u<v} f'(v)$ that is not in $f(u)$ and replace it with a "0".

Provided no other case appears, this defines a weight-preserving increasing function greater than f.

Claim. *During the construction, we always have*

$$|u| \quad \leq \quad \left|\bigwedge_{u<v} f'(v)\right| \quad \leq \quad 1 + |u| \ .$$

Note that the infimum can be obtained by looking at the successors* of u, whose weight is always $1 + |u|$. Since f' is weight preserving, we have $\left|\bigwedge_{u<v} f'(v)\right| \leq 1 + |u|$. Suppose by contradiction that $\left|\bigwedge_{u<v} f'(v)\right| < |u|$. By enumerating the successors $v_1, \ldots, v_k$ of u and expanding the infimum as

$$\bigvee_{v<u} f'(v) \quad = \quad \overbrace{\underbrace{f'(v_1) \ \vee \ \cdots \ \vee \ f'(v_i)}_{|\cdots| < |u|} \ \vee \ \cdots \ \vee \ f'(v_j)}^{|\cdots| \leq |u|} \ \vee \ \cdots$$

with minimal i and j, we can find some w_1 and w_2 s.t.

- w_1 and w_2 are successors of v,
- $|f'(w_1) \wedge f'(w_2)| < |u|$.

It is easy to see that either "$w_1 \stackrel{\text{def}}{=} v_1$ and $w_2 \stackrel{\text{def}}{=} v_j$" or "$w_1 \stackrel{\text{def}}{=} v_i$ and $w_2 \stackrel{\text{def}}{=} v_j$" will work. We obtain a contradiction because

$$\begin{array}{rcl} 2 + |u| & = & \left|f'(w_1 \vee w_2)\right| \\ & \geq & \left|f'(w_1) \vee f'(w_2)\right| \\ & = & \left|f'(w_1)\right| + \left|f'(w_2)\right| - \left|f'(w_1) \wedge f'(w_2)\right| \\ & > & 1 + |u| + 1 + |u| - |u| \\ & = & 2 + |u| \quad . \end{array}$$

This finishes the proof. □

4. Back to the preorder. We can now give a more elegant answer to the original question. Note that we can lift any function $f : \mathbf{B}^n \to \mathbf{B}^m$ to a function $\mathcal{P}_*(N)^n \to \mathcal{P}_*(N)^m$:

Definition 6. *Suppose $f : \mathbf{B}^n \to \mathbf{B}^m$, define $\widehat{f} : \mathcal{P}_*(N)^n \to \mathcal{P}_*(N)^m$ as*

$$\widehat{f}(\overline{Y}) \quad \stackrel{\text{def}}{=} \quad \overline{X} \qquad \textit{with } a \in X_i \textit{ iff } f\big(\chi_{\overline{Y}}(a)\big) \textit{ contains a 1 at coordinate } i \ .$$

This transformation is, in a precise categorical sense, *natural*. It amounts to lifting the boolean operations $\wedge$ and $\vee$ to their set theoretic versions $\cap$ and $\cup$ in a way that is compatible with function composition. For example, with the "and/or" function $(b_1, b_2) \mapsto (b_1 \wedge b_2, b_1 \vee b_2)$ we obtain "intersection/union" $(Y_1, Y_2) \mapsto (Y_1 \cap Y_2, Y_1 \cup Y_2)$. Putting Lemma 3 and Lemma 4 together, we can now answer the initial question:

* *successors* of u are those v's with $v > u$ and $|v| = |u| + 1$.

Proposition 2. *For any $\overline{X}$ and $\overline{Y}$, we have*

$$\overline{X} \sqsubseteq \overline{Y} \qquad \Leftrightarrow \qquad \overline{X} \subseteq \widehat{g}(\overline{Y}) \quad \textit{for some strictly increasing } g : \mathbf{B}^n \to \mathbf{B}^n.$$

Proof. We know that $\overline{X} \sqsubseteq \overline{Y}$ is equivalent to having $|f(u)| \leq |u|$ for all u in $\mathbf{B}^n$, where f is defined as in Lemma 3. By Lemma 4, this is equivalent to saying that there is a strictly increasing g s.t. $f \leq g$. The two following points are thus equivalent:

- $\overline{X} \sqsubseteq \overline{Y}$,
- there is a strictly increasing $g : \mathbf{B}^n \to \mathbf{B}^n$ s.t. $\chi_{\overline{Y}}(a) \leq u$ implies $\chi_{\overline{X}}(a) \leq g(u)$.

The second point implies that $\overline{X} \subseteq \widehat{g}(\overline{Y})$: use $u \stackrel{\text{def}}{=} \chi_{\overline{Y}}(a)$ to check that $a \in X_i$ is an element of the i-th set of $\widehat{g}(\overline{Y})$. For the converse, suppose g is strictly increasing s.t. $\overline{X} \subseteq \widehat{g}(\overline{Y})$ and let $\chi_{\overline{Y}}(a) \leq u$. Suppose that $\chi_{\overline{X}}(a)$ contains a 1 in position i, this means that $a \in X_i$ and thus a is in the i-th set of $\widehat{g}(\overline{Y})$, i.e., $g\big(\chi_{\overline{Y}}(a)\big)$ contains a 1 in position i. This implies that $g(u)$ also contains a 1 in position i. □

5. An algorithm. Proposition 2 is more elegant but Lemma 3 has an interesting byproduct: it gives a concrete algorithm to check if $\overline{X} \sqsubseteq \overline{Y}$. For that, construct the function f from Lemma 3 and check that it satisfies the condition. Just as a proof of concept, here is the main part of the algorithm, in the Python programming language. Minor alterations have been made to make it more readable.* The most difficult (and fun) part was to write the function "`combinations`" that generates all the vectors of length `n` and weight `w` using one of the subtle algorithms from [5]! (Go read it. It is quite fascinating...)

```
def check(N,n,X,Y):          # N is a set, n is an integer, X / Y are tuples of sets
    def combinations(w):
        # generates all vectors of weight w
        # omitted (see Knuth, or use you favorite method)
    def sup(u,v):            # complexity: O(n)
        # computes the pointwise "or" on n-tuples
        # omitted (simple)
    def weight(u):           # complexity: O(n)
        # computes the weight of an n-tuple
        # omitted (simple)
    def chi(a,Z):            # complexity: O(n log(z))   (z is cardinality of Z)
        for i in range(n):   # we use Python builtin "set" type
            if a in Z[i]:    # so that "a in Z[i]" mean "a belongs to Z[i]"
                u[i] = 1
        return u

    F = {}                   # F is a finite map with at most 2^n elements,
                             # access is logarithmic: O(log(2^n)) = O(n)
    for a in N:                          # complexity: c *
      chiX = chi(a,X)                    #                   n log(x)
      chiY = chi(a,Y)                    #                 + n log(y)
      F[chiY] = sup(F[chiY] , chiX)      #                 + 2n
    for w in range(n+1):                 # generating all tuples
        for u in combinations(w):        # complexity : about 2^n *
            v = F[u]                                    #           n
            for i in range(n):                          #         + n *
                if u[i] == 1:                           #
                    u[i] = 0                            #
                    v = sup(v,F[u])                     #               n^2
                    u[i] = 1                            #
            F[u] = v                                    #
            if weight(v) > w:                           #         + n
                return False
    return True     # if we reached this far, the condition is satisfied
```

If N has cardinality c and the sets X_i's and Y_i's have cardinalities at most x and y; and if we suppose that the standard operations on sets and finite functions have logarithmic complexity,

* The full version is available on the author's web-page.

the hints in the comments give a total complexity of roughly $O\big(cn\big(\log(x)+\log(y)\big)+2^n n^3\big)$. Because $x = O(c)$ and $y = O(c)$, we get a complexity of $O\big(nc\log(c)+2^n n^3\big)$. If c is fixed, this is $O(n^3 2^n)$; if n is fixed, this is $O\big(c\log(c)\big)$. In almost all cases, this is better (and much easier to write) than the naive approach that checks if each $\overline{a} \in \overline{X}$ is a section of $\overline{Y}$, even if we are allowed to use an oracle to guess the permutations.

∞. Further questions. The origin of the initial question can be found in a rather different area: denotional models of linear logic. In [4], the relation $\mathcal{S}\big(\overline{X}\big) \subseteq \mathcal{U}$ played an important role in the interpretation of the exponential modality of linear logic. The set $\mathcal{U}$ was an arbitrary collection of n-multisets but thinking about the preorder "$\sqsubseteq$" was a natural step from there.

There are more questions one can ask about this preorder, or more generally about strictly increasing boolean functions. Readers will probably come up with more examples but the most immediate ones are:

- Strictly increasing boolean functions aren't finitely generated, but how many cells are needed to generate those functions that are in $\mathbf{B}^n \to \mathbf{B}^n$? (Asymptotic bounds etc.) Can we give canonical minimal generating families? If so, what can be said about the complexity of strictly increasing boolean functions expressed with those cells?
- How many strictly increasing boolean functions from $\mathbf{B}^n$ to itself are there? The first four values of this sequence are "$1, 4, 66, 7128$" or "$1, 2, 11, 297$" if one divides each number by $n!$ to take symmetries on the outputs into account. Neither of them appears in Sloane's encyclopedia [6]. This question is a variant of *Dedekind's problem* from [1] of counting the number of increasing boolean functions form $\mathbf{B}^n$ to $\mathbf{B}$. Besides some asymptotic bounds, only the first eight Dedekind numbers are known [8]. The global nature of *strict* monotonicity makes in unlikely that the answers to these two questions are related.

References[†]

[1] Richard Dedekind, "Ueber Zerlegungen von Zahlen durch ihre grössten gemeinsamen Teiler", *Gesammelte mathematische Werke*, vol. II, pp. 103-148, 1897.

[2] Aleksej Dmitrievich Korshunov , "Monotone Boolean Functions", *Russian Mathematical Survey, 58–929*, 2003.

[3] Philip Hall, "On Representatives of Subsets", *Journal of the London Mathematical Society 10 (1)*, 1935.

[4] Pierre Hyvernat, "Predicate Transformers and Linear Logic: yet another Denotational Model", *Lecture Notes in Computer Science*, vol. 3210, 2004, Springer.

[5] Donald E. Knuth, "Generating all Combinations and Partitions", volume 4 / fascicle 3 of "The Art of Computer Programming", Addison-Wesley, 2005.

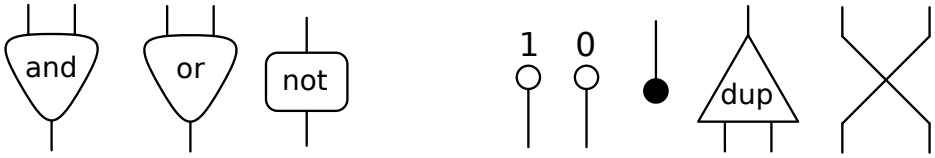
[6] Neil J. A. Sloane et al. "The On-Line Encyclopedia of Integer Sequences", web address: `http://oeis.org`

[7] Ingo Wegener, "The Complexity of Boolean Functions", Wiley-Teubner publishing, 1987.

[8] Doug Wiedemann, "A computation of the eighth Dedekind number", *Order 8 (1)*, 1991, Springer.

[†] None of the references in this bibliography is directly concerned with either the preorder "$\sqsubseteq$" or strictly increasing boolean functions. Any reference to papers or books mentionning them would be most welcome.

Appendix. Strictly increasing functions are not finitely generated. It is well known that all boolean functions $\mathbf{B}^n \to \mathbf{B}^m$ can be represented by a boolean circuit using only "and", "or" together with 'not" gates. Strictly speaking, we also need constant values and be able to forget / duplicate inputs:



where the last cells are:

- constants 0 and 1 (zero input, one output),
- drop (one input, zero output),
- duplicate (one input, two outputs),
- crossing (two inputs, two outputs).

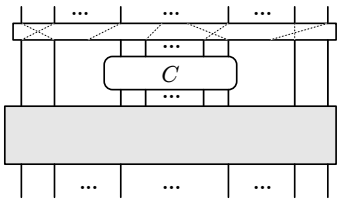
These cells, together with a finite set of relations expressing properties of the operations (associativity, etc.), give a finite presentation of the monoidal category of boolean functions.* We can generate the subcategory of increasing functions by removing the "not" cell from the generators. Unfortunately, no such thing is possible for *strictly increasing* boolean functions.

Proposition 3. *Strictly increasing boolean functions are not finitely generated.*

Proof. First note that because of the "ultra simple lemma", we cannot use cells with more inputs than outputs as they wouldn't be strictly increasing on their own. In particular, any strictly increasing boolean function from $\mathbf{B}^n$ to itself must be represented using cells with just as many inputs as outputs. An example of such cell could be $(b_1, b_2) \mapsto (b_1 \wedge b_2, b_1 \vee b_2)$, with two inputs and two outputs.

We will prove Proposition 3 by contradiction: suppose there is a finite set of cells that generates all strictly increasing boolean functions, and write A for the maximal arity of the cells in this set.

Any function has a representation



where

- the topmost rectangle contains only crossing (or invertible cells),
- the cell C is not invertible,
- and the lowermost rectangle contains the rest of the representation.

The cell C has arity less than A and is not a bijection. It implies that it has at least two input wires a and b s.t. C gives the same value on the two inputs consisting of 0's and a single 1 in position a or b. Because this is independant of the remaining wires, the whole function must give the same result when:

- a takes value 1, all other inputs to C (including b) take value 0 and the rest takes value $\overline{v}$,
- b takes value 1, all other inputs to C (including a) take value 0 and the rest takes value $\overline{v}$.

This will be true for any tuple of values $\overline{v}$ for the remaining wires.

* All of this has a precise algebraic meaning...

We will find a rather large n n and construct a function $f : \mathbf{B}^n \to \mathbf{B}^n$ that contradicts this fact: whenever we choose input wires a and b and put $A-2$ other input wires to 0, we can complete the other input wires in such a way that putting $a \stackrel{\text{def}}{=} 0$ and $b \stackrel{\text{def}}{=} 1$, or putting $b \stackrel{\text{def}}{=} 0$ and $a \stackrel{\text{def}}{=} 1$ makes a difference in the output of the function. Thus, this function will not be representable using the given set of cells.

Given a large n, define $f : \mathbf{B}^n \to \mathbf{B}^n$ as:

$$f(u) \stackrel{\text{def}}{=} \begin{cases} 0^n & \text{if } |u| = 0 \\ 1\ 0^{n-1} & \text{if } |u| = 1 \\ 1^k\ 0^{n-k} & \text{if } |u| = k \text{ is even} \\ 1101\ 0^{n-4} & \text{if } u = 0\cdots0\ 110^l1\ 0\cdots0, \text{ with } l > 0 \\ 1110\ 0^{n-4} & \text{if } |u| = 3 \text{ but } u \text{ not of the previous shape} \\ 1^{2^k}01\ 0^{n-2^k-2} & \text{if } u = 0\cdots0\ 1^{2^k}0^{2^k}1\ 0\cdots0, \text{ with } k > 1 \\ 1^{2^k}01\ 0^{n-2^k-2} & \text{if } u = 0\cdots0\ 10^{2^k}1^{2^k}\ 0\cdots0, \text{ with } k > 1 \\ 1^{2k}10\ 0^{n-2k-2} & \text{if } |u| = 2k+1 > 3 \text{ but } u \text{ not of the previous shapes.} \end{cases}$$

This function is strictly increasing because whenever v is a successor of u, we have $f(v) > f(u)$:

- $f(u) = 1^{2k}\ 0\cdots$ when $|u| = 2k$
- $f(u) = 1^{2k}10\ 0\cdots$ or $f(u) = 1^{2k}01\ 0\cdots$ when $|u| = 2k+1$.

Suppose input wires $i_1, \ldots, i_{A-2}$ are fixed to 0 and we want to differentiate between input wires a and b, with $a < b$. By putting some 1's in the remaining wires, we can make f give different results when "$a \stackrel{\text{def}}{=} 0$, $b \stackrel{\text{def}}{=} 1$" and "$a \stackrel{\text{def}}{=} 1$, $b \stackrel{\text{def}}{=} 0$".

- If there are two consecutive wires between a and b (but not touching b) which are not among $i_1, \ldots, i_{A-2}$, we put those two wires to 1 and all the other wires to 0. By lines 4 and 5 from the definition of f, we will get two different results.
- If not, the wires a and b cannot be too far apart. (There are at most $2A-2$ wires between them...) If we can find a sequence of 2^k consecutive wires at distance 2^k to the left of a, or a sequence of 2^k consecutive wires at distance 2^k to the right of b, we can put those wires to 1 and the rest to 0. By lines 6 and 8 or 7 and 8 of the definition of f, we will also get different results.

For this to work, we have to make sure n is big enough. At worst, the wires $i_1, \ldots, i_{A-2}$ can prevent us from finding an appropriate sequence $A-2$ times. In particular, if a is big enough (bigger than 2^{A+1}), such a sequence is bound to happen. The same is true when b is small enough compared to n. In the end, choosing n bigger than, say, 2^{2A+2} plus an additional ε will guarantee that we can differentiate any a and b among any set of A wires. A more careful analysis shows that it is in fact enough to take $n \stackrel{\text{def}}{=} 2^{A+1} + 4$. This finishes the proof. □