

O-notation in Algorithm Analysis (updated)

Kalle Rutanen

October 29, 2016

To my dear cats Kati (1996–2015) and Minari (2001–2016),
with whom I shared my life.

Abstract

In computer science, algorithm analysis is concerned with the correctness and complexity of algorithms. *Correctness analysis* is about verifying that an algorithm solves the problem it is claimed to solve, and *complexity analysis* is about counting the amount of resources an algorithm takes to run in an abstract machine. This paper is concerned with complexity analysis.

The amount of resources consumed by an algorithm is captured by a cost function which maps each input of the algorithm to a non-negative real number. The cost functions are ordered by an order-relation \preceq , which makes it possible to *simplify*, and to *compare* cost functions. A cost function can be simplified by replacing it with an equivalent, simpler cost function. The smaller the cost function is according to \preceq , the better the algorithm — at least for the measured resource.

The O-notation $O(f)$ is the set of functions g which satisfy $g \preceq f$. It contains the same simplification and comparison tools in a slightly different, but equivalent form.

How should the O-notation be defined? By the above, we may ask an equivalent question: how should the order relation \preceq be defined? We provide 8 intuitive properties for \preceq , and then show that there is exactly one definition of \preceq which satisfies these properties: linear dominance.

We show that Master theorems hold under linear dominance, define the O-mappings as a general tool for manipulating the O-notation, and abstract the existing definitions of the O-notation under local linear dominance.

Preface

The research presented in this paper was carried out at the Department of Mathematics at Tampere University of Technology during years 2013–2016. The paper was financially supported by the Finnish National Doctoral Programme in Mathematics and its Applications (2013–2014), and by the science fund of the city of Tampere (Summer 2015).

I am deeply thankful to my supervisors Sirkka-Liisa Eriksson, Karen Egiazarian, and Germán-Gómez Herrero for the support during these years, and the years before them.

Special thanks go to office assistant Riitta Lahti, financial administrative assistant Tiina Sävilahti, and laboratory engineer Kari Suomela, at the Department of Mathematics. When it came to practical matters, they made my life easy, all the while uplifting the atmosphere.

I want to thank Raphael Reitzig for providing me with extensive feedback on an Arxiv-version of the paper, related to the content and typographical concerns. Concerning typography, the idea to draw inside the O-symbols is his; this resulted in \mathcal{O} , \mathcal{O} , \mathcal{O} , \mathcal{O} , \mathcal{O} , \mathcal{O} , and \mathcal{O} , which are both suggestive and can be drawn inline without extending line-height.¹ I was led to prove the limit theorems after he asked me whether ratio-limits could still be used to reason about the O-notation. He was also the first external reader to provide me with positive feedback.

Jarkko Kari suggested to study the independence of the primitive properties, and pointed out how some of those results already followed from the study of candidate definitions.

David Wilding created the original Figure J.1, which I have adapted here with his permission. In the introduction, I have used *svg-cards* playing cards designed by David Bellot.

¹I will take the blame for the shapes.

Preface

I want to thank Heikki Huttunen and Pekka Ruusuvuori for introducing me to academic work as an undergraduate at the Department of Signal Processing. I am grateful to my friends and coworkers at the Department of Signal Processing, and at the Department of Mathematics for all those excellent detours from research. Namely, Tapio Manninen, Francesco Cricri, Pasi Hämäläinen, Henri Riihimäki, Petteri Laakkonen, Marko Järvenpää, Juho Lauri, Vesa Vuojo, Elina Viro, and Markku Åkerblom.

Lastly, I want to thank my family, friends, and Alina for their invaluable support.

Tampere, October 2016

Kalle Rutanen

Contents

Abstract	i
Preface	ii
Contents	iv
1: Introduction to the paper	1
1.1 Algorithms and their analysis	5
1.2 Simplified analysis by O-notation	6
1.3 Definitions of O-notation	7
1.4 Example analyses	9
1.5 Some history of O-notation	15
1.6 Objective and contributions	18
1.7 Relation to publications	22
1.8 Automated checking	23
1.9 Outline of the paper	23
2: Preliminaries	25
2.1 Algorithms	25
2.2 Computational model and cost-model	29
2.3 Primitive properties	30
2.4 Problem complexity	36
2.5 O-notation	37
2.6 Implicit conventions	38
2.7 Worst case, best case, average case	40
2.8 Misuses	43
2.9 Completeness	45

Contents

3: Desirable properties	48
3.1 Dominance properties	48
3.2 Non-triviality properties	52
3.3 Abstraction properties	53
3.4 Structural properties	55
4: Characterization of the \mathcal{O}-notation	65
4.1 Linear dominance is sufficient	65
4.2 Linear dominance is necessary	67
4.3 Minimal properties	70
5: Working with the \mathcal{O}-notation	73
5.1 Surprising aspects	73
5.2 Master theorems	75
5.3 \mathcal{O} -mappings	77
5.4 \mathcal{O} -equalities	78
6: Local linear dominance	82
6.1 Definition	82
6.2 Limit theorems	84
7: Conclusion	89
Bibliography	90
A: Notation	93
B: Howell's counterexample	101
C: Proofs of implied properties	104
D: Proofs of local linear dominance properties	121
E: Proofs of limit theorems	132
F: Proofs of Master theorems	139
F.1 Master theorem over powers	139
F.2 Master theorem over reals	145
F.3 Master theorem over integers	150

Contents

G: Comparison of definitions	159
G.1 Trivial linear dominance	159
G.2 Asymptotic linear dominance	161
G.3 Co-asymptotic linear dominance	164
G.4 Cofinite linear dominance	166
G.5 Linear dominance	169
G.6 Affine dominance	170
G.7 Containment relations	178
H: Proofs of minimality	181
H.1 Elementwise dominance	181
H.2 Multiple dominance	183
H.3 Non-transitive dominance	185
H.4 Power dominance	187
I: Partitioned sets	192
I.1 Generalized partition-inverse	193
I.2 Partition-injectivity and left partition-inverse	195
I.3 Partition-surjectivity and right partition-inverse	198
I.4 Partition-bijectivity and partition-inverse	200
I.5 Partition-embedding	201
J: Preordered sets	203
J.1 Order and partitions	204
J.2 Order-preserving functions and down-sets	205
J.3 Residuated functions	209
J.4 Residuated functions and supremum	213
J.5 Residuated functions and p-inverses	216
J.6 Residuated functions and order-embeddings	219

Chapter 1

Introduction to the paper

Suppose we want to sort playing cards into increasing order. The kind of cards does not matter, only that for any two cards A and B , either A comes before B , denoted by $A < B$, or B comes before A , denoted by $B < A$. One way to sort the deck is by the following *insertion sort*, which is visualized in Figure 1.1.

1. If there are no cards left in the deck, we are done.
2. Otherwise, we pick the top-most card A from the deck.
3. Starting from the right end of the cards on the table, we compare A to each card B on the table, until $A < B$ for the first time.
4. If there is no such card B , we place A on the table as the right-most card, and start again from step 1.
5. Otherwise, we place A on the table immediately left of B , and start again from step 1.

How many comparisons between cards do we have to perform to sort a given deck with n cards? In the best case, the deck is already sorted in increasing order, requiring us to perform only $n - 1$ comparisons. In the worst case, the deck is already sorted in decreasing order, requiring us to perform $1 + 2 + 3 + \dots + (n - 1) = (n - 1)n/2$ comparisons. Therefore, sorting a deck with n cards using insertion sort always requires something between $n - 1$ and $(n - 1)n/2$ comparisons.

Another way to sort the deck is by the following *merge sort*.

Chapter 1. Introduction to the paper

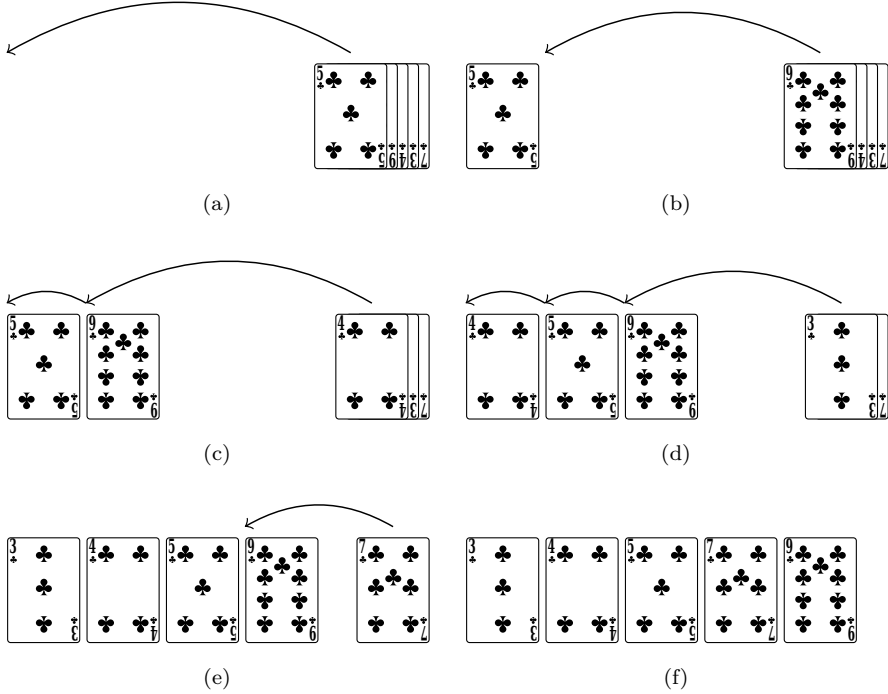
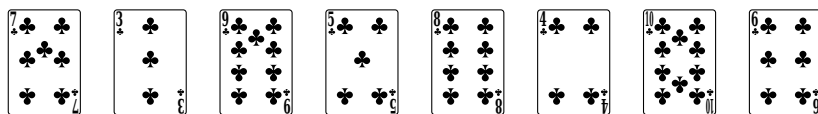


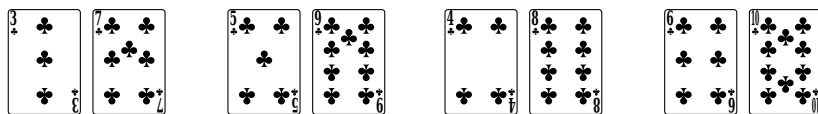
Figure 1.1: Sorting 5 playing cards with insertion sort. The cards on the table are visualized on the left, while the cards in the deck are visualized on the right. During sorting, 7 comparisons are made between cards.

1. We place all the cards in a row on the table. Every card A is considered to form a sequence (A) consisting of a single card.
2. If there is at most one card-sequence on the table, we are done.
3. For each two successive card-sequences, A and B , we combine A and B into a single sequence C by repeatedly picking the smallest of the cards at the left end of A and the left end of B (e.g., $(3, 7)$ and $(5, 9)$ becomes $(3, 5, 7, 9)$). If one of the sequences runs out before the

Chapter 1. Introduction to the paper



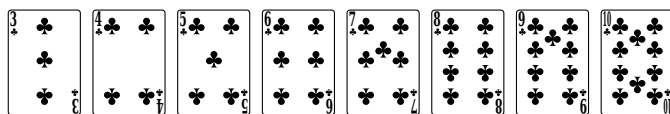
(a)



(b)



(c)



(d)

Figure 1.2: Sorting 8 playing cards with merge sort. During sorting, 17 comparisons are made between cards; the maximum possible for 8 cards. The merging procedure is repeated $\log_2(8) = 3$ times.

other, we place the remaining sequence at the right end of C without performing additional comparisons.

4. We start again from step 2.

How many comparisons between cards do we now have to perform to sort a given deck with n cards? Suppose n is a power of two, so that $n = 2^d$ for some integer d . In the best case, the cards on the table are already sorted, and we need to perform $(n/2) \log_2(n)$ comparisons to combine the sequences

Chapter 1. Introduction to the paper

Sorting procedure / n	64	256	1024	4096
Insertion sort — worst case	2016	32640	523776	8386560
Merge sort — worst case	321	1793	9217	45057
Insertion sort — best case	63	255	1023	4095
Merge sort — best case	192	1024	5120	24576

Table 1.1: Number of comparisons needed to sort n cards by using either insertion sort, or merge sort. Merge sort scales better than insertion sort in the worst case.

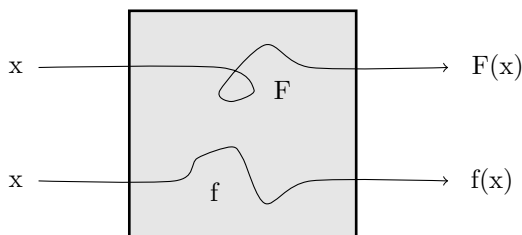


Figure 1.3: A black-box view of an algorithm. The algorithm, visualized here as a gray box, transforms an input x from an input-set X to an output $F(x)$ from an output-set Y . The transform is captured by a mapping $F: X \rightarrow Y$. The abstract machine executing the algorithm consumes resources at each step of the algorithm. The cost function, for given input and resource, is captured by a mapping $f: X \rightarrow \mathbb{R}^{\geq 0}$.

into a single deck. In the worst case, we need to perform $n(\log_2(n) - 1) + 1$ comparisons. A deck which attains this bound for $n = 8$ is given in Figure 1.2. Table 1.1 tabulates the number of comparisons required, in the worst case and in the best case, as a function of the number n of cards in the deck for both ways of sorting. It can be seen that merge sort scales better than insertion sort in the worst case. Note however, that there are decks — especially those which are sorted or almost sorted — for which insertion sort performs less comparisons than merge sort.

1.1 Algorithms and their analysis

Such step-wise procedures are called algorithms. A *run* of an algorithm is a finite sequence of *atomic operations* — in the above, picking a card from the deck, comparing two cards, and placing a card on the table. The atomic operations are performed one by one by an *abstract machine* — in the above the person sorting the cards. The abstract machine, running the algorithm, transforms a given *input* — such as a deck of cards — into *output* — such as the sorted input deck. This correspondence is captured by a function F , as visualized in Figure 1.3. The kind of algorithms that can be written depends on which atomic operations the abstract machine supports.

To each atomic operation, we associate a *cost* in resources — in the above one unit of resource to the task of comparing two cards. We then *analyze* the algorithm for the amount of resources that it uses. The result of the analysis is a *cost function* f which associates each possible input to the resource-cost. However, as for insertion sort and merge sort, the function f can be too hard to study directly — or to get an intuition to.

To make the analysis tractable, the input is grouped by some simpler property — in the above by the number of cards in the deck. The value of the property shared by a group is called the *label* of the group. Since a group may contain more than one input, we must come up with a way to summarize the costs inside each group.

Such summaries include *best case analysis* and *worst case analysis* — in the above providing us with the lower and upper bounds, respectively, for the number of comparisons required to sort a deck with a given number of cards. To analyze worst-case behavior under a given grouping, we pick from each group a *representative* input which triggers the worst resource-cost. Conceptually, we then analyze a *surrogate algorithm*, which takes a group-label as an input, maps it to the corresponding group-representative, and feeds the representative as an input to the actual algorithm.

Another common summary is *average case analysis*, where we assume a random distribution for the input, and then analyse the mean of the resulting random cost function.

Grouping is often required to make analysis tractable. However, too rough a grouping loses detail without aiding intuition. Sometimes grouping is not necessary at all — as when the input-set is \mathbb{Z} .

1.2 Simplified analysis by O-notation

Above we analyzed two algorithms for sorting playing cards, and noticed that merge sort is superior to insertion sort, at least when the measured resource is the number of performed comparisons in the worst case, and the deck contains at least 4 cards. This difference is fundamental, in the sense that the ratio by which merge sort beats insertion sort — in the worst case — keeps growing. For example, while for $n = 64$ the ratio is 6.3, for $n = 4096$ the ratio is already 186. Such *algorithmic efficiency* is often much more important than the speed at which a given abstract machine executes the atomic operations.

Our analyses above were based on careful counting of the comparisons, with an aim of capturing the worst case and best case bounds exactly. Such analysis is only possible with an algorithm that is sufficiently simple. Numerical analysis is a field whose algorithms (e.g., matrix QR-decomposition) have traditionally been analyzed in this way.

With a complex algorithm, a closed form expression for the cost function may not exist, or otherwise be too complex for a human to make sense of — not to mention the ingenuity needed to solve the combinatorial problems that occur when deriving those expressions.

The way out of these problems is to declare that we are not interested in small differences, and to simplify them out as they occur. For example, the relative error of approximating the worst-case resource consumption $(n - 1)n/2$ of insertion sort by $n^2/2$ is only $1/10$ for $n = 11$, and decreases to zero as n grows.

In addition, we declare that we are only interested in algorithmic efficiency, and so are ready to ignore the speed of the abstract machine up to a constant. It does not matter whether some other machine is 2 times faster to perform the algorithm. What matters is how the algorithm scales — does it scale like n^2 or like $n \log_2(n)$?

We then say that insertion sort takes about n^2 comparisons, because this is simple, and does not fundamentally differ from the exact count. However, here we must be careful. Since we aim to simplify expressions whenever possible, we need a guarantee that computing with the simplified expressions provides the same answer as first computing the exact cost function, and then simplifying it. We need *simplification rules* for algorithmic cost functions.

The simplification rules are formalized by a tool called *O-notation*. Intuitively, whenever we want to simplify, we surround the expression with an O, as in $O((n - 1)n/2)$, and then use whatever simplification rules the O-notation provides. With proper assumptions on the domain of n , we can then show that

Chapter 1. Introduction to the paper

$$\begin{aligned}
 O((n-1)n/2) &= O((n-1)n) \\
 &= O(n^2 - n) \\
 &= O(n^2).
 \end{aligned} \tag{1.1}$$

The notation $O(f)$, where f is a real-valued function, stands for the set of real-valued functions which scale at least as well as f . Having the same O -set divides the functions into equivalence classes — in the above, we then say that the functions $(n-1)n/2$ and n^2 are *equivalent*.

However, simplification is not the whole story. In addition, the O -notation must induce an order, so that the cost functions can be *compared*, and the order must respect the structure that is present in algorithms: looping, branching, and composition.

How do we define such an O -notation, and which rules do we need?

1.3 Definitions of O -notation

Here are some of the proposed definitions¹ for an O -notation in algorithm analysis.

Definition 1.2 (Asymptotic linear dominance). *Asymptotic linear dominance* Θ is defined by $g \in \Theta_X(f)$ if and only if

$$\exists c \in \mathbb{R}^{>0}, \exists y \in \mathbb{R}^d : (g|_{X^{\geq y}}) \leq c(f|_{X^{\geq y}}), \tag{1.3}$$

for all $f, g \in R_X$, and all $X \in U$, where $U = \bigcup_{d \in \mathbb{N}^{>0}} \mathcal{P}(\mathbb{R}^d)$.

Note 1.4. This definition is given in [1] in the univariate form on \mathbb{N} , and generalized to \mathbb{N}^2 in a later exercise [1, page 50]. ■

Definition 1.5 (Coasymptotic linear dominance). *Coasymptotic lin-*

¹ $\mathbb{R}^{>0}$ is the set of positive real numbers, $\mathcal{P}(X)$ is the set of subsets of X , $(f|_A)$ is the restriction of a function f to a set A , $f \leq g$ for functions f and g means $\forall x \in X : f(x) \leq g(x)$, and R_X is the set of functions from X to $\mathbb{R}^{\geq 0}$. See Appendix A for more notation.

Chapter 1. Introduction to the paper

ear dominance Θ is defined by $g \in \Theta_X(f)$ if and only if

$$\exists c \in \mathbb{R}^{>0}, \exists y \in \mathbb{R}^d : (g|(X \setminus X^{<y})) \leq c(f|(X \setminus X^{<y})). \quad (1.6)$$

for all $f, g \in R_X$, and all $X \in U$, where $U = \bigcup_{d \in \mathbb{N}^{>0}} \mathcal{P}(\mathbb{R}^d)$.

Note 1.7. This definition is given in [2] in the univariate form on \mathbb{N} , and generalized to \mathbb{N}^2 in a later exercise [2, page 53]. ■

Definition 1.8 (Cofinite linear dominance). *Cofinite linear dominance* Θ is defined by $g \in \Theta_X(f)$ if and only if^a

$$\exists c \in \mathbb{R}^{>0}, \exists A \in \bar{\mathcal{P}}(X) : (g|A) \leq c(f|A), \quad (1.9)$$

for all $f, g \in R_X$, and all sets X .

$$^a\bar{\mathcal{P}}(X) = \{A \in \mathcal{P}(X) : |X \setminus A| < \infty\}.$$

Note 1.10 (Symbol shapes). The drawing inside a given version of Θ above mimics the shape of its restriction sets in \mathbb{R}^2 . ■

Definition 1.11 (Full linear dominance). *Full linear dominance* Θ is defined by $g \in \Theta_X(f)$ if and only if

$$\exists c \in \mathbb{R}^{>0} : g \leq cf, \quad (1.12)$$

for all $f, g \in R_X$, and all $X \in U$, where U is the class of all sets.

Note 1.13 (Symbol shape). The drawing inside Θ mimics a line. ■

Note 1.14 (Linear dominance). We will often shorten full linear dominance to linear dominance. ■

Chapter 1. Introduction to the paper

Definition 1.15 (Affine dominance). *Affine dominance* \oplus is defined by $g \in \oplus_X(f)$ if and only if

$$\exists c \in \mathbb{R}^{>0} : g \leq cf + c, \quad (1.16)$$

for all $f, g \in R_X$, and all $X \in U$, where U is the class of all sets.

Note 1.17 (Symbol shape). The drawing inside \oplus mimics the plus operator. ■

Definition 1.18 (Related notations). Given an O-notation, we define the related notations $\Omega_X, \omega_X, \circ_X, \Theta_X$ as follows:

$$\begin{aligned} g \in \Omega_X(f) &\iff f \in O_X(g), \\ g \in \omega_X(f) &\iff g \notin O_X(f) \text{ and } f \in O_X(g), \\ g \in \circ_X(f) &\iff g \in O_X(f) \text{ and } f \notin O_X(g), \\ g \in \Theta_X(f) &\iff g \in O_X(f) \text{ and } f \in O_X(g). \end{aligned} \quad (1.19)$$

Note 1.20 (Equivalence from equality of O-sets). The expression $O_X(f) = O_X(g)$ is equivalent to $f \in \Theta_X(g)$; equality between O-sets can be used to establish the equivalence of functions. This is also true for $\Omega_X(f) = \Omega_X(g)$ and $\Theta_X(f) = \Theta_X(g)$. ■

Note 1.21 (Study of the O-notation suffices). It suffices to study the O-notation, since the related notations are completely determined by the O-notation. ■

We will show linear dominance \ominus to be the correct definition of O-notation for algorithm analysis.

1.4 Example analyses

In the following, we provide some example analyses of algorithms, and demonstrate why most of the definitions of O-notation in the previous section — except linear dominance — are not suitable for algorithm analysis.

Note 1.22 (Function from an expression). We will often denote a function in the parameter of O_X with an expression, as in $O_{\mathbb{N}}(n)$, where we actually mean $O_{\mathbb{N}}(f)$, with $f \in R_{\mathbb{N}}$ such that $f(n) = n$.

When the expression contains multiple symbols, as in $O_{\mathbb{N}^2}(n^2m)$, we interpret the symbols to be assigned to the input-tuple in alphabetical order, as in $O_{\mathbb{N}^2}((m, n) \mapsto n^2m)$. This is to acknowledge that $(m, n) \mapsto n^2m$ and $(n, m) \mapsto n^2m$ are different functions. ■

Note 1.23 (A cost model for examples). In the following examples, each addition operation $x + y$ costs one unit, while other operations cost nothing. ■

Algorithm 1 An algorithm which takes as input $(m, n) \in \mathbb{N}^2$, and outputs n , if $m = 0$, and 0 otherwise.

```

1: procedure COMPUTEONPLANE( $m, n$ )
2:    $j := 0$ 
3:   if  $m = 0$  then
4:     for  $i \in [0, n[_{\mathbb{N}}$  do
5:        $j := j + 1$ 
6:     end for
7:   end if
8:   return  $j$ 
9: end procedure

```

Algorithm 2 An algorithm which takes as input $n \in \mathbb{N}$, and returns n .

```

1: procedure MAPNATURALS TO PLANE( $n$ )
2:   return COMPUTEONPLANE( $0, n$ )
3: end procedure

```

Example 1.24 (Algorithm on \mathbb{N}^2). Consider Algorithm 1, which takes as input $(m, n) \in \mathbb{N}^2$. The cost function for this algorithm is $f \in R_{\mathbb{N}^2}$ such

that

$$f(m, n) = \begin{cases} n, & m = 0, \\ 0, & m > 0, \end{cases} \quad (1.25)$$

We can show that $f \in \Theta_{\mathbb{N}^2}(0)$. \triangle

Example 1.26 (Calling algorithm on \mathbb{N}^2). Consider Algorithm 2, which takes as input $n \in \mathbb{N}$, and outputs the result of Algorithm 1 at $(0, n)$. The cost function for this algorithm is $g \in R_{\mathbb{N}}$ such that

$$g(n) = n. \quad (1.27)$$

We can show that $g \in \Theta_{\mathbb{N}}(n)$. Therefore, by calling an algorithm which is $\Theta_{\mathbb{N}^2}(0)$, we get an algorithm which is $\Theta_{\mathbb{N}}(n)$. This shows that Θ is not suitable for a definition of O-notation in algorithm analysis. \triangle

Algorithm 3 An algorithm which takes as input $z \in \mathbb{Z}$, and returns $\max(-z, 0)$.

```

1: procedure COMPUTEONINTEGERS( $z$ )
2:    $j := 0$ 
3:   if  $z < 0$  then
4:     for  $i \in [0, -z]_{\mathbb{N}}$  do
5:        $j := j + 1$ 
6:     end for
7:   end if
8:   return  $j$ 
9: end procedure

```

Algorithm 4 An algorithm which takes as input $n \in \mathbb{N}$, and returns $\max(n, 0)$.

```

1: procedure MAPNATURALS TOINTEGERS( $n$ )
2:   return COMPUTEONINTEGERS( $-n$ )
3: end procedure

```

Example 1.28 (Algorithm on \mathbb{Z}). Consider Algorithm 3, which takes as input $z \in \mathbb{Z}$ and returns $\max(-z, 0)$. The cost function of this algorithm

is $f \in R_{\mathbb{Z}}$ such that

$$f(z) = \max(-z, 0). \quad (1.29)$$

We can show that $f \in \Theta_{\mathbb{Z}}(0)$. \triangle

Example 1.30 (Calling algorithm on \mathbb{Z}). Consider Algorithm 4, which takes as input $n \in \mathbb{N}$, and outputs the result of Algorithm 3 at $-n$. The cost function of this algorithm is $g \in R_{\mathbb{N}}$ such that

$$g(n) = n. \quad (1.31)$$

We can show that $g \in \Theta_{\mathbb{N}}(n)$. Therefore, by calling an algorithm which is $\Theta_{\mathbb{Z}}(0)$, we get an algorithm which is $\Theta_{\mathbb{N}}(n)$. This shows that Θ is not suitable for a definition of O-notation in algorithm analysis. \triangle

Algorithm 5 An algorithm which takes as input $n \in \mathbb{N}$ and evaluates a sub-algorithm n times at 0.

```

1: procedure G( $n$ )
2:   for  $i \in [0, n[_{\mathbb{N}}$  do
3:     F(0)
4:   end for
5: end procedure

```

Example 1.32 (Course-exercise). Consider Algorithm 5, which takes as input $n \in \mathbb{N}$ and calls a sub-algorithm F at 0 repeatedly n times. Denote the cost function of F by $f \in R_{\mathbb{N}}$, and the cost function of G by $g \in R_{\mathbb{N}}$. Suppose $f \in \Theta_{\mathbb{N}}(1)$. What is $\Theta_{\mathbb{N}}(g)$?

The given information is not sufficient to solve this problem. In particular, let $f_1, f_2 \in R_{\mathbb{N}}$ be such that $f_1(n) = 1$, and

$$f_2(n) = \begin{cases} 0, & n = 0, \\ 1, & n > 0. \end{cases} \quad (1.33)$$

Chapter 1. Introduction to the paper

Then $f_1, f_2 \in \Theta_{\mathbb{N}}(1)$, but

$$\begin{aligned} g_1 &\in \Theta_{\mathbb{N}}(n), \\ g_2 &\in \Theta_{\mathbb{N}}(0), \\ \Theta_{\mathbb{N}}(n) \cap \Theta_{\mathbb{N}}(0) &= \emptyset. \end{aligned} \tag{1.34}$$

This shows that Θ is not suitable for a definition of O-notation in algorithm analysis.

There is a fundamental difference between consuming resources ($f(0) > 0$) and not consuming resources ($f(0) = 0$), which Θ ignores here.

In contrast, $g \in \Theta_{\mathbb{N}}(n)$ provided $f \in \Theta_{\mathbb{N}}(1)$, as expected. \triangle

Algorithm 6 An algorithm which takes as input $x \in \mathbb{R}^{>0}$ and returns $\max(\lceil -\log_2(x) \rceil, 0) \in \mathbb{N}$.

```

1: procedure DOUBLES_TO_ONE( $x$ )
2:    $k \leftarrow 0$ 
3:    $y \leftarrow x$ 
4:   while  $y < 1$  do
5:      $y \leftarrow 2y$ 
6:      $k \leftarrow k + 1$ 
7:   end while
8:   return  $k$ 
9: end procedure

```

Algorithm 7 An algorithm which takes as input $n \in \mathbb{N}$, and returns n .

```

1: procedure IDENTITY( $n$ )
2:   return DOUBLES_TO_ONE( $2^{-n}$ )
3: end procedure

```

Example 1.35 (Algorithm on $\mathbb{R}^{>0}$). Consider Algorithm 6, which takes as input $x \in \mathbb{R}^{>0}$ and outputs $\max(\lceil -\log_2(x) \rceil, 0) \in \mathbb{N}$; this is the number of times that x must be doubled to grow ≥ 1 . The cost function of this

algorithm is $f \in R_{\mathbb{R}^{>0}}$ such that

$$f(x) = \max(\lceil -\log_2(x) \rceil, 0). \quad (1.36)$$

We can show that $f \in \Theta_{\mathbb{R}^{>0}}(0)$. \triangle

Example 1.37 (Calling an algorithm on $\mathbb{R}^{>0}$). Consider Algorithm 7, which takes in $n \in \mathbb{N}$, and outputs the result of Algorithm 6 at 2^{-n} . The cost function of this algorithm is $g \in R_{\mathbb{N}}$ such that

$$g(n) = n. \quad (1.38)$$

We can show that $g \in \Theta_{\mathbb{N}}(n)$. Therefore, by calling an algorithm which is $\Theta_{\mathbb{R}^{>0}}(0)$, we get an algorithm which is $\Theta_{\mathbb{N}}(n)$. This shows that Θ is not suitable for a definition of O-notation in algorithm analysis. \triangle

Algorithm 8 An algorithm which takes as input $n \in \mathbb{N}$, and outputs n .

```

1: procedure ALMOSTIDENTITY( $n$ )
2:    $j := 0$ 
3:   if  $n = 4$  then
4:     return 4
5:   end if
6:   for  $i \in [0, n[_{\mathbb{N}}$  do
7:      $j := j + 1$ 
8:   end for
9:   return  $j$ 
10: end procedure

```

Example 1.39 (Almost identity on \mathbb{N}). Consider Algorithm 8, which takes as input $n \in \mathbb{N}$, and outputs n . The cost function of this algorithm is $f \in R_{\mathbb{N}}$ such that

$$f(n) = \begin{cases} n, & n \neq 4, \\ 0, & n = 4. \end{cases} \quad (1.40)$$

We can show that $f \in \Theta_{\mathbb{N}}(n)$ and $f \in \Theta_{\mathbb{N}}(n)$. \triangle

Algorithm 9 An algorithm which takes as input $(m, n) \in \mathbb{N}^2$ and outputs n .

```

1: procedure SECONDCOMPONENT( $m, n$ )
2:   return ALMOSTIDENTITY( $n$ )
3: end procedure

```

Example 1.41 (Calling an almost identity). Consider Algorithm 9, which takes as input $(m, n) \in \mathbb{N}^2$, and outputs n . The cost function of this algorithm is $g \in R_{\mathbb{N}^2}$ such that

$$g(m, n) = \begin{cases} n, & n \neq 4, \\ 0, & n = 4. \end{cases} \quad (1.42)$$

We can show that $g \in \mathfrak{o}_{\mathbb{N}^2}(n)$. Therefore, by calling an algorithm which is $\Theta_{\mathbb{N}}(n)$, we get an algorithm which is $\mathfrak{o}_{\mathbb{N}^2}(n)$. This shows that Θ is not suitable for a definition of O-notation in algorithm analysis. \triangle

Note 1.43 (Blow-up and zeros). There are two kinds of problems with those definitions of O-notations — such as \mathfrak{O} , Θ , \mathfrak{O} — which ignore points from the input-set.

If the definition ignores an infinite set — as in $\mathfrak{O}_{\mathbb{N}^2}$, $\Theta_{\mathbb{Z}}$, and $\mathfrak{O}_{\mathbb{R}^{>0}}$ — then it is possible to construct a cost function to blow up in that ignored set, while not being reflected in the O-notation. This was demonstrated in Example 1.26 for $\mathfrak{O}_{\mathbb{N}^2}$, in Example 1.30 for $\Theta_{\mathbb{Z}}$, and in Example 1.37 for $\mathfrak{O}_{\mathbb{R}^{>0}}$. The last example demonstrates a blow-up in a bounded set.

If the definition ignores even a single point — as in \mathfrak{O} — then it is possible to construct a cost function with a zero at that point, while not being reflected in the O-notation. This was demonstrated in Example 1.32 for \mathfrak{O} .

Such definitions make it impossible to treat the cost functions as black boxes through their O-sets — something which is essential for cost analysis based on O-notation. \blacksquare

1.5 Some history of O-notation

The first reference to the O-notation seems to be that of Bachmann [3, page 401], in 1894, who gave a rather brief and informal definition of the O-notation:

Chapter 1. Introduction to the paper

... wenn wir durch das Zeichen $O(n)$ eine Grösse ausdrücken, deren Ordnung in Bezug auf n die Ordnung von n nicht überschreitet; ...

which we translate as: when we use the symbol $O(n)$ to represent some quantity, its order w.r.t. n does not exceed the order of n .

Landau [4, page 31], in 1909, put this definition on a formal grounding by

$$f \in O(g) : \Longleftrightarrow \exists c \in \mathbb{R}^{>0}, \exists N \in \mathbb{N}, \forall n \geq N : f(n) \leq cg(n), \quad (1.44)$$

for all $f, g: \mathbb{N} \rightarrow \mathbb{R}$. This is asymptotic linear dominance on \mathbb{N} . On page 883, Landau credits this definition to [3].

Both Bachmann and Landau were writing about analytic number theory, which is a branch of mathematics studying the properties of integers using tools from analysis. Correspondingly, the O -notation was constructed to address the needs in this field, such as to bound the error of truncating a series. Landau's definition was adopted by computer science, and remains the most wide-spread definition of O -notation on \mathbb{N} to this day.

In [5], in 1968, Knuth defined the O -notation as

$$f \in O(g) : \Longleftrightarrow \exists c \in \mathbb{R}^{>0} : |f| \leq c|g|, \quad (1.45)$$

for all $f, g: S \rightarrow \mathbb{R}$, where $S = \mathbb{N}$, or S is an interval of \mathbb{R} . Knuth credited the definition to Bachmann [3]. However, Knuth's definition probably contained an omission, since it does not correspond to Bachmann's definition, at least by Landau's interpretation of asymptotic linear dominance. This view is supported by that in [6], in 1973, the definition was replaced with Equation 1.44. To the best of my knowledge, in both editions of the book Knuth uses the O -notation solely to bound the error of truncating a series, and not for the analysis of algorithms. For analysis of algorithms, he deals with explicit bounds instead.

It is unclear to me when exactly the O -notation was first used in the analysis of algorithms. The earliest reference we can find is [7], from 1972. The book [8], from 1974, uses the O -notation in a modern way in the analysis of algorithms, with the following definition on \mathbb{N} :

A function $g(n)$ is said to be $O(f(n))$ if there exists a constant c such that $g(n) < cf(n)$ for all but some finite (possibly empty) set of nonnegative values for n .

This is cofinite linear dominance on \mathbb{N} .

Chapter 1. Introduction to the paper

On [8, page 39], an exercise encourages to study the properties of the following definition of O-notation on \mathbb{N} :

$$f \in O(g) :\iff \exists c \in \mathbb{R}^{>0} : f \leq cg. \quad (1.46)$$

This is linear dominance on \mathbb{N} .

On [8, page 39], another exercise asks for the equivalence of cofinite affine dominance on \mathbb{N} and cofinite linear dominance on \mathbb{N} under certain assumptions, with *cofinite affine dominance* on \mathbb{N} defined by

$$f \in O(g) :\iff \exists c, d \in \mathbb{R}^{>0} : f \leq cg + d, \quad (1.47)$$

for all but a finite number of input-arguments.

Prior to this paper, there were no attempts at studying the O-notation systematically, in order to check whether the definition from analytic number theory was suitable for the analysis of algorithms. Indeed, it seemed as if the definition were simply a matter of taste; to quote [9],

On the basis of the issues discussed here, I propose that members of SIGACT, and editors of computer science and mathematics journals, adopt the O, Ω , and Θ notations as defined above, unless a better alternative can be found reasonably soon.

The two exercises in [8] show that the possibility of using a different definition of the O-notation was certainly noted, but that an argument to favor one over another was missing.

Related notations

There are several other notations which resemble the way the O-notation works, which perhaps force strictness, reverse the ordering, or otherwise vary the ordering. In [4, page 61], Landau defined the o-notation as

$$f \in o(g) :\iff \lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = 0, \quad (1.48)$$

for all $f, g: \mathbb{R} \rightarrow \mathbb{R}$. On page 883, Landau states that the o-notation is his own.

Knuth [9], in 1976, defined the Ω -notation as $f \in \Omega(g) :\iff g \in O(f)$, the Θ -notation as $f \in \Theta(g) :\iff f \in O(g)$ and $g \in O(f)$, and the ω -notation as $f \in \omega(g) :\iff g \in o(f)$, for all $f, g: \mathbb{N} \rightarrow \mathbb{R}$. This was to remedy the occasional misuse where O was used in place of the now-defined Θ or Ω .

Vitányi [10] argued that Ω should be defined asymmetrically by

$$f \in \Omega(g) :\iff \exists c \in \mathbb{R}^{>0}, \forall y \in \mathbb{N}, \exists x \geq y : f(x) \geq cg(x). \quad (1.49)$$

1.6 Objective and contributions

The objective of this paper is to provide a rigorous mathematical foundation for the O -notation in algorithm analysis. In the following is a list of problems this paper solves.

Problem 1: What is a cost function?

A cost function of an algorithm has invariably been explained in books — such as [8] and [2] — as a function which maps the *size of the input* to $\mathbb{R}^{\geq 0}$. Unfortunately, the term *size* has never been defined formally. Instead, the concept has been demonstrated by examples such as the length of a sequence, the number of bits in a number, or the pair (n, m) , where n is the number of vertices in a graph, and m is the number of edges in a graph.

On Turing machines, input-size refers to the number of consecutive bits on the input-tape that need to be written in order to set the initial state. If the input is not in binary form already, the input has to be encoded as such, and the encoding specified. This definition of input-size *does not* formalize the intuitive concept of input-size used in books. For example, an encoding of a graph could have an input-size proportional to $m + n$. However, there are graph algorithms whose cost-functions are only dependent on n , and graph algorithms whose cost-functions² depend non-trivially on both m and n . Such cost-functions cannot be captured as a function of $m + n$.

Here are some examples of the confusion this concept creates. The prototypical examples of *size* are cardinality and volume, with an implied linear order. But how does one define a linear order *sensibly* on, say, the pairs (n, m) ? If that does not make sense, should the concept of size be extended to a partial order, or even a preorder? When the cost function is a function of input size, what does it mean to remap the input-sizes by function composition from the right (i.e. $f \circ s$)? When this paper allows the domain of the cost function to be arbitrary, such as \mathbb{Z} , \mathbb{R} or \mathbb{C} , what exactly does it mean for an input to have size -5 , 4.5 or $2 + 4i$?

The present paper resolves these questions as follows. The term *input-size* does not make sense for an arbitrary model of computation. Instead, this paper defines the cost function as a function which maps the input-set to $\mathbb{R}^{\geq 0}$ — the most detailed characterization of the cost function of an algorithm. Function composition from the right is used to access the input in a different order

²e.g. $m^2 \log_2(n + 1)$.

— as when the algorithm is called as a sub-algorithm of another algorithm. Compatibility with this operation means that merely a different access-order cannot be made to show inconsistencies in the O-notation.

The grouping of the input-set need only be done if the cost function by itself is too complex to make sense of. In such a case, we choose a grouping property — such as the pair (n, m) for graphs — and partition the input-set based on that property. No additional properties — such as an ordering — are required from a grouping property.

Problem 2: Why should the O-notation be defined as it is?

Introductory texts on algorithms, such as [1], traditionally explain the definition of the O-notation by the merits it has, such as abstracting out differences in the speeds of otherwise identical machines. The asymptotic part — ignoring parts of the input — is explained by being essential for the notation to concentrate on the scaling behavior on large inputs, and not on some possible artifacts on small inputs.

Unfortunately, these explanations fail to pinpoint why we should not pick any other definition with seemingly the same properties, or whether it is necessary at all to ignore part of the input to gain sensitivity to the scaling behavior. Indeed, we show that, in general, no part of the input can be ignored without the notation failing in some way. The first hint towards this fact is seen by making an exhaustive list³ of all the rules we need the O-notation to possess; to express each one, none requires any restrictions — such as an ordering or a sense of convergence — on the input domain.

The present paper resolves this question as follows. First, the O-notation should provide us with the simplification we wanted. Second, it should impose an order on the functions. Third, it should be compatible with all those operations — such as addition, multiplication, and scalar multiplication — that are needed to combine the actual cost functions during explicit analysis. Specifically, the following procedures must yield the same answer:

- Apply a sequence of operations to cost functions, and then simplify the result with the O-notation.
- Simplify every cost function with the O-notation, and then apply the same operations as above to the O-sets.

³An exhaustive list can be found in Table 3.1.

Chapter 1. Introduction to the paper

Perhaps the most important operation for algorithm analysis — identified in this paper — is that of *function composition* from the right, which corresponds to reordering or remapping the input. Compatibility with function composition corresponds to the requirement that the cost function of an algorithm must stay consistent when it is called with distorted input as a sub-algorithm of another algorithm. This is a strong requirement: it forces the definitions of O-notations on *different domains* to be consistent with each other. When combined with the other requirements, this turns the definition from a matter of taste to a unique one.

Problem 3: How should the multivariate O-notation be defined?

When analyzing a graph algorithm, the input-set is often grouped by a two-dimensional property (n, m) , where n is the number of vertices in the graph, and m is the number of edges in the graph. To provide the complexity in the O-notation under such a grouping, the O-notation must also be defined in \mathbb{N}^2 .

Textbooks on algorithms approach the definition on \mathbb{N}^2 in two ways. The first way — exemplified in [8, page 312] — is to define the O-notation only in \mathbb{N} , and then silently continue using the notation over multiple variables too. That is, with no definition. The second way — exemplified in [1] and [2] — is to define the O-notation in \mathbb{N} , and later provide some definition for a generalization to \mathbb{N}^2 , with the implication that the generalization does not bring anything new. Among all the books we have looked at, [1] and [2] were the only ones to provide a definition for \mathbb{N}^2 — although different ones. The former provided asymptotic linear dominance, while the latter provided coasymptotic linear dominance.

The first documented symptom of a non-trivial generalization to \mathbb{N}^2 was given by Howell [11], who demonstrated a flaw⁴ in asymptotic linear dominance on \mathbb{N}^2 . Perhaps it was for this issue, or a similar issue, that [2] changed the definition to coasymptotic linear dominance.

How can one be sure that *this* definition as coasymptotic linear dominance has the desirable properties? In [11], Howell thought he had shown that a definition with desirable properties is impossible in \mathbb{N}^d . However, we will show in Appendix B that — due to excessively strong assumptions — Howell only showed that asymptotic linear dominance does not have the desirable properties.

⁴We will take a closer look at this in Appendix B

Chapter 1. Introduction to the paper

The present paper resolves this question by showing that the only definition which has the desirable properties is linear dominance; the multivariate definition is merely an instance of the definition on \mathbb{N}^d .

Problem 4: How are the desirable properties connected to each other?

To make the theory in this paper robust to variation, and to self-test it, this paper studies the connections between the desirable properties of the O-notation. This study reveals that certain combinations of properties imply other properties, and that certain combinations of properties are equivalent to each other. As a side effect, this paper does not merely study the O-notation in algorithm analysis, but also other possible or historical definitions of the O-notation.

The study on the connections culminates in the discovery of the 8 primitive properties, which imply all the desirable properties. Therefore, the question of whether the desirable properties are reasonable reduces to asking whether the primitive properties are reasonable.

Problem 5: Do Master theorems hold for linear dominance?

Master theorems are useful tools for analyzing the cost function of a recursive algorithm up to an O-equivalence. Therefore, it is important to make sure that the conclusions of Master theorems still hold after adopting linear dominance as the definition.

The present paper provides the necessary proofs in Appendix F. In summary, Master theorems hold similarly as with asymptotic linear dominance, with the simplification that there is no need for so-called regularity conditions [2].

Problem 6: How do existing definitions compare to each other?

The present paper provides in Appendix G a comparison between various definitions of the O-notation, based on which primitive properties each fulfills. The study of how exactly the existing definitions fail the primitive properties is important, because a priori there is a risk that they may have produced incorrect analyses.

Fortunately, coasymptotic linear dominance on \mathbb{N}^d — on positive functions — is equivalent to linear dominance on \mathbb{N}^d . This covers most of the analyses that have been done in algorithm analysis. In practice, the saving grace is that

the O -notation has been manipulated by assuming that the rules on \mathbb{N} work in general, and by implicitly assuming that sub-composability holds.

Problem 7: How should the related notations be defined?

There has been some confusion on how to define the related notations Ω , and o , as discussed in Section 1.5.

The present paper argues that these notations are merely different viewpoints of the same underlying order-relation between cost functions; they are fixed by the definition of the O -notation.

Problem 8: Can existing techniques of analysis still be used?

The present paper shows in Section 6.2 that local linear dominance — a definition which covers most of the existing definitions — has an equivalent definition by limits. In addition, under suitable conditions, the other existing definitions are sometimes equivalent to linear dominance. Together, this allows to continue using existing analysis tools — such as taking limits — as before, provided one makes sure that there is a way to transfer the result to linear dominance.

Problem 9: Point out misuses

The present paper points out in Section 2.8 some misuses which occur in actual publications in computer science. In particular, the O -notation is sometimes used as a general something-like operator to generalize statements, where it actually does not make sense. It is my hope that making such misuses explicit eventually improves communication between computer scientists.

1.7 Relation to publications

The present paper extends a peer-reviewed publication in the Bulletin of EATCS [12], a doctoral thesis with the same name, as well as various non-peer-reviewed versions in Arxiv [13].

Most of the proofs have been deferred to appendices. This is to make the paper readable to the largest possible audience, emphasizing the intuitive ideas. This does not reflect an ordering in importance — to us, the proofs are the most important part of this paper.

The research and writing related to this paper, and the above publications, were done solely by KR; the co-authors checked.

1.8 Automated checking

This paper was written in such a way that the dependencies between the theorems can be — and have been — checked by a machine. Each theorem in the \LaTeX source is annotated — in a lightweight, but machine-readable manner — with its assumed and implied properties. When the proof of Theorem A references Theorem B, it inherits the properties implied by B, from that point on, provided that the assumptions of B are satisfied at that point. For each theorem, the software checks that

- the assumptions of each referred theorem are satisfied, and that
- there are no extraneous assumptions.

When a property is not proved by referring to a theorem, it is explicitly marked proved by the writer, based on the preceding non-machine-checkable proof.

Such automatic checking was useful in the research phases to guarantee that theorems were not broken due to changing assumptions, or otherwise to point them out. The software for checking the dependencies, written in Python, can be obtained from my homepage.⁵

1.9 Outline of the paper

This paper is organized as follows. Chapter 1 introduces the paper, reviews the history of the topic, and lists the contributions of the paper. Chapter 2 provides a more formal introduction to the concepts related to algorithms and their analysis. Chapter 3 provides a list of desirable properties for an \mathcal{O} -notation. Chapter 4 proves that the primitive properties are equivalent to the definition of the \mathcal{O} -notation as linear dominance. Chapter 5 provides tools for working with the \mathcal{O} -notation, including Master theorems and \mathcal{O} -mapping rules. Chapter 6 studies local linear dominance, a generalization which covers most of the existing definitions. Chapter 7 concludes the paper.

Appendix A provides the notation. Appendix B reviews Howell’s counterexample in more detail. Appendix C shows that the desirable properties reduce to a set of 8 primitive properties which imply the other properties. Appendix D shows properties of local linear dominance. Appendix E shows that local linear dominance can be characterized by ratio-limits. Appendix F shows that Master theorems work as before under linear dominance. Appendix G compares several

⁵<http://kaba.hilvi.org>

Chapter 1. Introduction to the paper

definitions for an O-notation, and shows how each of them fail the desirable properties. Appendix H provides additional definitions for an O-notation, with an aim to show the minimality of pre-primitive properties. Appendix I provides some theory of partitioned sets. Appendix J provides some theory of preordered sets.

Chapter 2

Preliminaries

In this chapter we provide a brief introduction to algorithms, the computational model, the cost-model, the primitive properties, and the O-notation.

2.1 Algorithms

What is an algorithm? We adopt an extremely liberal, but completely formalized view: an *algorithm* is an abstract state machine [14, 15, 16].

A variable in an abstract state machine M is identified with a string, called a *(function) symbol*. Each symbol has an *arity* $n \in \mathbb{N}$, which gives the number of arguments the symbol accepts as input. A *(ground) term* is defined recursively as follows:

- a 0-ary symbol is a term, and
- if f is an n -ary symbol, and t_1, \dots, t_n are terms, then the string $f(t_1, \dots, t_n)$ is a term, and
- there are no other terms.

The set of user-defined symbols, together with a small set of predefined symbols — such as true, false, or, and, not, =, undef — is called the *vocabulary* of the abstract state machine M .

Each n -ary symbol f is associated with a function $\bar{f}: S^n \rightarrow S$.¹ The function \bar{f} is an *interpretation* of f . The set S is the *base-set*, which is common to all

¹ $S^0 = \{\emptyset\}$.

Chapter 2. Preliminaries

interpretations. The *value* of a term t , denoted by $[t]$, is defined recursively as follows:

- if x is a 0-ary symbol, then $[x] = \bar{x}(\emptyset)$,
- if f is an n -ary symbol, for $n \in \mathbb{N}^{>0}$, and t_1, \dots, t_n are terms, then

$$[f(t_1, \dots, t_n)] = \bar{f}([t_1], \dots, [t_n]).$$

In the following, by $f: X \rightarrow Y$ — where f is a symbol, $X \subset S^n$, and $Y \subset S$ — we mean that $\bar{f}(X) \subset Y$. In addition, if x is a 0-ary symbol, then by $x \in X$ we mean that $x: \{\emptyset\} \rightarrow X$ in the previous sense.

Compared to an ordinary programming language, a 0-ary symbol corresponds to a variable, while an n -ary symbol, for $n \in \mathbb{N}^{>0}$, corresponds to an n -dimensional array — however, here the index can be an arbitrary set element.

The abstract machine specifies how the interpretations of symbols are to be modified at each step. The program driving the abstract machine is a finite sequence of conditional assignments of the form

if condition then

$t_1 := s_1$

\vdots

$t_n := s_n$

end if

where condition, t_1, \dots, t_n , and s_1, \dots, s_n are terms. The formula $t_i := s_i$ can be thought of as copying an element from an array to another, $[t_i] := [s_i]$, provided $[\text{condition}] = [\text{true}]$. This sequence of assignments is repeated until (possible) termination.

All of the assignments in a single step are carried out in parallel — not in sequence. For example, $t_1 := s_1$ followed by $s_1 := t_1$ causes t_1 and s_1 to swap values in the next step.

The basic definition of abstract state machines is both simple, and extremely general. The generality derives from the virtue of making the whole of set-theory available for modeling variables. Further abstraction-tools — such as sequential composition, sub-machine calls, local variables, and return values — are constructed over this basic definition. For example, Turbo-ASMs [16] provide such features. From now on, we will assume that such abstraction tools have already been defined.

Chapter 2. Preliminaries

Consider Algorithm 10, which is Newton’s method for finding local zeros of differentiable functions.² The input-symbols to this algorithm are a 1-ary continuously differentiable function $f: \mathbb{R} \rightarrow \mathbb{R}$, a 0-ary initial guess $x^* \in \mathbb{R}$, and a 0-ary error threshold $\varepsilon \in \mathbb{R}^{\geq 0}$; the input-set is $C_1(\mathbb{R} \rightarrow \mathbb{R}) \times \mathbb{R} \times \mathbb{R}^{\geq 0}$. The output — provided the algorithm terminates — is a 0-ary point $x \in \mathbb{R}$ such that $|f(x)| \leq \varepsilon$; the output-set is \mathbb{R} . Other symbols are a 1-ary symbol $': C_1(\mathbb{R} \rightarrow \mathbb{R}) \rightarrow C_1(\mathbb{R} \rightarrow \mathbb{R})$ — differentiation —, a 1-ary symbol $|\cdot|: \mathbb{R} \rightarrow \mathbb{R}^{\geq 0}$ — absolute value — a 2-ary symbol $>: \mathbb{R} \times \mathbb{R} \rightarrow \{0, 1\}$ — greater-than — and 2-ary symbols $-: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ and $/: \mathbb{R} \times \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}$ — subtraction and division. We have used infix notation for subtraction, division, and greater-than; postfix notation for differentiation; and midfix notation for the absolute value.

Algorithm 10 Newton’s method for finding an element $x \in X$, such that $|f(x)| \leq \varepsilon$, for a continuously differentiable function $f \in C_1(\mathbb{R} \rightarrow \mathbb{R})$.

```

1: procedure FINDZEROORHANG( $f, x^*, \varepsilon$ )
2:    $x := x^*$ 
3:   while  $|f(x)| > \varepsilon$  do
4:      $x := x - f(x)/f'(x)$ 
5:   end while
6:   return  $x$ 
7: end procedure

```

Algorithm 10 reads like pseudo-code, but is a completely formalized abstract state machine. With abstract state machines, the programmer is free to use the most fitting abstraction for the problem at hand. It should be clear how using such an abstract programming language enhances communication between software engineers and domain experts (e.g., physicists).

Termination is not required for an algorithm; consider for example an operating system. Depending on the input, Algorithm 10 may terminate, or not. Suppose we require Algorithm 10 to always terminate. To satisfy this requirement, the programmer can either restrict the input-set to terminating inputs, or to modify the algorithm — perhaps by limiting the number of iterations. From now on, we assume every analyzed algorithm to terminate.

A software development project utilizing abstract state machines starts by creating the most abstract description of the software as an abstract state machine, called the *ground model* [16]. This model captures the requirements,

²This example generalizes an example from [17] where Newton’s method is formalized as an algorithm for real rational functions under the real-RAM model.

Chapter 2. Preliminaries

but does not provide any additional details on how the goals are to be attained. The project then proceeds to refine the model until it can be implemented in a concrete programming language. The model — in all stages — is used for verification and validation, and may even be used to generate code automatically.

This brief introduction to abstract state machines is to encourage the reader to look beyond the Church-Turing paper, and to realize the usefulness of even the most abstract algorithms — not just those computable algorithms which work with natural numbers. These are algorithms which take arbitrary sets as input, and produce arbitrary sets as output. To analyze such abstract algorithms, we need correspondingly abstract tools.

Here are some examples of input-sets that algorithms can have, or by which they can be modeled.

Example 2.1 (Algorithms on integers). An algorithm which takes integers as input could be modeled by \mathbb{Z}^d . \triangle

Example 2.2 (Algorithms on floating-point numbers). An algorithm which takes floating-point numbers as input could be modeled by \mathbb{R}^d . \triangle

Example 2.3 (Algorithms on complex numbers). An algorithm which takes complex numbers as input could be modeled by \mathbb{C}^d . \triangle

Example 2.4 (Algorithms on matrices). An algorithm which takes a matrix as input could be modeled by $\mathbb{C}^{m \times n}$. \triangle

Example 2.5 (Algorithms on sequences). An algorithm which takes a sequence in X as input could be modeled by

$$X^* = \bigcup_{d \in \mathbb{N}} X^d. \quad (2.6)$$

\triangle

Example 2.7 (Algorithms on combinations of the above). An algorithm which takes a sequence in X and an integer as input could be modeled by $X^* \times \mathbb{Z}$. \triangle

Example 2.8 (Algorithms on graphs). A graph algorithm is often analyzed under the domain $\mathbb{N} \times \mathbb{N}$. This domain comes from the worst-case analysis (say), where the input is grouped according to the number of vertices and edges. \triangle

Example 2.9 (Algorithms in computational geometry). An algorithm in computational geometry [18] works directly with points in \mathbb{R}^d , and may not mention floating-point numbers at all. The worst-case (or direct) analysis may require the domain \mathbb{R}^d . \triangle

Example 2.10 (Algorithms on different dimensions). We may be interested in how seeing how a geometric algorithm in \mathbb{R}^d scales with respect to the dimension $d \in \mathbb{N}$. In this case the input-set may consist of a union of data structures (e.g. range tree) for different dimensionalities. Dimensionality can then be used as a grouping property. \triangle

Example 2.11 (Approximation schemes). When considering an NP-hard problem, we may consider an approximation scheme [2], where $\varepsilon \in \mathbb{R}^{\geq 1}$ specifies the quality of the approximation; ε is specified as part of the input. \triangle

2.2 Computational model and cost-model

Before an algorithm can be written, the writer must decide on the model of computation. A *model of computation* is a mathematical structure, and a set of atomic operations which manipulate that structure. Some models of computation are the Turing machine, the random-access machine (RAM) [19], the real-RAM [17], and the abstract state machine.

The result of complexity analysis — for a given model of computation, a given algorithm, and a given resource — is a function $f: X \rightarrow \mathbb{R}^{\geq 0}$, which

Chapter 2. Preliminaries

provides for each *input* of the algorithm a non-negative real number. This number tells how much of that resource the algorithm consumes with the given input. As discussed in Section 2.1, the input-set X can be arbitrary.

Before a complexity analysis can be carried out, the analyst must decide on the cost-model. A *cost-model* specifies the amount of resources that running an atomic operation takes on a given input. A given computational model can assume different cost-models. When the cost-model is unspecified — as it often is — the cost of each atomic operation is assumed to be one unit irrespective of input.

Example 2.12 (Constant cost-models). The most common cost-model is the unit-cost model, which counts the number of performed atomic operations. Zero costs can be used to concentrate the interest to specific resources, such as order-comparisons. \triangle

Example 2.13 (Non-constant cost-models). An example of a non-constant cost-model is to assign the addition of natural numbers^a a cost which is proportional to the logarithms of the arguments, so as to be proportional to the number of bits in their binary representations. \triangle

^aAssuming the computational model supports such an operation.

Example 2.14 (Cost-models for abstract state machines). An abstract state machine specifies costs for reading or writing a memory location through a given symbol. Reading an addition symbol $+$ at $(x, y) \in \mathbb{N}^2$ — $x + y$ — could be assigned a logarithmic cost as described above. \triangle

2.3 Primitive properties

Complexity analysis aims to classify and compare algorithms based on their resource consumptions; the less an algorithm uses resources to solve a given problem, the better it is compared to other algorithms which solve the same problem. The cost functions of the algorithms to solve a problem $P: X \rightarrow Y$ are elements of R_X .³ The most general way to compare them is to define a

³ $R_X = X \rightarrow \mathbb{R}^{\geq 0}$; see Appendix A.

Chapter 2. Preliminaries

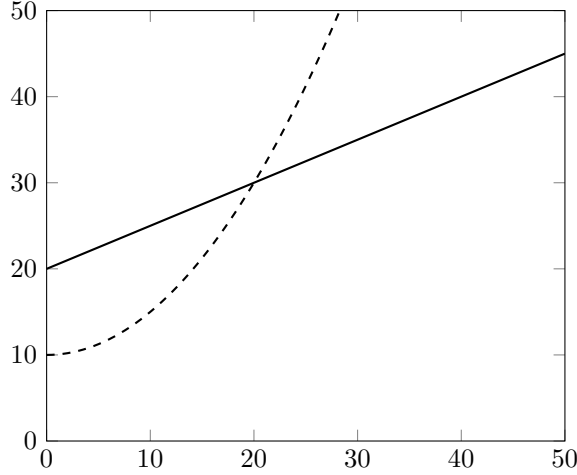


Figure 2.1: Here $f_A, f_B \in R_{\mathbb{N}}$ are such that $f_A(n) = n/2 + 20$ (solid line), and $f_B(n) = n^2/20 + 10$ (dashed line). Should $f_A \preceq_X f_B$, $f_B \preceq_X f_A$, $f_A \approx_X f_B$, or should f_A and f_B be incomparable?

preorder⁴ $\preceq_X: R_X \leftrightarrow R_X$. This relation should capture the intuitive concept of a cost function $f \in R_X$ being either better than or equivalent to the cost function $g \in R_X$ — in some sense. For brevity, we use the term *dominated by*.

Note 2.15 (Not worse). It is tempting to use the phrase *not worse than* instead of *better or equivalent*. However, the former means better, equivalent, or *incomparable*, which is not what we want. ■

Note 2.16 (Cost function of an algorithm). Given an algorithm $A: X \rightarrow Y$, we shall denote its cost function by $f_A \in R_X$. ■

Example 2.17 (Essentially better?). Consider Figure 2.1, where $f_A, f_B \in R_{\mathbb{N}}$ are such that $f_A(n) > f_B(n)$ for $n \in [0, 20)_{\mathbb{N}}$, and $f_A(n) \leq f_B(n)$, for

⁴A preorder on a set X is a reflexive and transitive relation $\preceq_X: X \leftrightarrow X$.

Chapter 2. Preliminaries

$n \in [20, \infty)_{\mathbb{N}}$. We would then be inclined to say that A has a better cost function, since $f_A(n)$ is small anyway on the finite interval $n \in [0, 20)_{\mathbb{N}}$; f_A is “essentially” better than f_B . How can this intuition be formalized? \triangle

To decide between various definitions, we reflect on the fundamental properties that the analyst needs to complete his/her complexity analysis. An obvious *dominance property* is

order-consistency

if an algorithm A never uses more resources than algorithm B , then f_A is better than or equivalent to f_B :

$$f_A \leq f_B \implies f_A \preceq_X f_B. \quad (2.18)$$

Note 2.19 (Reflexivity). In particular, order-consistency implies that \preceq_X is reflexive: $f \preceq_X f$. ■

Since we want \preceq_X to be a preorder, we also need

transitivity

if f_A is better than or equivalent to f_B , and f_B is better than or equivalent to f_C , then f_A is better than or equivalent to f_C :

$$(f_A \preceq_X f_B \text{ and } f_B \preceq_X f_C) \implies f_A \preceq_X f_C. \quad (2.20)$$

In addition to being a preorder, the dominance relation must preserve the *structure* present in algorithms: conditional branching, calls with transformed input, and looping.

Algorithm 11 An algorithm to demonstrate locality. Improving sub-algorithm F for even integers improves the whole algorithm. We assume resources are only spent in F .

```

1: procedure BRANCHING( $n$ )
2:   if mod( $n$ , 2) = 0 then
3:     return  $F(n)$ 
4:   end if
5:   return 0
6: end procedure

```

Chapter 2. Preliminaries

locality

substituting a sub-algorithm *for a subset of the input-set* with a better or equivalent sub-algorithm results in a better or equivalent algorithm overall:

$$(\forall i \in [1, n]_{\mathbb{N}} : (f|_{A_i}) \preceq_{A_i} (g|_{A_i})) \implies f \preceq_X g, \quad (2.21)$$

for every finite cover $A_1, \dots, A_n \subset X$. Algorithm 11 demonstrates locality.

Algorithm 12 An algorithm to demonstrate sub-composability. Improving sub-algorithm F improves the whole algorithm. We assume resources are only spent in F .

```

1: procedure CALLING( $n$ )
2:   return  $F(s(n))$ 
3: end procedure

```

sub-composability

substituting a sub-algorithm *called with transformed input* with a better or equivalent sub-algorithm results in a better or equivalent algorithm overall:

$$f_A \preceq_X f_B \implies f_A \circ s \preceq_Y f_B \circ s, \quad (2.22)$$

for all $s: Y \rightarrow X$. Algorithm 12 demonstrates sub-composability.

Algorithm 13 An algorithm to demonstrate sub-homogeneity in \mathbb{N} . Improving sub-algorithm F improves the whole algorithm. We assume resources are only spent in F .

```

1: procedure LOOPING( $n$ )
2:   for  $i \in [0, u(n)]_{\mathbb{N}}$  do
3:      $F(n)$ 
4:   end for
5: end procedure

```

sub-homogeneity in \mathbb{N}

substituting a sub-algorithm *in a loop* with a better or equivalent sub-algorithm results in a better or equivalent algorithm overall:

$$f_A \preceq_X f_B \implies u f_A \preceq_X u f_B, \quad (2.23)$$

Chapter 2. Preliminaries

for all $u \in R_X$ such that $\vec{u}(X) \subset \mathbb{N}$. Algorithm 13 demonstrates sub-homogeneity in \mathbb{N} .

sub-homogeneity in $1/\mathbb{N}^{>0}$

if an algorithm that loops sub-algorithm A is better than or equivalent to an algorithm that loops sub-algorithm B, then it is because f_A is better than or equivalent to f_B :

$$f_A \preceq_X f_B \implies u f_A \preceq_X u f_B, \quad (2.24)$$

for all $u \in R_X$ such that $\vec{u}(X) \subset 1/\mathbb{N}^{>0}$.

Note 2.25 (Many preorders). These properties reveal that the preorders in different sets cannot be defined independently of each other; they are tightly connected by locality and sub-composability. Rather, the problem is to find a consistent class of preorders $\{\preceq_X : X \in U\}$, where U is a given universe. ■

Note 2.26 (Trivial dominance). A problem with the listed properties thus far is that the trivial preorder $\preceq_X = R_X \times R_X$, for all $X \in U$, fulfills them all; then the functions in R_X are all equivalent. Therefore, for the comparison to be useful, we need to require \preceq_X to distinguish at least some functions in R_X , at least for some $X \in U$. ■

The primitive *non-triviality property* is:

one-separation

$$n \not\preceq_{\mathbb{N}^{>0}} 1.$$

Note that $1 \preceq_{\mathbb{N}^{>0}} n$ already holds by order-consistency; one-separation prevents the order from collapsing due to equivalence.

Finally, there is the question of robustness. Writing an algorithm is an iterative process, which causes the cost function of an algorithm to change constantly. If every change is reflected in the ordering, then each change invalidates an existing complexity analysis of the algorithm. Worse, the change invalidates all the analyses of the algorithms which use the changed algorithm as a sub-algorithm. Since an algorithm may face hundreds or thousands of changes before stabilising into something usable in practice, such an approach

Chapter 2. Preliminaries

is infeasible. Therefore, the ordering needs to introduce identification, to make it robust against small changes in the algorithms. But what is a small change?

The key realization is that for a given algorithm $F \in \mathcal{A}(P)$ it is often easy, with small changes, to produce an algorithm $G \in \mathcal{A}(P)$, for which the improvement-ratio⁵ f_F/f_G stays bounded.⁶ In contrast, obtaining an unbounded improvement-ratio often requires considerable insight and fundamental changes to the way an algorithm works — a new way of structuring data and making use of it. This definition of interesting provides the desired robustness against small changes in algorithms.

The primitive *abstraction property* is:

scale-invariance

if $f_A \preceq_X f_B$, then $f_A \preceq_X \alpha f_B$, for all $\alpha \in \mathbb{R}^{>0}$.

Note 2.27 (Dominance is linear dominance). We will show in Chapter 4 that the 8 primitive properties are equivalent to the definition of dominance as linear dominance:

$$f \preceq_X g \iff \exists c \in \mathbb{R}^{>0} : f \leq cg. \quad (2.28)$$

■

Note 2.29 (Generators and propagators). When the primitive properties are used as axioms, every proof of $f \preceq_X g$ must appeal to order-consistency, and every proof of $f \not\preceq_X g$ must appeal to one-separation. All the other primitive properties propagate these results. ■

Definition 2.30 (Related relations). A preorder \preceq_X induces the fol-

⁵Assuming $f_F > 0$.

⁶For example, change to use Single Instruction Multiple Data (SIMD) instructions.

lowing related relations in R_X :

$$\begin{aligned}
 f \approx_X g &\iff f \preceq_X g \text{ and } g \preceq_X f, \\
 f \prec_X g &\iff f \preceq_X g \text{ and } f \not\approx_X g, \\
 f \succ_X g &\iff g \prec_X f, \\
 f \succeq_X g &\iff g \preceq_X f.
 \end{aligned}
 \tag{2.31}$$

Note 2.32 (Strict comparison is weaker). In a preorder which is not a partial order, the \prec_X alone cannot be used to deduce whether $f \approx_X g$; it is a weaker concept than \preceq_X . However, providing \prec_X and \approx_X suffices. This is in contrast to a partial order, where \approx_X is known to be the set-equality in X . ■

2.4 Problem complexity

Apart from analyzing the resource-cost of a specific algorithm, complexity analysts also have a greater underlying goal: that of analyzing the resource-cost of the underlying problem $P: X \rightarrow Y$ itself — for a given computational model, a given cost-model, and a given resource. This activity divides into two sub-activities; finding a lower-bound and an upper-bound for the cost function of P .

Definition 2.33 (Lower bound). A cost function $f \in R_X$ is a *lower-bound* for a problem $P: X \rightarrow Y$, if $f \preceq_X f_A$, for all $A \in \mathcal{A}(P)$.

Definition 2.34 (Optimal cost function). A cost function $f \in R_X$ is *optimal* for a problem $P: X \rightarrow Y$, if f is a lower-bound for P , and $g \preceq_X f$ for each $g \in R_X$ a lower-bound for P .^a

^aThat is, f is a greatest lower-bound.

Note 2.35 (Optimal cost function may not exist). An optimal cost function may not exist for a problem, as shown in [Incompleteness] (2.83).

When it exists, the optimal cost function for P ,

$$\inf_{\preceq_X} \{f_A : A \in \mathcal{A}(P)\}, \quad (2.36)$$

is unique up to equivalence. ■

Definition 2.37 (Optimal algorithm). An algorithm $A \in \mathcal{A}(P)$ is *optimal*, if f_A is optimal for P .

Note 2.38 (Optimal algorithm may not exist). An optimal algorithm may not exist for a given problem. ■

Definition 2.39 (Upper bound). A cost function $f \in R_X$ is an *upper-bound* for a problem $P: X \rightarrow Y$, if $f_A \preceq_X f$, for some $A \in \mathcal{A}(P)$.

An upper-bound is found by finding an actual algorithm for solving P . While there are computational problems which cannot be solved at all,⁷ establishing at least one upper-bound for a solvable problem is often easy. These are the *brute-force* algorithms, which compute or check everything without making any use of the underlying structure in the problem.

2.5 O-notation

Definition 2.40 (O-notation). An *O-notation* over the universe U is a class of functions

$$\mathcal{O} := \{\mathcal{O}_X : R_X \rightarrow \mathcal{P}(R_X) : X \in U\}, \quad (2.41)$$

where

$$\mathcal{O}_X(f) = \{g \in R_X : g \preceq_X f\}. \quad (2.42)$$

⁷e.g., the halting problem under Turing machines.

Note 2.43 (Different symbols). The \mathbf{O} is used as a generic symbol for studying how the different desirable properties of \mathbf{O} interact with each other. We will use drawings inside the \mathbf{O} symbol for specific versions of the \mathbf{O} -notation, such as Θ for asymptotic linear dominance. ■

Note 2.44 (Different viewpoints). It is equivalent to define either the functions $\{\mathbf{O}_X : X \in U\}$, or the preorders $\{\preceq_X : X \in U\}$; one can be recovered from the other. We shall give the theorems in terms of $\{\mathbf{O}_X : X \in U\}$, since this is more familiar to computer scientists. However, we find that intuition works better when working with $\{\preceq_X : X \in U\}$. ■

Note 2.45 (Related notations). The related notations are given in terms of the dominance relation as:

$$\begin{aligned} \mathbf{o}_X(f) &= \{g \in R_X : g \prec_X f\}, \\ \Omega_X(f) &= \{g \in R_X : g \succeq_X f\}, \\ \omega_X(f) &= \{g \in R_X : g \succ_X f\}, \\ \Theta_X(f) &= \{g \in R_X : g \approx_X f\}. \end{aligned} \tag{2.46}$$

The \mathbf{O} -notation extends naturally to sets of functions; such generality is sometimes needed.

Definition 2.47 (\mathbf{O} -notation). The \mathbf{O} -notation on a set X is a function $\mathbf{O}_X : \mathcal{P}(R_X) \rightarrow \mathcal{P}(R_X)$ such that

$$\mathbf{O}_X(A) = \bigcup_{f \in A} \mathbf{O}_X(f). \tag{2.48}$$

2.6 Implicit conventions

An *implicit convention* is an overload of notation adopted by people working in a given field. Since it is an overload, the reader is required to deduce the correct meaning of such notation from the context. Here are some implicit conventions related to the \mathbf{O} -notation — as commonly used in computer science.

Placeholder convention

Definition 2.49 (Placeholder convention). The *placeholder convention* is to use $A \subset \mathcal{P}(R_X)$ as a placeholder for an anonymous function $g \in \mathbf{O}_X(A)$. It then must be guessed from the context whether the author means by A the actual set, or the anonymous function g .

Example 2.50 (Cost of an algorithm). An algorithm costs $\mathbf{O}_{\mathbb{N}}(n^2)$ if its cost function is an element of $\mathbf{O}_{\mathbb{N}}(\mathbf{O}_{\mathbb{N}}(n^2)) = \mathbf{O}_{\mathbb{N}}(n^2)$. \triangle

Example 2.51 (Exponential of an O-set). Consider the statement that an algorithm costs $2^{\mathbf{O}_{\mathbb{N}}(n)}$. As a set, $0.5 \notin 2^{\mathbf{O}_{\mathbb{N}}(n)}$. However,

$$0.5 \in \mathbf{O}_{\mathbb{N}}\left(2^{\mathbf{O}_{\mathbb{N}}(n)}\right), \quad (2.52)$$

since $0 \in \mathbf{O}_{\mathbb{N}}(n)$ and $0.5 \in \mathbf{O}_{\mathbb{N}}(2^0)$. Similarly,

$$2^{2.5n} \in \mathbf{O}_{\mathbb{N}}\left(2^{\mathbf{O}_{\mathbb{N}}(n)}\right), \quad (2.53)$$

since $2.5n \in \mathbf{O}_{\mathbb{N}}(n)$. \triangle

Domain convention

Definition 2.54 (Domain convention). The *domain convention* is to leave off the domain of the O-notation, say $\mathbf{O}(n^2)$, and then let the reader guess, for each use, the domain from the context. Sometimes the domain convention leads to a difficult interpretation.

Example 2.55 (Difficult interpretation). Consider an algorithm [20] under the unit-cost w -bit RAM model, where $w \in \mathbb{N}$, which for $I \subset [0, 2^w)_{\mathbb{N}} = U$ finds a nearest neighbor of $i \in U$ in I in time

$$\mathbf{O}(\log_2(\log_2(\Delta + 4))), \quad (2.56)$$

Chapter 2. Preliminaries

where $\Delta \in \mathbb{N}$ is the distance between i and its nearest neighbor in I . Intuitively, this sounds reasonable, but what is the domain? Our thinking process went as follows.

Since the expression contains only a single symbol Δ , we assumed it to be a univariate O -notation. Our first guess was $O_U(\log_2(\log_2(\Delta + 4)))$ — with w fixed. However, since U is bounded, this is equal to $O_U(1)$. The guess had to be wrong; otherwise the authors would have reported the complexity as $O(1)$.

Our second guess was $O_N(\log_2(\log_2(\Delta + 4)))$ — again with w fixed. However, this arbitrarily extends the complexity analysis to elements outside the input-set, since $\Delta < 2^w$. In addition, it is not always possible to do such an extension, such as when the function is $\log_2(\log_2(2^w - \Delta + 4))$ instead.

Finally, we observed that the complexity depends both on w and Δ — although $O(\log_2(\log_2(\Delta + 4)))$ never mentions w . The correct formalization is given by $O_D(\log_2(\log_2(\Delta + 4)))$, where $D = \{(w, \Delta) \in \mathbb{N}^2 : \Delta \in [0, 2^w)\}$. The corresponding algorithm would then take as input

$$\begin{aligned} w &\in \mathbb{N}^{>0}, \\ I &\in \cup_{k \in \mathbb{N}^{>0}} \mathcal{P}([0, 2^k)_{\mathbb{N}}), \\ i &\in \cup_{k \in \mathbb{N}^{>0}} [0, 2^k), \end{aligned} \tag{2.57}$$

subject to $I \subset [0, 2^w)_{\mathbb{N}}$ and $i \in [0, 2^w)$. \triangle

2.7 Worst case, best case, average case

In this section we will formalize the concepts of worst-case, best-case, and average-case analyses.

Definition 2.58 (Grouping). A *grouping* of X is a function $g: X \rightarrow Z$.

Definition 2.59 (Case). A *case* over a grouping $g: X \rightarrow Z$ is a function $s: \overrightarrow{g}(X) \rightarrow X$ such that $g \circ s = \text{id}_Z$; a right inverse of g .

Chapter 2. Preliminaries

Definition 2.60 (Worst case). A case $s: \vec{g}(X) \rightarrow X$ over a grouping $g: X \rightarrow Z$ is called *worst* of $f \in R_X$, if

$$(f \circ s)(z) = \sup \vec{f}(\overleftarrow{g}(\{z\})), \quad (2.61)$$

for all $z \in \vec{g}(X)$.

Note 2.62. A worst case may not exist. ■

Definition 2.63 (Worst-case analysis). A *worst-case analysis* of $f \in R_X$ over a grouping $g: X \rightarrow Z$ is the process of finding out

$$\sup \vec{f}(\overleftarrow{g}(\{z\})) \quad (2.64)$$

or some O-set which contains it.

Definition 2.65 (Best case). A case $s: \vec{g}(X) \rightarrow X$ over a grouping $g: X \rightarrow Z$ is called *best* of $f \in R_X$ if

$$(f \circ s)(z) = \inf \vec{f}(\overleftarrow{g}(\{z\})), \quad (2.66)$$

for all $z \in \vec{g}(X)$.

Note 2.67. A best case may not exist. ■

Definition 2.68 (Best-case analysis). A *best-case analysis* of $f \in R_X$ over a grouping $g: X \rightarrow Z$ is the process of finding out

$$\inf \vec{f}(\overleftarrow{g}(\{z\})) \quad (2.69)$$

or some Ω -set which contains it.

Example 2.70 (Analysis of insertion sort). Assume the RAM model, with unit cost for comparison of integers and zero cost for other atomic operations. Let $\mathbb{N}^* = \bigcup_{d \in \mathbb{N}} \mathbb{N}^d$ be the set of all finite sequences over \mathbb{N} . Let $F: \mathbb{N}^* \rightarrow \mathbb{N}^*$ be the insertion sort algorithm [2], which sorts a given input sequence x into increasing order. Let $f \in R_{\mathbb{N}^*}$ be the number of comparisons made by F . Let $g: \mathbb{N}^* \rightarrow \mathbb{N}$ be such that $g(x) = |x|$, the length of the sequence x . Let $s: \mathbb{N} \rightarrow \mathbb{N}^*$ be the worst case of f over g ; each such sequence is decreasing. Then the worst-case complexity of f over g is $f \circ s$, and the worst-case analysis of f provides $O_{\mathbb{N}}(f \circ s) = O_{\mathbb{N}}(n^2 + 1)$. Let $r: \mathbb{N} \rightarrow \mathbb{N}^*$ be the best case of f over g ; each such sequence is increasing. Then the best-case complexity of f over g is $f \circ r$, and the best-case analysis of f provides $\Omega_{\mathbb{N}}(f \circ r) = \Omega_{\mathbb{N}}(n + 1)$. For an arbitrary case $p: \mathbb{N} \rightarrow \mathbb{N}^*$ of f over g , it holds that $(f \circ p) \in \Omega_{\mathbb{N}}(n + 1) \cap O_{\mathbb{N}}(n^2 + 1)$. \triangle

Definition 2.71 (Average-case analysis). Let $(X, \Sigma_X, \mathbb{P})$ be a probability space, and (Z, Σ_Z) be a measurable space. Let $g: X \rightarrow Z$ be a random element, and $f \in R_X$ be a random variable. An *average-case analysis of f over g* is the process of finding out $O_Z(\mathbb{E}[f \mid g] \circ s)$, where $s: \overrightarrow{g}(X) \rightarrow X$ is any case over g , and \mathbb{E} stands for (conditional) expectation.

Since worst-case, best-case, and average-case analyses are the most common forms of complexity analysis in computer science — with the grouping set almost always $Z \subset \mathbb{N}^d$, for some $d \in \mathbb{N}^{>0}$ — this has led to the often repeated claim that the result of complexity analysis is a function which maps an ‘input size’ to the amount of used resources. For example, [2, page 25] writes as follows (emphasis theirs):

The best notion for *input size* depends on the problem being studied. For many problems, such as sorting or computing discrete Fourier transforms, the most natural measure is the *number of items in the input* - for example, the array size n for sorting. For many other problems, such as multiplying two integers, the best measure of input size is the *total number of bits* needed to represent the input in ordinary binary notation. Sometimes, it is more appropriate to describe the size of the input with two numbers rather than one. For instance, if the input to an algorithm is a graph, the input size can be described by the numbers of vertices and edges in the graph.

Chapter 2. Preliminaries

We shall indicate which input size measure is being used with each problem we study.

Complexity theorists sometimes study the cost functions of Turing machines with respect to input-tape-size. However, this is not a formalization of the input-size as described in the above quotation.

The term input-size — as it has been used — is a synonym for a group label. A group label need not have any properties, such as an order. A grouping need only be done if the analysis or its interpretation otherwise seems difficult. It would seem clearer to use the term input-size only in those cases where the group labels form a set which is linearly ordered and contains a least element (e.g. $\mathbb{N}^{\geq 0}$, $\mathbb{R}^{\geq 0}$, or cardinal numbers).

We have shown above how input-size-thinking is subsumed by the more general input-set-thinking. In the input-set thinking, a set is used to provide a mathematical model for a data structure, and a cost function is a function of this data.

2.8 Misuses

The O-notation (and related definitions) is sometimes misused even by experienced researchers in computer science. By a *misuse* we mean to use the notation in a context which does not have a formal meaning — even after applying the implicit conventions specific to computer science.

In the following we review some misuses made by experienced researchers in computer science. Let us note that, despite the misuses, the sources we refer to here are both great reading: [2] is a classic book about algorithms, data-structures, and their analysis, while [21] provides an ingenious scheme for converting a static dictionary to a linear-space dynamic dictionary.

Example 2.72 (Recurrence equations). Reference [2, page 102] studies the solutions to the recurrence equation

$$T(n) = \begin{cases} aT(n/b) + F(n), & n > 1, \\ \Theta(1), & n = 1, \end{cases} \quad (2.73)$$

where $a \in \mathbb{R}^{\geq 1}$, $b \in \mathbb{R}^{> 1}$, $B = \{b^i : i \in \mathbb{N}\}$, and $T, F \in R_B$. The intent here is to study the Master theorem over powers — as we do in Appendix F.1. Unfortunately, since $\Theta(1) := \Theta_X(1) \subset R_X$ is a set of functions, and not a

Chapter 2. Preliminaries

non-negative real number, this equation does not have a formal meaning. In addition, the set X is left undefined.

Applying the placeholder convention leads to

$$T(n) = \begin{cases} aT(n/b) + F(n), & n > 1, \\ \widehat{1}, & n = 1, \end{cases} \quad (2.74)$$

where $\widehat{1} \in \Theta_X(1)$. This still does not make sense; $\widehat{1}$ is a function, not a non-negative real number.

The motivation for this recurrence equation is the analysis of divide-and-conquer algorithms, where the algorithm recursively solves b – perhaps overlapping — sub-problems (when b is an integer), and then combines their results to solve the original problem. The call-graph of such an algorithm is a tree, and in each leaf of this tree we would perhaps like to assign a *different* constant for the amount of resources it takes. The problem is that the function T can have only one value for $T(1)$.

One way to fix the recurrence equation is to simplify it to

$$T(n) = \begin{cases} aT(n/b) + F(n), & n > 1, \\ d, & n = 1, \end{cases} \quad (2.75)$$

where $d \in \mathbb{R}^{>0}$, which is the form we study in Appendix F. We will show that $\Theta_B(T)$ is independent of the choice of d . Therefore, the simplified recurrence equation provides a solution for the divide-and-conquer analysis even when the costs in the leaf nodes vary in a fixed closed interval.^a We believe this is the idea that [2] was aiming for; it just is not captured by replacing d with $\Theta(1)$. \triangle

^aAssuming the interval does not contain zero.

Example 2.76 (Conditional statements). Let $k \in \mathbb{N}^{>0}$, and $S, T \in R_{\mathbb{N}^{>0}}$. In [21, page 9], there is the following statement:

Then for $n = \omega(1)$, we have

$$n \geq O\left(n^{1/k}\right) \geq O\left(n^{(1-1/k)/k}\right),$$

Chapter 2. Preliminaries

and

$$n^{1-1/k} \geq O\left(n^{(1-1/k)^2}\right).$$

For $n = O(1)$, we trivially have

$$T(n) = O(1) = O(S(n)).$$

We decode this as follows:

Then for $n \in \omega_{\mathbb{N}>0}(1)$, we have

$$O_{\mathbb{N}>0}(n) \supset O_{\mathbb{N}>0}\left(n^{1/k}\right) \supset O_{\mathbb{N}>0}\left(n^{(1-1/k)/k}\right),$$

and

$$O_{\mathbb{N}>0}\left(n^{1-1/k}\right) \supset O_{\mathbb{N}>0}\left(n^{(1-1/k)^2}\right).$$

For $n \in O_{\mathbb{N}>0}(1)$, we trivially have

$$T \in O_{\mathbb{N}>0}(1) = O_{\mathbb{N}>0}(S).$$

The expression $n \in \omega_{\mathbb{N}>0}(1)$ is always true for Θ , Θ , Θ and Θ . Similarly, the expression $n \in O_{\mathbb{N}>0}(1)$ is always false. The authors have an intuitive concept which they want to transmit. However, the formalization of the intuition is incorrect, and so the communication fails. \triangle

2.9 Completeness

In this section we study the completeness of the dominance relation \preceq_X .

Definition 2.77 (Completeness over a family). A preorder $\preceq_X : R_X \leftrightarrow R_X$ is *complete* over $A \subset \mathcal{P}(R_X)$, if every $F \in A$ which has a lower-bound (an upper-bound) in R_X has a greatest lower-bound (a least upper-bound) in R_X .

Note 2.78. Different types of completeness are listed in Table 2.1. \blacksquare

Chapter 2. Preliminaries

Concept	Complete over...
Complete	$\mathcal{P}(R_X)$
Directed-complete	directed subsets of R_X
Chain-complete	linearly-ordered subsets of R_X
Lattice	$\widehat{\mathcal{P}}(R_X)$
Algorithm-complete	$\{\{f_F : F \in \mathcal{A}(P)\} : P \in (X \rightarrow P(X))\}$

Table 2.1: Different types of completeness.

Proposition 2.79 (Lattice structure is implied). \preceq_X has order-consistency, locality, and injective sub-composability. $\implies \preceq_X$ has lattice structure.

Proof. Let $f_1, \dots, f_n \in R_X$. Then $f_1, \dots, f_n \leq \max(f_1, \dots, f_n)$, and by order-consistency $f_1, \dots, f_n \preceq_X \max(f_1, \dots, f_n)$. Suppose $h \in R_X$ is such that $f_1, \dots, f_n \preceq_X h \preceq_X \max(f_1, \dots, f_n)$. Let

$$F_i = \{x \in X : f_i(x) = \max(f_1, \dots, f_n)\}. \quad (2.80)$$

By injective sub-composability, $(f_i|_{F_i}) \preceq_{F_i} (h|_{F_i}) \preceq_{F_i} (f_i|_{F_i})$. Therefore $(f_i|_{F_i}) \approx_{F_i} (h|_{F_i})$. By locality, $\max(f_1, \dots, f_n) \approx_X h$. Therefore

$$\sup\{f_1, \dots, f_n\} \approx_X \max(f_1, \dots, f_n). \quad (2.81)$$

Similarly, $\inf\{f_1, \dots, f_n\} \approx_X \min(f_1, \dots, f_n)$. □

Note 2.82 (Equivalence between types of completeness). When \preceq_X is a lattice, every subset of R_X is directed, and therefore directed-complete is equivalent to complete. Further, chain-complete is equivalent to directed-complete by Zorn's lemma.^a ■

^aZorn's lemma is equivalent to the axiom of choice.

Theorem 2.83 (Incompleteness). $\preceq_{\mathbb{N}^{>0}}$ has order-consistency, transitivity, sub-homogeneity in \mathbb{N} , sub-homogeneity in $1/\mathbb{N}^{>0}$, scale-invariance, and one-separation. $\implies \preceq_{\mathbb{N}^{>0}}$ is not complete.

Chapter 2. Preliminaries

Proof. $\preceq_{\mathbb{N}^{>0}}$ has sub-homogeneity by [Sub-homogeneity is implied] (C.4).

Let $f_\alpha \in R_{\mathbb{N}^{>0}}$ be such that $f_\alpha(n) = \alpha^n$, for all $\alpha \in \mathbb{R}^{\geq 0}$. By order-consistency, $\alpha \leq \beta \implies f_\alpha \preceq_{\mathbb{N}^{>0}} f_\beta$, for all $\alpha, \beta \in \mathbb{R}^{\geq 0}$.

Suppose there exists $\alpha, \beta \in \mathbb{R}^{\geq 0}$ such that $\alpha < \beta$ and $f_\alpha \succeq_{\mathbb{N}^{>0}} f_\beta$. By sub-homogeneity, $(f_\beta/f_\alpha) \preceq_{\mathbb{N}^{>0}} 1$. It can be shown that $n \leq \frac{1}{e \ln(\beta/\alpha)} (f_\beta/f_\alpha)$. By order-consistency and scale-invariance, $n \preceq_{\mathbb{N}^{>0}} (f_\beta/f_\alpha)$. By transitivity, $n \preceq_{\mathbb{N}^{>0}} 1$, which contradicts one-separation. Therefore, $\alpha < \beta \implies f_\alpha \prec_{\mathbb{N}^{>0}} f_\beta$, for all $\alpha, \beta \in \mathbb{R}^{\geq 0}$.

Let $F = \{f_\alpha : \alpha \in \mathbb{R}^{>2}\}$. Then f_2 is a lower-bound of F . Let $\underline{f} \in R_{\mathbb{N}^{>0}}$ be a lower-bound of F such that $f_2 \preceq_{\mathbb{N}^{>0}} \underline{f}$.

Suppose $\underline{f} \approx_{\mathbb{N}^{>0}} \beta^n$, for some $\beta \in \mathbb{R}^{>2}$. Then $\left(\frac{2+\beta}{2}\right)^n \prec_{\mathbb{N}^{>0}} \beta^n \approx_{\mathbb{N}^{>0}} \underline{f}$, which contradicts \underline{f} being a lower-bound of F . Therefore $\underline{f} \prec_{\mathbb{N}^{>0}} \alpha^n$, for all $\alpha \in \mathbb{R}^{>2}$.

By order-consistency, $\underline{f} \preceq_{\mathbb{N}^{>0}} n\underline{f}$. Suppose $n\underline{f} \preceq_{\mathbb{N}^{>0}} \underline{f}$. By sub-homogeneity, $n \preceq_{\mathbb{N}^{>0}} 1$, which contradicts one-separation. Therefore $\underline{f} \prec_{\mathbb{N}^{>0}} n\underline{f}$.

By sub-homogeneity, $n\underline{f} \prec_{\mathbb{N}^{>0}} n\alpha^n$, for all $\alpha \in \mathbb{R}^{>2}$. It can be shown that $n\alpha^n \leq \frac{\beta/\alpha}{\ln(\beta/\alpha)e} \beta^n$, for all $\alpha, \beta \in \mathbb{R}^{>2}$ such that $\alpha < \beta$. By order-consistency and scale-invariance, $n\alpha^n \preceq_{\mathbb{N}^{>0}} \beta^n$, for all $\alpha, \beta \in \mathbb{R}^{>2}$ such that $\alpha < \beta$. By transitivity, $n\underline{f} \prec_{\mathbb{N}^{>0}} \alpha^n$, for all $\alpha \in \mathbb{R}^{>2}$; $n\underline{f}$ is a lower-bound for F . Since $\underline{f} \prec_{\mathbb{N}^{>0}} n\underline{f}$, there is no greatest lower-bound for F .

□

Chapter 3

Desirable properties

In this section we provide an extensive list of desirable properties for an O -notation. We will show that they all hold for linear dominance in Appendix C. Each property is given for O_X , where $X \in U$. A given property holds for O if it holds for O_X , for all $X \in U$. The desirable properties are listed in Table 3.1.

Note 3.1 (A computational model for examples). When analyzing the cost functions of the example-algorithms in this section, addition costs one unit, while all other operations cost nothing. ■

3.1 Dominance properties

The *dominance properties* are those which mirror the desire for O_X to represent a down-set of a preorder, where the preorder is consistent with the partial order \leq in R_X .

Definition 3.2 (Order-consistency). O_X has *order-consistency*, if

$$f \leq g \implies f \in O_X(g), \quad (3.3)$$

for all $f, g \in R_X$.

Chapter 3. Desirable properties

Name	Property
Order-consistency	$f \leq g \implies f \in O_X(g)$
Reflexivity	$f \in O_X(f)$
Transitivity	$(f \in O_X(g) \text{ and } g \in O_X(h)) \implies f \in O_X(h)$
Orderness	$f \in O_X(g) \iff O_X(f) \subset O_X(g)$
Zero-separation	$1 \notin O_{\mathbb{N}^{>0}}(0)$
One-separation	$n \notin O_{\mathbb{N}^{>0}}(1)$
Zero-triviality	$O_X(0) = \{0\}$
Scale-invariance	$O_X(\alpha f) = O_X(f)$
Bounded translation-inv.	$O_X(f + \beta + \alpha) = O_X(f + \beta)$
Power-homogeneity	$O_X(f)^\alpha = O_X(f^\alpha)$
Additive consistency	$u O_X(f) + v O_X(f) = (u + v) O_X(f)$
Multiplicative consistency	$O_X(f)^u \cdot O_X(f)^v = O_X(f)^{u+v}$
Maximum consistency	$\max(O_X(f), O_X(f)) = O_X(f)$
Locality	$(\forall D \in C : (f D) \in O_D(g D)) \implies f \in O_X(g)$
Scalar homogeneity	$\alpha O_X(f) = O_X(\alpha f)$
Sub-homogeneity	$u O_X(f) \subset O_X(uf)$
Sub-homogeneity in $\mathbb{Q}^{\geq 0}$	$u O_X(f) \subset O_X(uf) \quad (\vec{u}(X) \subset \mathbb{Q}^{\geq 0})$
Sub-homogeneity in \mathbb{N}	$u O_X(f) \subset O_X(uf) \quad (\vec{u}(X) \subset \mathbb{N})$
Sub-hom. in $1/\mathbb{N}^{>0}$	$u O_X(f) \subset O_X(uf) \quad (\vec{u}(X) \subset 1/\mathbb{N}^{>0})$
Super-homogeneity	$u O_X(f) \supset O_X(uf)$
Sub-multiplicativity	$O_X(f) \cdot O_X(g) \subset O_X(fg)$
Super-multiplicativity	$O_X(f) \cdot O_X(g) \supset O_X(fg)$
Sub-restrictability	$(O_X(f) D) \subset O_D(f D)$
Super-restrictability	$(O_X(f) D) \supset O_D(f D)$
Additivity	$O_X(f) + O_X(g) = O_X(f + g)$
Summation rule	$O_X(f + g) = O_X(\max(f, g))$
Maximum rule	$\max(O_X(f), O_X(g)) = O_X(\max(f, g))$
Maximum-sum rule	$\max(O_X(f), O_X(g)) = O_X(f) + O_X(g)$
Sub-composability	$O_X(f) \circ s \subset O_Y(f \circ s)$
Injective super-comp.	$O_X(f) \circ s \supset O_Y(f \circ s) \quad (s \text{ injective})$
Extensibility	$O_X(f) \circ [\underline{X} \times Y] \subset O_{X \times Y}(f \circ [\underline{X} \times Y])$
Subset-summability	$O_X \left(\sum_{(y,z) \in S_x} a(z) h(y) \right) \subset$ $O_X \left(\sum_{(y,z) \in S_x} a(z) \bar{h}(y) \right)$

Table 3.1: Desirable properties for an O-notation. Here $X, Y, Z \in U$, $f, g, u, v \in R_X$, $\bar{h} \in R_Y$, $h \in O_Y(\bar{h})$, $\alpha, \beta \in \mathbb{R}^{>0}$, $D \subset X$, $s: Y \rightarrow X$, $S: X \rightarrow \widehat{\mathcal{P}}(Y \times Z)$, $a \in R_Z$, and $C \subset \mathcal{P}(X)$ is a finite cover of X . Primitive properties marked with a bold face.

Chapter 3. Desirable properties

Example 3.4. Let $X \in U$ and $f \in R_X$. By order-consistency, $0 \in O_X(f)$.
 \triangle

Example 3.5 (Powers after 1). Suppose $\mathbb{R} \in U$. Let $\alpha, \beta \in \mathbb{R}$ be such that $\alpha \leq \beta$. Then

$$x^\alpha \leq x^\beta, \quad (3.6)$$

for all $x \in \mathbb{R}^{\geq 1}$. By order-consistency,

$$x^\alpha \in O_{\mathbb{R}^{\geq 1}}(x^\beta). \quad (3.7)$$

\triangle

Example 3.8 (Powers before 1). Suppose $\mathbb{R} \in U$. Let $\alpha, \beta \in \mathbb{R}$ be such that $\alpha \leq \beta$. Then

$$x^\beta \leq x^\alpha, \quad (3.9)$$

for all $x \in (0, 1] \subset \mathbb{R}$. By order-consistency,

$$x^\beta \in O_{(0,1]}(x^\alpha). \quad (3.10)$$

\triangle

Example 3.11 (Positive power dominates a logarithm). Suppose $\mathbb{R}^{\geq 1} \in U$. Let $\alpha, \gamma \in \mathbb{R}^{>0}$ and $\beta \in \mathbb{R}^{>1}$. It can be shown that

$$\log_\beta(x)^\gamma \leq \left(\frac{\gamma}{\alpha e \ln(\beta)} \right)^\gamma x^\alpha, \quad (3.12)$$

for all $x \in \mathbb{R}^{\geq 1}$. By order-consistency,

$$\log_\beta(x)^\gamma \in O_{\mathbb{R}^{\geq 1}} \left(\left(\frac{\gamma}{\alpha e \ln(\beta)} \right)^\gamma x^\alpha \right). \quad (3.13)$$

\triangle

Example 3.14 (Positive power dominates a logarithm, generalized). Let $X \in U$, $\alpha, \gamma \in \mathbb{R}^{>0}$, $\beta \in \mathbb{R}^{>1}$, and $f \in R_X$ be such that $f \geq 1$. Then

$$\log_\beta^\gamma \circ f \leq \left(\frac{\gamma}{\alpha e \ln(\beta)} \right)^\gamma f^\alpha. \quad (3.15)$$

By order-consistency,

$$\log_\beta^\gamma \circ f \in O_X \left(\left(\frac{\gamma}{\alpha e \ln(\beta)} \right)^\gamma f^\alpha \right). \quad (3.16)$$

Since this holds for all $\alpha \in \mathbb{R}^{>0}$,

$$\log_\beta^\gamma \circ f \in \bigcap_{\alpha \in \mathbb{R}^{>0}} O_X \left(\left(\frac{\gamma}{\alpha e \ln(\beta)} \right)^\gamma f^\alpha \right). \quad (3.17)$$

△

Example 3.18 (Functions on a finite set). Let $X \in U$ be finite, and $f \in R_X$. Then $f \leq \max(\vec{f}(X))$. By order-consistency,

$$f \in O_X \left(\max(\vec{f}(X)) \right). \quad (3.19)$$

△

Definition 3.20 (Reflexivity). O_X has *reflexivity*, if

$$f \in O_X(f), \quad (3.21)$$

for all $f \in R_X$.

Example 3.22. $(5n + 3) \in O_{\mathbb{N}}(5n + 3)$.

△

Definition 3.23 (Transitivity). O_X has *transitivity*, if

$$(f \in O_X(g) \text{ and } g \in O_X(h)) \implies f \in O_X(h), \quad (3.24)$$

for all $f, g, h \in R_X$.

Example 3.25. Let $g \in O_{\mathbb{N}}(n^2)$, and $f \in O_{\mathbb{N}}(g)$. By transitivity, $f \in O_{\mathbb{N}}(n^2)$. \triangle

Definition 3.26 (Orderness). O_X has *orderness*, if

$$f \in O_X(g) \iff O_X(f) \subset O_X(g), \quad (3.27)$$

for all $f, g \in R_X$.

Example 3.28. $n \in O_{\mathbb{N}}(n^2) \iff O_{\mathbb{N}}(n) \subset O_{\mathbb{N}}(n^2)$. \triangle

3.2 Non-triviality properties

The *non-triviality properties* are those which require that the O -notation be detailed enough.

Example 3.29. The class of functions $O_X: R_X \rightarrow \mathcal{P}(R_X)$, such that $O_X(f) := R_X$, for all $X \in U$, satisfies all of the desirable properties for an O -notation except those of non-triviality. \triangle

Definition 3.30 (Zero-separation). O has *zero-separation*, if

$$1 \notin O_{\mathbb{N}^{>0}}(0). \quad (3.31)$$

Definition 3.32 (One-separation). O has *one-separation*, if

$$n \notin O_{\mathbb{N}^{>0}}(1). \quad (3.33)$$

Definition 3.34 (Zero-triviality). O_X has *zero-triviality*, if

$$O_X(0) = \{0\}. \quad (3.35)$$

3.3 Abstraction properties

The *abstraction properties* are those which define the way in which the O -notation identifies functions.

Definition 3.36 (Scale-invariance). O_X has *scale-invariance*, if

$$O_X(\alpha f) = O_X(f), \quad (3.37)$$

for all $f \in R_X$, and $\alpha \in \mathbb{R}^{>0}$.

Example 3.38. $O_{\mathbb{N}}(2n) = O_{\mathbb{N}}(n)$. \triangle

Example 3.39 (Power dominates a logarithm, continued). Let $X \in U$, $\beta \in \mathbb{R}^{>1}$, $\gamma \in \mathbb{R}^{>0}$, and $f \in R_X$ be such that $f \geq 1$. Continuing Example 3.14, by scale-invariance,

$$O_X(\log_{\beta}^{\gamma} \circ f) \subset \bigcap_{\alpha \in \mathbb{R}^{>0}} O_X(f^{\alpha}). \quad (3.40)$$

\triangle

Example 3.41 (Functions on a finite set, continued). Let $X \in U$ be finite, and $f \in R_X$. By Example 3.18, $f \in O_X(\max(\vec{f}(X)))$. By scale-invariance, $f \in O_X(1)$, if $f \neq 0$, and $f \in O_X(0)$, if $f = 0$. \triangle

Definition 3.42 (Bounded translation-invariance). O_X has *bounded translation-invariance*, if

$$O_X(f + \beta + \alpha) = O_X(f + \beta), \quad (3.43)$$

for all $f \in R_X$, and $\alpha, \beta \in \mathbb{R}^{>0}$.

Example 3.44. $O_{\mathbb{N}^{>0}}(n + 1) = O_{\mathbb{N}^{>0}}(n)$. \triangle

Note 3.45 (The role of β in bounded translation-invariance). Suppose $f \in R_X$ is such that that $0 \in \vec{f}(X)$. Then β protects against transforming a zero cost to a non-zero cost.

Suppose $f \in R_X$ is such that $f > 0$ and $\inf \vec{f}(X) = 0$ — e.g., $f \in R_{\mathbb{N}^{>0}}$ such that $f(n) = 1/n^2$. Then it is possible to call the corresponding algorithm F arbitrary number of times — with different inputs — while spending a bounded amount of resources. The β protects against transforming a bounded cost function into an unbounded one. \blacksquare

Example 3.46. Suppose $\mathbb{R}^{>0}, X \in U$, and $X \neq \emptyset$. Then the following do *not* follow from bounded translation-invariance: $O_{\mathbb{N}}(n + 1) = O_{\mathbb{N}}(n)$, $O_X(1) = O_X(0)$, and $O_{\mathbb{R}^{>0}}((1/x) + 1) = O_{\mathbb{R}^{>0}}(1/x)$. \triangle

Definition 3.47 (Power-homogeneity). O_X has *power-homogeneity*, if

$$O_X(f)^\alpha = O_X(f^\alpha), \quad (3.48)$$

for all $f \in R_X$, and $\alpha \in \mathbb{R}^{>0}$.

Example 3.49. $O_{\mathbb{N}}(n)^2 = O_{\mathbb{N}}(n^2)$. \triangle

Definition 3.50 (Additive consistency). O_X has *additive consistency*, if

$$u O_X(f) + v O_X(f) = (u + v) O_X(f), \quad (3.51)$$

for all $f, u, v \in R_X$.

Example 3.52. $3 O_{\mathbb{N}}(n) + n^2 O_{\mathbb{N}}(n) = (3 + n^2) O_{\mathbb{N}}(n)$. \triangle

Definition 3.53 (Multiplicative consistency). O_X has *multiplicative consistency*, if

$$O_X(f)^u \cdot O_X(f)^v = O_X(f)^{u+v}, \quad (3.54)$$

for all $f, u, v \in R_X$.

Example 3.55. $O_{\mathbb{N}}(n)^3 O_{\mathbb{N}}(n)^{n^2} = O_{\mathbb{N}}(n)^{3+n^2}$. \triangle

Definition 3.56 (Maximum consistency). O_X has *maximum consistency*, if

$$\max(O_X(f), O_X(f)) = O_X(f), \quad (3.57)$$

for all $f \in R_X$.

3.4 Structural properties

The *structural properties* are those which mirror the structure of algorithms: repetition, conditional branching, and abstraction.

Definition 3.58 (Locality). O has *locality*, if

$$(\forall D \in C : (f|D) \in O_D(g|D)) \implies f \in O_X(g). \quad (3.59)$$

for all $X \in U$, $f, g \in R_X$, and $C \subset \mathcal{P}(X)$ a finite cover of X .

Definition 3.60 (Scalar homogeneity). O_X has *scalar homogeneity*, if

$$\alpha O_X(f) = O_X(\alpha f), \quad (3.61)$$

for all $f \in R_X$, and $\alpha \in \mathbb{R}^{>0}$.

Example 3.62. $2 O_{\mathbb{N}}(n) = O_{\mathbb{N}}(2n)$. \triangle

Definition 3.63 (Sub-homogeneity). O_X has *sub-homogeneity*, if

$$u O_X(f) \subset O_X(uf), \quad (3.64)$$

for all $f, u \in R_X$.

Definition 3.65 (Super-homogeneity). O_X has *super-homogeneity*, if

$$u O_X(f) \supset O_X(uf), \quad (3.66)$$

for all $f, u \in R_X$.

Definition 3.67 (Homogeneity). O_X has *homogeneity*, if it has sub-homogeneity and super-homogeneity.

Definition 3.68 (Sub-multiplicativity). O_X has *sub-multiplicativity*, if

$$O_X(f) \cdot O_X(g) \subset O_X(fg), \quad (3.69)$$

for all $f, g \in R_X$.

Definition 3.70 (Super-multiplicativity). O_X has *super-multiplicativity*,

ity, if

$$O_X(f) \cdot O_X(g) \supset O_X(fg), \quad (3.71)$$

for all $f, g \in R_X$.

Definition 3.72 (Multiplicativity). O_X has *multiplicativity*, if it has sub-multiplicativity and super-multiplicativity.

Algorithm 14 An algorithm to demonstrate multiplicativity.

```

1: procedure H( $x$ )
2:   for  $i \in [0, G(x)]_{\mathbb{N}}$  do
3:     F( $x$ )
4:   end for
5: end procedure

```

Example 3.73 (Demonstration of multiplicativity). Consider Algorithm 14, $H: X \rightarrow Y$. This algorithm runs the same sub-algorithm, $F: X \rightarrow Z$, repeatedly $G(x) \in \mathbb{N}$ times, where the algorithm $G: X \rightarrow \mathbb{N}$ has zero cost (i.e. does not perform additions). Let the cost functions of H and F be $h, f \in R_X$, respectively. Then

$$h = fG. \quad (3.74)$$

By multiplicativity,

$$\begin{aligned} O_X(h) &= O_X(fG) \\ &= O_X(f) \cdot O_X(G). \end{aligned} \quad (3.75)$$

That is, if we have analyzed F to have complexity $O_X(f)$, and know $O_X(G)$, then the complexity of H is given by $O_X(h) = O_X(f) \cdot O_X(G)$. \triangle

Example 3.76. $O_{\mathbb{N}}(n) \cdot O_{\mathbb{N}}(n^2) = O_{\mathbb{N}}(n^3)$. \triangle

Example 3.77. $O_{\mathbb{N}}(0) \cdot O_{\mathbb{N}}(n) = O_{\mathbb{N}}(0)$.

\triangle

Example 3.78. Suppose $\mathbb{R}^{>0} \in U$. Then $O_{\mathbb{R}^{>0}}(1/x) \cdot O_{\mathbb{R}^{>0}}(x) = O_{\mathbb{R}^{>0}}(1)$.

\triangle

Definition 3.79 (Sub-restrictability). O_X has *sub-restrictability*, if

$$(O_X(f)|D) \subset O_D(f|D), \quad (3.80)$$

for all $f \in R_X$, and $D \subset X$.

Definition 3.81 (Super-restrictability). O_X has *super-restrictability*, if

$$(O_X(f)|D) \supset O_D(f|D), \quad (3.82)$$

for all $f \in R_X$, and $D \subset X$.

Definition 3.83 (Restrictability). O_X has *restrictability*, if it has sub-restrictability and super-restrictability.

Algorithm 15 An algorithm to demonstrate restrictability.

```

1: procedure H( $x$ )
2:   return F( $x$ )
3: end procedure

```

Example 3.84 (Demonstration of restrictability). Let $X \in U$, $D \subset X$, and Y be a set. Consider Algorithm 15, $H: D \rightarrow Y$. This algorithm passes its argument — as it is — to the algorithm $F: X \rightarrow Y$. Let $h \in R_D$ and $f \in R_X$ be the cost functions of algorithms H and F , respectively. Then

$$h = (f|D). \quad (3.85)$$

Chapter 3. Desirable properties

By restrictability,

$$O_D(h) = (O_X(f)|D). \quad (3.86)$$

That is, if we have analyzed F to have the complexity $O_X(f)$, then the complexity of H is given by $O_D(h) = (O_X(f)|D)$. \triangle

Example 3.87. $(O_{\mathbb{N}}(n)|2\mathbb{N}) = O_{2\mathbb{N}}(n)$ \triangle

Example 3.88. Let $D := \{(n, n) \in \mathbb{N}^2 : n \in \mathbb{N}\}$. Then $(O_{\mathbb{N}^2}(mn)|D) = O_D(n^2)$. \triangle

Example 3.89. $(O_{\mathbb{N}^2}(mn)|\{(1, 1)\}) = O_{\{(1, 1)\}}(1)$ \triangle

Example 3.90. Suppose $\mathbb{R}^{>0} \in U$, and $g \in O_{\mathbb{R}^{>0}}(1/x)$. By sub-restrictability, $(g|\mathbb{R}^{\geq 1}) \in O_{\mathbb{R}^{\geq 1}}(1/x)$. By order-consistency, $1/x \in O_{\mathbb{R}^{\geq 1}}(1)$. By transitivity, $(g|\mathbb{R}^{\geq 1}) \in O_{\mathbb{R}^{\geq 1}}(1)$ \triangle

Definition 3.91 (Additivity). O_X has *additivity*, if

$$O_X(f) + O_X(g) = O_X(f + g), \quad (3.92)$$

for all $f, g \in R_X$.

Algorithm 16 An algorithm to demonstrate additivity.

```

1: procedure H( $x$ )
2:   return ( $F(x)$ ,  $G(x)$ )
3: end procedure

```

Example 3.93 (Demonstration of additivity). Consider Algorithm 16, $H: X \rightarrow Y \times Z$, which decomposes into two sub-algorithms $F: X \rightarrow Y$, and $G: X \rightarrow Z$, such that $H(x) = (F(x), G(x))$. Suppose the cost functions of

Chapter 3. Desirable properties

H , F , and G are $h, f, g \in R_X$, respectively. Then

$$h = f + g. \quad (3.94)$$

By additivity,

$$\begin{aligned} O_X(h) &= O_X(f + g) \\ &= O_X(f) + O_X(g). \end{aligned} \quad (3.95)$$

That is, if we have analyzed F and G to have complexities $O_X(f)$ and $O_X(g)$, respectively, then the complexity of H is given by $O_X(h) = O_X(f) + O_X(g)$. \triangle

Example 3.96. $O_{\mathbb{N}}(n) + O_{\mathbb{N}}(n^2) = O_{\mathbb{N}}(n + n^2)$. \triangle

Example 3.97. Let $X \in U$, and $f, g \in R_X$. By order-consistency, and additivity,

$$\begin{aligned} f + O_X(g) &\subset O_X(f) + O_X(g) \\ &= O_X(f + g). \end{aligned} \quad (3.98)$$

\triangle

Definition 3.99 (Summation rule). O_X has *the summation rule*, if

$$O_X(f + g) = O_X(\max(f, g)), \quad (3.100)$$

for all $f, g \in R_X$.

Note 3.101. Summation rule states that, in a finite sequence of algorithm calls, the worst cost function over all calls determines the complexity of the call-sequence. \blacksquare

Example 3.102. $O_{\mathbb{N}}(n + n^2) = O_{\mathbb{N}}(\max(n, n^2)) = O_{\mathbb{N}}(n^2)$. \triangle

Example 3.103. Suppose $\mathbb{R}^{\geq 1} \in U$. Then

$$\begin{aligned} O_{\mathbb{R}^{\geq 1}}(x + x^2) &= O_{\mathbb{R}^{\geq 1}}(\max(x, x^2)) \\ &= O_{\mathbb{R}^{\geq 1}}(x^2). \end{aligned} \tag{3.104}$$

△

Example 3.105. Suppose $\mathbb{R}^{\geq 0} \in U$. It does not follow from the summation rule that $O_{\mathbb{R}^{\geq 0}}(x + x^2) = O_{\mathbb{R}^{\geq 0}}(x^2)$. △

Definition 3.106 (Maximum rule). O_X has *the maximum rule*, if

$$\max(O_X(f), O_X(g)) = O_X(\max(f, g)), \tag{3.107}$$

for all $f, g \in R_X$.

Example 3.108. $\max(O_{\mathbb{N}}(n), O_{\mathbb{N}}(n^2)) = O_{\mathbb{N}}(\max(n, n^2)) = O_{\mathbb{N}}(n^2)$. △

Definition 3.109 (Maximum-sum rule). O_X has *the maximum-sum rule*, if

$$\max(O_X(f), O_X(g)) = O_X(f) + O_X(g), \tag{3.110}$$

for all $f, g \in R_X$.

Example 3.111. $\max(O_{\mathbb{N}}(n), O_{\mathbb{N}}(n^2)) = O_{\mathbb{N}}(n) + O_{\mathbb{N}}(n^2)$. △

Definition 3.112 (Sub-composability). O_X has *sub-composability*, if

$$O_X(f) \circ s \subset O_Y(f \circ s), \tag{3.113}$$

for all $f \in R_X$ and $s: Y \rightarrow X$.

Definition 3.114 (Injective sub-composability). O_X has *injective sub-composability*, if it has sub-composability for all injective $s: Y \rightarrow X$.

Definition 3.115 (Injective super-composability). O_X has *injective super-composability*, if

$$O_X(f) \circ s \supset O_Y(f \circ s), \quad (3.116)$$

for all $f \in R_X$ and injective $s: Y \rightarrow X$.

Note 3.117 (Super-composability does not make sense without injectivity). Let $s: Y \rightarrow X$ be a positive constant; $s(y) = c$ for all $y \in Y$, and some $c \in \mathbb{R}^{>0}$. Then $O_X(f) \circ s = \{s\}$, and $O_Y(f \circ s) = O_Y(c)$. Let $g \in O_Y(c) \setminus \{s\}$. Then $g \notin O_X(f) \circ s = \{s\}$. ■

Algorithm 17 An algorithm to demonstrate injective composability. We assume that F does not consume any resources.

```

1: procedure  $H(x)$ 
2:   return  $G(F(x))$ 
3: end procedure

```

Example 3.118 (Demonstration of injective composability). Consider Algorithm 17, $H: X \rightarrow Z$, which decomposes into two sub-algorithms $F: X \rightarrow Y$ and $G: Y \rightarrow Z$ such that F is injective and $H = G \circ F$. Denote the cost functions of H , F and G by $h, f \in R_X$ and $g \in R_Y$, respectively. Suppose $f = 0$. Then

$$\begin{aligned} h &= g \circ F + f \\ &= g \circ F. \end{aligned} \quad (3.119)$$

Chapter 3. Desirable properties

By injective composability,

$$\begin{aligned} O_X(h) &= O_X(g \circ F) \\ &= O_Y(g) \circ F. \end{aligned} \tag{3.120}$$

That is, if $f = 0$, and we know $O_Y(g)$, then the complexity of H is given by $O_X(h) = O_Y(g) \circ F$. \triangle

Definition 3.121 (Extensibility). O_X has *extensibility*, if

$$O_X(f) \circ [\underline{X} \times Y] \subset O_{X \times Y}(f \circ [\underline{X} \times Y]), \tag{3.122}$$

for all $X, Y \in U$ and $f \in R_X$.

Note 3.123. Extensibility is a special case of sub-composability. \blacksquare

Definition 3.124 (Subset-summability). O has *subset-summability*, if

$$O_X \left(\sum_{(y,z) \in S_x} a(z) f(y) \right) \subset O_X \left(\sum_{(y,z) \in S_x} a(z) \bar{f}(y) \right), \tag{3.125}$$

for all $X, Y, Z \in U$, $S: X \rightarrow \widehat{\mathcal{P}}(Y \times Z)$, $a \in R_Z$, $\bar{f} \in R_Y$, and $f \in O_Y(\bar{f})$.

Algorithm 18 An algorithm to demonstrate subset-summability.

```

1: procedure  $G(x)$ 
2:    $\alpha := 0$ 
3:   for  $(y, z) \in S_x$  do
4:      $\alpha := \alpha \cdot F(y)$ 
5:   end for
6:   return  $\alpha$ 
7: end procedure

```

Example 3.126 (Demonstration of subset-summability). Consider Algorithm 18, $G: X \rightarrow \mathbb{Z}$, which for $x \in X$ sums $F: Y \rightarrow \mathbb{Z}$ over $S_x \in \widehat{\mathcal{P}}(Y \times Z)$. Let the algorithms F and G take $f \in R_Y$ and $g \in R_X$ operations, respectively. Then

$$g(x) = \sum_{y \in S_x} f(y). \quad (3.127)$$

Suppose we have shown that $f \in O_Y(\bar{f})$. By subset-summability,

$$\begin{aligned} O_X(g) &= O_X\left(\sum_{y \in S_x} f(y)\right) \\ &\subset O_X\left(\sum_{y \in S_x} \bar{f}(y)\right). \end{aligned} \quad (3.128)$$

That is, if f has an upper bound of \bar{f} , then we can construct an upper bound of g from \bar{f} . Suppose we have also shown that $\bar{f} \in O_Y(f)$ — i.e. $f \in \Theta_Y(\bar{f})$. Then

$$O_X(g) = O_X\left(\sum_{y \in S_x} \bar{f}(y)\right). \quad (3.129)$$

△

Note 3.130 (Factor a in subset-summability). Example 3.126 does not make use of the factor a in subset-summability. Such general sums occur naturally in the analysis of recursive algorithms; see Appendix F. ■

Chapter 4

Characterization of the O-notation

In this chapter we show that the primitive properties are equivalent to the definition of O-notation as linear dominance.

4.1 Linear dominance is sufficient

In this section we will show the following theorem:

Theorem 4.1 (Linear dominance has primitive properties). *Let $\mathcal{O}_X: R_X \rightarrow \mathcal{P}(R_X)$ be defined by $g \in \mathcal{O}_X(f)$ if and only if*

$$\exists c \in \mathbb{R}^{>0} : g \leq cf, \quad (4.2)$$

for all $f, g \in R_X$, and all $X \in U$, where the universe U is the class of all sets. Then \mathcal{O} satisfies the primitive properties.

Proof. The result follows directly from the Lemmas in this section. □

We shall apply the following lemma repeatedly without mentioning it.

Lemma 4.3 (Simplification lemma). *Let $X \in U$, I be a finite set, $X_i \subset X$, $f_i \in R_{X_i}$, and $\widehat{f}_i \in \mathcal{O}_{X_i}(f_i)$, for all $i \in I$. Then there exists $c \in \mathbb{R}^{>0}$, such that $\widehat{f}_i \leq cf_i$, for all $i \in I$.*

Chapter 4. Characterization of the O-notation

Proof. Since $\widehat{f}_i \in \Theta_{X_i}(f_i)$, there exists $c_i \in \mathbb{R}^{>0}$, such that $\widehat{f}_i \leq c_i f_i$, for all $i \in I$. Let $c = \max\{c_i : i \in I\}$. Then $\widehat{f}_i \leq c f_i$, for all $i \in I$. \square

Lemma 4.4 (Linear dominance has order-consistency). *Let $X \in U$, and $f, g \in R_X$. Then*

$$f \leq g \implies f \in \Theta_X(g). \quad (4.5)$$

Proof. Since $f \leq 1g$, it holds that $f \in \Theta_X(g)$. \square

Lemma 4.6 (Linear dominance has transitivity). *Let $X \in U$, and $f, g, h \in R_X$. Then*

$$(f \in \Theta_X(g) \text{ and } g \in \Theta_X(h)) \implies f \in \Theta_X(h). \quad (4.7)$$

Proof. Let $f \in \Theta_X(g)$, and $g \in \Theta_X(h)$. Then there exists $c \in \mathbb{R}^{>0}$, such that $f \leq cg$ and $g \leq ch$. It follows that $f \leq c^2h$. Therefore $f \in \Theta_X(h)$. \square

Lemma 4.8 (Linear dominance has locality). *Let $X \in U$, $f, g \in R_X$, and $C \subset \mathcal{P}(X)$ be a finite cover of X . Then*

$$(\forall D \in C : (f|D) \in \Theta_D(g|D)) \implies f \in \Theta_X(g). \quad (4.9)$$

Proof. Assume $(f|D) \in \Theta_D(g|D)$, for all $D \in C$. Then there exist $c \in \mathbb{R}^{>0}$ such that $(f|D) \leq c(g|D)$, for all $D \in C$. Since C covers X , $f \leq cg$. Therefore $f \in \Theta_X(g)$. \square

Lemma 4.10 (Linear dominance has one-separation).

$$n \notin \Theta_{\mathbb{N}^{>0}}(1). \quad (4.11)$$

Proof. For all $c \in \mathbb{R}^{>0}$, there exists $n \in \mathbb{N}^{>0}$ — for example $n = \lceil c \rceil + 1$ — such that $n > c1$. Therefore $n \notin \Theta_{\mathbb{N}^{>0}}(1)$. \square

Lemma 4.12 (Linear dominance has scale-invariance). *Let $X \in U$, $f \in R_X$, and $\alpha \in \mathbb{R}^{>0}$. Then $\mathcal{O}_X(f) \subset \mathcal{O}_X(\alpha f)$.*

Proof. Assume $\hat{f} \in \mathcal{O}_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$, such that

$$\begin{aligned}\hat{f} &\leq cf \\ &= (c/\alpha)(\alpha f).\end{aligned}\tag{4.13}$$

Therefore $\hat{f} \in \mathcal{O}_X(\alpha f)$. □

Lemma 4.14 (Linear dominance has sub-homogeneity in \mathbb{N} and sub-homogeneity in $1/\mathbb{N}^{>0}$). *Let $X \in U$, and $f, u \in R_X$. Then*

$$u \mathcal{O}_X(f) \subset \mathcal{O}_X(uf).\tag{4.15}$$

Proof. Let $\hat{f} \in \mathcal{O}_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$, such that $\hat{f} \leq cf$. This implies $u\hat{f} \leq cuf$. Therefore $u\hat{f} \in \mathcal{O}_X(uf)$; \mathcal{O}_X has sub-homogeneity. Since $\mathbb{N} \subset \mathbb{R}^{\geq 0}$, \mathcal{O}_X has sub-homogeneity in \mathbb{N} . Since $\frac{1}{\mathbb{N}^{>0}} \subset \mathbb{R}^{>0}$, \mathcal{O}_X has sub-homogeneity in $1/\mathbb{N}^{>0}$. □

Lemma 4.16 (Linear dominance has sub-composability). *Let $X \in U$, $f \in R_X$, and $s: Y \rightarrow X$. Then*

$$\mathcal{O}_X(f) \circ s \subset \mathcal{O}_Y(f \circ s).\tag{4.17}$$

Proof. Let $\hat{f} \in \mathcal{O}_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$, such that $\hat{f} \leq cf$. This implies $\hat{f} \circ s \leq c(f \circ s)$. Therefore $\hat{f} \circ s \in \mathcal{O}_Y(f \circ s)$. □

4.2 Linear dominance is necessary

In this section we will show the following theorem.

Theorem 4.18 (Primitive properties imply linear dominance). *Suppose \mathcal{O} has order-consistency, transitivity, one-separation, locality, scale-invariance, sub-homogeneity in \mathbb{N} , sub-homogeneity in $1/\mathbb{N}^{>0}$, and sub-*

Chapter 4. Characterization of the O-notation

composability. Then

$$f \in O_X(g) \iff \exists c \in \mathbb{R}^{>0} : f \leq cg. \quad (4.19)$$

To prove this result, we will use some of the results from Appendix C.

Lemma 4.20 ($O_X(1)$ equals the bounded functions). *Suppose O has order-consistency, transitivity, one-separation, locality, scale-invariance, and injective sub-composability. Then*

$$O_X(1) = \{f \in R_X : \exists c \in \mathbb{R}^{>0} : f \leq c\}, \quad (4.21)$$

provided $X \neq \emptyset$.

Proof.

Implied properties

O has injective super-composability by [Injective super-composability is implied] (C.13), and orderness by [Orderness is implied] (C.10).

\subseteq

Assume $f \in O_X(1)$ such that f is unbounded. Then for every $n \in \mathbb{N}^{>0}$ there exists $x_n \in X$ such that $f(x_n) \geq n$. Therefore, let $s: \mathbb{N}^{>0} \rightarrow X$ be injective such that $n \leq (f \circ s)(n)$, for all $n \in \mathbb{N}^{>0}$. By orderness, $O_X(f) \subset O_X(1)$. By order-consistency, order-consistency and transitivity, injective super-composability, and injective sub-composability,

$$\begin{aligned} (n \mapsto n) &\in O_{\mathbb{N}^{>0}}(n) \\ &\subset O_{\mathbb{N}^{>0}}(f \circ s) \\ &\subset O_X(f) \circ s \\ &\subset O_X(1) \circ s \\ &\subset O_{\mathbb{N}^{>0}}(1 \circ s) \\ &= O_{\mathbb{N}^{>0}}(1). \end{aligned} \quad (4.22)$$

This contradicts O having one-separation. Therefore f is bounded, which is equivalent to $\exists c \in \mathbb{R}^{>0} : f \leq c$.

\supseteq

Chapter 4. Characterization of the O-notation

Assume $\exists c \in \mathbb{R}^{>0} : f \leq c$. By order-consistency, $f \in O_X(c)$. By scale-invariance, $f \in O_X(1)$. □

Lemma 4.23 (O-notation for positive functions). *Suppose O_X has order-consistency, transitivity, scale-invariance, sub-homogeneity in \mathbb{N} , and sub-homogeneity in $1/\mathbb{N}^{>0}$. Then*

$$f \in O_X(g) \iff f/g \in O_X(1), \quad (4.24)$$

for all $f, g \in R_X$ such that $g > 0$.

Proof. O_X has sub-homogeneity by [Sub-homogeneity is implied] (C.4).

\implies

By sub-homogeneity,

$$\begin{aligned} & f \in O_X(g) \\ \implies & f/g \in O_X(g)/g \\ \implies & f/g \in O_X(g/g) \\ \implies & f/g \in O_X(1). \end{aligned} \quad (4.25)$$

\impliedby

By sub-homogeneity,

$$\begin{aligned} & f/g \in O_X(1) \\ \implies & f \in O_X(1)g \\ \implies & f \in O_X(g). \end{aligned} \quad (4.26)$$

□

Theorem 4.27 (Primitive properties imply linear dominance). *Suppose O has order-consistency, transitivity, one-separation, locality, scale-invariance, sub-homogeneity in \mathbb{N} , sub-homogeneity in $1/\mathbb{N}^{>0}$, and sub-composability. Then*

$$f \in O_X(g) \iff \exists c \in \mathbb{R}^{>0} : f \leq cg. \quad (4.28)$$

Proof.

Chapter 4. Characterization of the O-notation

Implied properties

O_X has sub-restrictability by [Sub-restrictability is implied] (C.18), and zero-triviality by [Zero-triviality is implied] (C.11).

Positive subset

Let $G := \overleftarrow{g}(\mathbb{R}^{>0})$ and $\overline{G} := X \setminus G$. Then

$$\begin{aligned}
 & (f|G) \in O_G(g|G) \\
 \iff & \frac{(f|G)}{(g|G)} \in O_G(1) \\
 \iff & \exists c \in \mathbb{R}^{>0} : \frac{(f|G)}{(g|G)} \leq c \\
 \iff & \exists c \in \mathbb{R}^{>0} : (f|G) \leq c(g|G)
 \end{aligned} \tag{4.29}$$

where we used [O-notation for positive functions] (4.23) and [$O_X(1)$ equals the bounded functions] (4.20).

Zero subset

By zero-triviality,

$$\begin{aligned}
 & (f|\overline{G}) \in O_{\overline{G}}(g|\overline{G}) \\
 \iff & (f|\overline{G}) = 0.
 \end{aligned} \tag{4.30}$$

Whole set

By locality and sub-restrictability,

$$\begin{aligned}
 & f \in O_X(g) \\
 \iff & (f|G) \in O_G(g|G) \text{ and } (f|\overline{G}) \in O_{\overline{G}}(g|\overline{G}) \\
 \iff & (\exists c \in \mathbb{R}^{>0} : (f|G) \leq c(g|G)) \text{ and } (f|\overline{G}) = 0 \\
 \iff & \exists c \in \mathbb{R}^{>0} : f \leq cg.
 \end{aligned} \tag{4.31}$$

□

4.3 Minimal properties

In this section we will show that, excluding locality, a given primitive property can not be deduced from the remaining primitive properties.

Chapter 4. Characterization of the O-notation

	\leq	T	1	α	*	/	\circ
\ominus	\times H.21	\checkmark H.23	\checkmark H.31	\checkmark H.26	\checkmark H.32	\checkmark H.32	\checkmark H.35
\emptyset	\checkmark H.40	\times H.42	\checkmark H.49	\checkmark H.45	\checkmark H.51	\checkmark H.51	\checkmark H.55
\emptyset	\checkmark D.6	\checkmark D.10	\times G.4	\checkmark D.20	\checkmark D.23	\checkmark D.23	\checkmark G.5
\ominus	\checkmark H.3	\checkmark H.4	\checkmark H.12	\times H.7	\checkmark H.13	\checkmark H.13	\checkmark H.16
\oplus	\checkmark G.61	\checkmark G.63	\checkmark G.70	\checkmark G.72	\times G.82	\checkmark G.85	\checkmark G.76
\emptyset	\checkmark H.62	\checkmark H.64	\checkmark H.75	\checkmark H.70	\checkmark H.81	\times H.84	\checkmark H.87
\ominus	\checkmark D.6	\checkmark D.10	\checkmark G.14	\checkmark D.20	\checkmark D.23	\checkmark D.23	\times G.18
\ominus	\checkmark D.6	\checkmark D.10	\checkmark 4.10	\checkmark D.20	\checkmark D.23	\checkmark D.23	\checkmark 4.16

Table 4.1: The pre-primitive properties fulfilled by each candidate definition. The abbreviations are: \leq for order-consistency, T for transitivity, 1 for one-separation, α for scale-invariance, $*$ for sub-homogeneity in \mathbb{N} , $/$ for sub-homogeneity in $1/\mathbb{N}^{>0}$, and \circ for sub-composability

Definition 4.32 (Minimal set of axioms). A set \mathcal{S} of axioms is *minimal*, if no axiom in \mathcal{S} can be proved from the other axioms in \mathcal{S} .

Note 4.33 (Proving minimality). The minimality of an axiom set \mathcal{S} can be proved by showing that for any axiom $A \in \mathcal{S}$, there is a model of $\mathcal{S} \setminus \{A\}$ in which A holds and another model of $\mathcal{S} \setminus \{A\}$ in which A does not hold. ■

Note 4.34 (Locality is implied). Locality is implied by the other primitive properties by [Locality is implied] (C.31); primitive properties are not minimal. ■

Note 4.35 (Pre-primitive properties). The *pre-primitive properties* are the primitive properties, with locality excluded. ■

Theorem 4.36 (Pre-primitive properties are minimal). *Pre-primitive properties form a minimal set of axioms for linear dominance.*

Chapter 4. Characterization of the O-notation

Proof. That pre-primitive properties are equivalent to the definition of O-notation as linear dominance follows from [Locality is implied] (C.31), [Primitive properties imply linear dominance] (4.27), and [Linear dominance has primitive properties] (4.1).

That pre-primitive properties are minimal follows from Table 4.1, following Note 4.33. \square

Chapter 5

Working with the \mathcal{O} -notation

In this section we adopt the linear dominance \mathcal{O} -notation as *the* \mathcal{O} -notation and develop more refined tools for working with it. These tools are useful in a day-to-day basis for an algorithm analyst, because they provide shortcuts over tedious derivations. A cheat sheet for working with the \mathcal{O} -notation is given in Figure 5.1 — it is a simplified version of Figure 3.

5.1 Surprising aspects

In this section we will look at some aspects of \mathcal{O} -notation which at first may seem surprising.

Algorithm 19 A family of algorithms, parametrized by $i \in \mathbb{N}$, which take as input $n \in \mathbb{N}$, and output n .

```
1: procedure IDENTITY $i$ ( $n$ )  
2:   if  $n < i$  then  
3:     return  $n$   
4:   end if  
5:   return  $n + 0$   
6: end procedure
```

Example 5.1 (Infinite descent). Consider a family of algorithms in Algorithm 19, for which there is a separate implementation for each $i \in \mathbb{N}$.

Chapter 5. Working with the O-notation

Name	Property
Order-consistency	$f \leq g \implies f \in \Theta_X(g)$
Reflexivity	$f \in \Theta_X(f)$
Transitivity	$(f \in \Theta_X(g) \text{ and } g \in \Theta_X(h)) \implies f \in \Theta_X(h)$
Orderness	$f \in \Theta_X(g) \iff \Theta_X(f) \subset \Theta_X(g)$
Zero-separation	$1 \notin \Theta_{\mathbb{N}^{>0}}(0)$
One-separation	$n \notin \Theta_{\mathbb{N}^{>0}}(1)$
Zero-triviality	$\Theta_X(0) = \{0\}$
Scale-invariance	$\Theta_X(\alpha f) = \Theta_X(f)$
Bounded translation-inv.	$\Theta_X(f + \beta + \alpha) = \Theta_X(f + \beta)$
Power-homogeneity	$\Theta_X(f)^\alpha = \Theta_X(f^\alpha)$
Additive consistency	$u \Theta_X(f) + v \Theta_X(f) = (u + v) \Theta_X(f)$
Multiplicative consistency	$\Theta_X(f)^u \cdot \Theta_X(f)^v = \Theta_X(f)^{u+v}$
Maximum consistency	$\max(\Theta_X(f), \Theta_X(f)) = \Theta_X(f)$
Locality	$(\forall D \in C : (f D) \in \Theta_D(g D)) \implies f \in \Theta_X(g)$
Homogeneity	$u \Theta_X(f) = \Theta_X(uf)$
Multiplicativity	$\Theta_X(f) \cdot \Theta_X(g) = \Theta_X(fg)$
Restrictability	$(\Theta_X(f) D) = \Theta_D(f D)$
Additivity	$\Theta_X(f) + \Theta_X(g) = \Theta_X(f + g)$
Summation rule	$\Theta_X(f + g) = \Theta_X(\max(f, g))$
Maximum rule	$\max(\Theta_X(f), \Theta_X(g)) = \Theta_X(\max(f, g))$
Maximum-sum rule	$\max(\Theta_X(f), \Theta_X(g)) = \Theta_X(f) + \Theta_X(g)$
Sub-composability	$\Theta_X(f) \circ s \subset \Theta_Y(f \circ s)$
Injective composability	$\Theta_X(f) \circ s = \Theta_Y(f \circ s) \quad (s \text{ injective})$
Extensibility	$\Theta_X(f) \circ [\underline{X} \times Y] \subset \Theta_{X \times Y}(f \circ [\underline{X} \times Y])$
Subset-summability	$\Theta_X \left(\sum_{(y,z) \in S_x} a(z)h(y) \right) \subset \Theta_X \left(\sum_{(y,z) \in S_x} a(z)\bar{h}(y) \right)$

Table 5.1: Cheat sheet for Θ -notation. Here $X, Y, Z \in U$, $f, g, u, v \in R_X$, $\bar{h} \in R_Y$, $h \in \Theta_Y(\bar{h})$, $\alpha, \beta \in \mathbb{R}^{>0}$, $D \subset X$, $s: Y \rightarrow X$, $S: X \rightarrow \hat{\mathcal{P}}(Y \times Z)$, $a \in R_Z$, and $C \subset \mathcal{P}(X)$ is a finite cover of X .

Chapter 5. Working with the O-notation

Suppose we are only interested in the number of performed additions. The cost functions are given by $f_i \in R_{\mathbb{N}}$ such that

$$f_i(n) = \begin{cases} 1, & n \geq i, \\ 0, & \text{otherwise,} \end{cases} \quad (5.2)$$

where $i \in \mathbb{N}$. Then

$$f_0 \in \Theta_{\mathbb{N}}(1), \quad (5.3)$$

and

$$f_{i+1} \in \Theta_{\mathbb{N}}(f_i), \quad (5.4)$$

for all $i \in \mathbb{N}$. That is, with linear dominance, the functions f_i form a decreasing sequence of functions. This is how it should be: it is fundamentally different to use resources — no matter how small an amount — than to not use resources at all. \triangle

Example 5.5 (Same expressions, different functions). It may be surprising that $1/x \in \Theta_{\mathbb{N}^{>0}}(1)$, but $1/x \notin \Theta_{\mathbb{R}^{>0}}(1)$. The function $1/x$ seems to be the same — why don't they belong to the same Θ -sets?

A function is a triple, which consists of the domain, the codomain, and the rule connecting each element of the domain to exactly one element of the codomain. An expression such as $1/x$ is ambiguous as a function definition, because it does not specify the domain and the codomain. The two occurrences of $1/x$ here are different functions, and need not share any other property apart from being equal on $\mathbb{N}^{>0}$.

In our example, $1/x$ on $\mathbb{N}^{>0}$ is bounded, while $1/x$ on $\mathbb{R}^{>0}$ is unbounded. \triangle

5.2 Master theorems

Master theorems are popular for solving recurrence equations arising in the analysis of divide-and-conquer algorithms [2] — up to Θ -equivalence. We state the theorems here, and prove them in Appendix F.

Definition 5.6 (Master function over integers). Let $a \in \mathbb{R}^{\geq 1}$, $b \in \mathbb{R}^{\geq 2}$, $d \in \mathbb{R}^{>0}$, and $F \in R_{\mathbb{N}^{\geq 1}}$. A *master function over integers* is a function

Chapter 5. Working with the O-notation

$T \in R_{\mathbb{N} \geq 1}$ defined by the recurrence equation

$$T(n) = \begin{cases} aT(\lceil n/b \rceil) + F(n), & n \geq b, \\ d, & n < b. \end{cases} \quad (5.7)$$

The set of such functions is denoted by $\mathcal{M}_I(a, b, d, F)$.

Theorem 5.8 (Master theorem over integers). *Let $T \in \mathcal{M}_I(a, b, d, F)$ be a Master function over integers, and $F \in \Theta_{\mathbb{N} \geq 1}(n^c)$, where $c \in \mathbb{R}^{\geq 0}$. Then*

$$\begin{aligned} \log_b(a) < c &\implies T \in \Theta_{\mathbb{N} \geq 1}(n^c), \\ \log_b(a) = c &\implies T \in \Theta_{\mathbb{N} \geq 1}(n^c \log_b(bn)), \\ \log_b(a) > c &\implies T \in \Theta_{\mathbb{N} \geq 1}\left(n^{\log_b(a)}\right). \end{aligned} \quad (5.9)$$

If $F \in \Theta_{\mathbb{N} \geq 1}(n^c)$, then each $\Theta_{\mathbb{N} \geq 1}$ can be replaced with $\Theta_{\mathbb{N} \geq 1}$.

Definition 5.10 (Master function over reals). Let $a \in \mathbb{R}^{\geq 1}$, $b \in \mathbb{R}^{> 1}$, $d \in \mathbb{R}^{> 0}$, and $f \in R_{\mathbb{R} \geq 1}$. A *master function over reals* is a function $t \in R_{\mathbb{R} \geq 1}$ defined by the recurrence equation

$$t(x) = \begin{cases} at(x/b) + f(x), & x \geq b, \\ d, & x < b. \end{cases} \quad (5.11)$$

The set of such functions is denoted by $\mathcal{M}_R(a, b, d, f)$.

Theorem 5.12 (Master theorem over reals). *Let $t \in \mathcal{M}_R(a, b, d, f)$ be a Master function over reals, and $f \in \Theta_{\mathbb{R} \geq 1}(x^c)$, where $c \in \mathbb{R}^{\geq 0}$. Then*

$$\begin{aligned} \log_b(a) < c &\implies t \in \Theta_{\mathbb{R} \geq 1}(x^c), \\ \log_b(a) = c &\implies t \in \Theta_{\mathbb{R} \geq 1}(x^c \log_b(bx)), \\ \log_b(a) > c &\implies t \in \Theta_{\mathbb{R} \geq 1}\left(x^{\log_b(a)}\right). \end{aligned} \quad (5.13)$$

If $f \in \Theta_{\mathbb{R} \geq 1}(x^c)$, then each $\Theta_{\mathbb{R} \geq 1}$ can be replaced with $\Theta_{\mathbb{R} \geq 1}$.

5.3 \mathcal{O} -mappings

Definition 5.14 (\mathcal{O} -mapping). A function $T: R_X \rightarrow R_Y$ is an \mathcal{O} -mapping, if

$$\vec{T}(\mathcal{O}_X(f)) \subset \mathcal{O}_Y(T(f)), \quad (5.15)$$

for all $f \in R_X$.

Theorem 5.16 (\mathcal{O} -mapping by linear dominance). Let $T: R_X \rightarrow R_Y$. Then T is an \mathcal{O} -mapping if and only if

$$(\exists c \in \mathbb{R}^{>0} : f \leq cg) \implies (\exists d \in \mathbb{R}^{>0} : T(f) \leq dT(g)) \quad (5.17)$$

for all $f, g \in R_X$.

Proof. By definition. □

Example 5.18 (Non-negative translation is an \mathcal{O} -mapping). Let $T: R_X \rightarrow R_X$ be such that $T(f) = f + \alpha$, where $\alpha \in \mathbb{R}^{\geq 0}$. Let $f, g \in R_X$ be such that $f \in \mathcal{O}_X(g)$. Then there exists $c \in \mathbb{R}^{>0}$ such that $f \leq cg$, and

$$\begin{aligned} T(f) &= f + \alpha \\ &\leq cg + \alpha \\ &\leq \max(c, 1)(g + \alpha) \\ &= \max(c, 1)T(g). \end{aligned} \quad (5.19)$$

Therefore T is an \mathcal{O} -mapping. The inverse of T does not exist, since it does not always hold that $f - \alpha \geq 0$. △

Example 5.20 (Composition is an \mathcal{O} -mapping). Let $T: R_X \rightarrow R_Y$ be such that $T(f) = f \circ s$, where $s: Y \rightarrow X$. Let $f, g \in R_X$ be such that

Chapter 5. Working with the \mathcal{O} -notation

$f \in \mathcal{O}_X(g)$. Then there exists $c \in \mathbb{R}^{>0}$ such that $f \leq cg$, and

$$\begin{aligned} T(f) &= f \circ s \\ &\leq (cg) \circ s \\ &= c(g \circ s) \\ &= cT(g). \end{aligned} \tag{5.21}$$

Therefore T is an \mathcal{O} -mapping. \triangle

Theorem 5.22 (Subset-sum is an \mathcal{O} -mapping). *Let $X, Y, Z \in U$, $S: X \rightarrow \widehat{\mathcal{P}}(Y \times Z)$, $a \in R_Z$, $f \in R_Y$, and $\widehat{f} \in \mathcal{O}_X(f)$. Let $T: R_Y \rightarrow R_X$ be such that*

$$T(f) = \left(x \mapsto \sum_{(y,z) \in S_x} a(z)f(y) \right). \tag{5.23}$$

Then T is an \mathcal{O} -mapping.

Proof. There exists $c \in \mathbb{R}^{>0}$ such that $\widehat{f} \leq cf$, and so

$$\begin{aligned} T(\widehat{f}) &= \left(x \mapsto \sum_{(y,z) \in S_x} a(z)\widehat{f}(y) \right) \\ &\leq \left(x \mapsto \sum_{(y,z) \in S_x} a(z)(cf)(y) \right) \\ &= c \left(x \mapsto \sum_{(y,z) \in S_x} a(z)f(y) \right) \\ &= cT(f). \end{aligned} \tag{5.24}$$

Therefore T is an \mathcal{O} -mapping. \square

5.4 \mathcal{O} -equalities

Definition 5.25 (\mathcal{O} -residual). A function $\widehat{T}: R_Y \rightarrow R_X$ is an \mathcal{O} -residual

of $T: R_X \rightarrow R_Y$, if

$$g \in \mathcal{O}_Y(T(f)) \iff \widehat{T}(g) \in \mathcal{O}_X(f), \quad (5.26)$$

for all $f \in R_X$, and $g \in R_Y$.

Definition 5.27 (\mathcal{O} -residuated function). A function $f: X \rightarrow Y$ is *\mathcal{O} -residuated*, if it has an \mathcal{O} -residual.

Note 5.28. The definitions given here are special cases of the theory of partitioned sets — given in Appendix I — and of the theory of preordered sets — given in Appendix J, formulated in terms of \mathcal{O} -sets. ■

Definition 5.29 (Strong \mathcal{O} -equality). A *strong \mathcal{O} -equality* is a surjective \mathcal{O} -residuated function $T: R_X \rightarrow R_Y$.

Theorem 5.30 (Strong \mathcal{O} -equality rule). Let $T: R_X \rightarrow R_Y$ be a strong \mathcal{O} -equality. Then

$$T(\mathcal{O}_X(f)) = \mathcal{O}_Y(T(f)), \quad (5.31)$$

for all $f \in R_X$.

Proof. This is proved in [Transpose-residuated surjective function preserves down-sets] (J.97). □

Note 5.32 (\mathcal{O} -equality). We call the \mathcal{O} -equality strong, because we do not know how to neatly characterize the preservation of the \mathcal{O} -set under a mapping T ; such functions would be called \mathcal{O} -equalities. ■

Theorem 5.33 (Strong \mathcal{O} -equality by linear dominance). Suppose $T: R_X \rightarrow R_Y$. Then T is a strong \mathcal{O} -equality if and only if there exists $\widehat{T}: R_Y \rightarrow R_X$ such that

$$T(\widehat{T}(g)) = g, \quad (5.34)$$

and

$$(\exists c \in \mathbb{R}^{>0} : g \leq cT(f)) \iff (\exists d \in \mathbb{R}^{>0} : \widehat{T}(g) \leq df), \quad (5.35)$$

for all $f \in R_X$, and $g \in R_Y$.

Proof. By definition of linear dominance and [Partition-surjectivity is equivalent to having a right p-inverse] (I.41), whose special case says that surjectivity is equivalent to having a right-inverse. \square

Example 5.36 (Injective composition is a strong Θ -equality). Let $s: Y \rightarrow X$ be injective, and $T: R_X \rightarrow R_Y$ be such that $T(f) = f \circ s$. Let $\widehat{s}: Y \rightarrow \vec{s}(Y)$ be such that $\widehat{s}(y) = s(y)$. Then \widehat{s} is bijective, and $T(f) = f \circ \widehat{s}$. Let $\widehat{T}: R_Y \rightarrow R_X$ be such that $\widehat{T}(g) = \langle g \circ \widehat{s}^{-1} \rangle$, where $\langle \cdot \rangle$ is a domain extension of a function from $\vec{s}(Y)$ to X by mapping the new elements to zeros. Let $f \in R_X$, $g \in R_Y$, and $c \in \mathbb{R}^{>0}$. Then $T(\widehat{T}(g)) = g$, and

$$\begin{aligned} g &\leq cT(f) \\ \iff g &\leq c(f \circ \widehat{s}) \\ \iff \langle g \circ \widehat{s}^{-1} \rangle &\leq cf \\ \iff \widehat{T}(g) &\leq cf. \end{aligned} \quad (5.37)$$

Therefore T is a strong Θ -equality. \triangle

Example 5.38 (Positive power is a strong Θ -equality). Let $T: R_X \rightarrow R_X$ be such that $T(f) = f^\alpha$, where $\alpha \in \mathbb{R}^{>0}$. Let $\widehat{T}: R_X \rightarrow R_X$ be such that $\widehat{T}(g) = g^{1/\alpha}$. Let $f, g \in R_X$, and $c \in \mathbb{R}^{>0}$. Then $T(\widehat{T}(g)) = g$, and

$$\begin{aligned} g &\leq cT(f) \\ \iff g &\leq cf^\alpha \\ \iff g^{1/\alpha} &\leq c^{1/\alpha} f \\ \iff \widehat{T}(g) &\leq c^{1/\alpha} f. \end{aligned} \quad (5.39)$$

Therefore T is a strong Θ -equality. \triangle

Example 5.40 (Positive multiplication is a strong \mathcal{O} -equality). Let $T: R_X \rightarrow R_X$ be such that $T(f) = \alpha f$, where $\alpha \in \mathbb{R}^{>0}$. Let $\widehat{T}: R_X \rightarrow R_X$ be such that $\widehat{T}(g) = g/\alpha$. Let $f, g \in R_X$, and $c \in \mathbb{R}^{>0}$. Then $T(\widehat{T}(g)) = g$, and

$$\begin{aligned} & g \leq cT(f) \\ \iff & g \leq c\alpha f \\ \iff & g/\alpha \leq cf \\ \iff & \widehat{T}(g) \leq cf. \end{aligned} \tag{5.41}$$

Therefore T is a strong \mathcal{O} -equality. \triangle

Chapter 6

Local linear dominance

In this section we will study a class of candidate definitions for the O-notation, the *local linear dominances*. These definitions work over all universes — unless a specific version makes additional assumptions.

6.1 Definition

In this section we provide the definition of local linear dominance. We prove its properties in Appendix D.

Definition 6.1 (Filter basis). A set $\mathcal{F} \subset \mathcal{P}(X)$ is a *filter basis* in a set X if it is

Name	Universe	$\mathcal{F}(X)$	Reference
\emptyset Trivial	sets	$\{\emptyset\}$	[12]
\emptyset Cofinite	sets	$\mathcal{P}(X)$	[8], [12]
\emptyset Asymptotic	$\bigcup_{d \in \mathbb{N}} \mathcal{P}(\mathbb{R}^d)$	$\{X^{\geq y} : y \in \mathbb{R}^d\}$	[1]
\emptyset Co-asymptotic	$\bigcup_{d \in \mathbb{N}} \mathcal{P}(\mathbb{R}^d)$	$\{(X \setminus X^{< y}) : y \in \mathbb{R}^d\}$	[2]
\emptyset Full	sets	$\{X\}$	[12]

Table 6.1: Example versions of local linear dominance. Here U is the universe, $X \in U$, and $\mathcal{F}(X)$ is the filter basis in X . The name is used to replace the word ‘local’, as in asymptotic linear dominance.

non-empty

$$\mathcal{F} \neq \emptyset,$$

\subset -directed

$$\forall A, B \in \mathcal{F}, \exists C \in \mathcal{F} : C \subset A \text{ and } C \subset B.$$

Note 6.2 (Filter basis may not be proper). Some authors require a filter basis to be proper — $\emptyset \notin \mathcal{F}(X)$. We allow a filter basis to be non-proper. ■

Definition 6.3 (Local linear dominance). *Local linear dominance* Θ on $X \in U$ is defined by $g \in \Theta_X(f)$ if and only if

$$\exists c \in \mathbb{R}^{>0}, \exists A \in \mathcal{F}(X) : (g|A) \leq c(f|A), \quad (6.4)$$

where $\{\mathcal{F}(X) \subset \mathcal{P}(X) : X \in U\}$ is a class of filter bases with *induced sub-structure*:

$$\mathcal{F}(D) = \{A \cap D : A \in \mathcal{F}(X)\}, \quad (6.5)$$

for all $D \subset X$.

Note 6.6 (Versions). Each choice of filter bases corresponds to a version of local linear dominance. Some such versions are given in Table 6.1. ■

Note 6.7 (Filter basis and limits). A filter basis in X is the minimal amount of structure needed to make sense of the limit of a function $f \in R_X$. In fact, a local linear dominance can be characterized by a ratio-limit. ■

Note 6.8 (Motivation). We have already shown in Chapter 4 that only one instance of local linear dominance works for algorithm analysis. Why study local linear dominances? There are two reasons.

First, local linear dominance is commonly used for local function approximation in various fields of mathematics. Such results may indirectly find their way into a complexity analysis, in which case we need a way

transfer such results to the algorithmic side. This can be done using the tools provided in Section 6.2.

Second, local linear dominances help to train the intuition behind the primitive properties, because they provide successively better approximations to linear dominance. This is shown in Appendix D. ■

Note 6.9 (Eventually non-negative). A function $f: X \rightarrow \mathbb{R}$ is *eventually non-negative*, if there exists $A \in \mathcal{F}(X)$, such that $(f|_A) \geq 0$. It is possible to generalize the definition of local linear dominance to functions which are eventually non-negative. When $\mathcal{F}(X) = \{X\}$, eventually non-negative reduces to non-negative. ■

Theorem 6.10 (Θ is almost equal to Θ for cofinite filter sets). Suppose $|X \setminus A| < \infty$, for all $A \in \mathcal{F}(X)$. Then

$$\Theta_X(g) = \Theta_X(g), \quad (6.11)$$

for all $g \in R_X^{>0}$.

Example 6.12 (Linear dominance from asymptotic linear dominance). Let $f \in R_{\mathbb{N}^{>0}}$ be such that $f(n) = n^2|\sin(n)| + n + 3$. It holds that $f \in \Theta_{\mathbb{N}^{>0}}(n^2)$ by Example 6.24. Since $n^2 > 0$, and $|\mathbb{N}^{>0} \setminus \mathbb{N}^{\geq y}| < \infty$, for all $y \in \mathbb{R}$, it holds that $f \in \Theta_{\mathbb{N}^{>0}}(n^2)$ by [Θ is almost equal to Θ for cofinite filter sets] (D.59). △

6.2 Limit theorems

In this section we show how to transfer a result from a local linear dominance Θ to linear dominance Θ . First, the filter bases associated with Θ make it possible to define the concepts of limit superior, limit inferior, and limit. These limits can then be used to characterize the Θ -notation. Second, $f \in \Theta_X(g)$ can sometimes be used to deduce $f \in \Theta_X(g)$. The proofs are given in Appendix E.

Note 6.13 (Related notations for local linear dominance). We shall denote the related notations corresponding to Θ by Ω , ω , Θ , and ω . ■

Definition 6.14 (Limit superior and limit inferior under a filter basis). Let $\mathcal{F} \subset \mathcal{P}(X)$ be a filter basis in a set X . Then

$$\begin{aligned}\limsup_{\mathcal{F}} f &:= \inf \left\{ \sup \vec{f}(A) : A \in \mathcal{F} \right\}, \\ \liminf_{\mathcal{F}} f &:= \sup \left\{ \inf \vec{f}(A) : A \in \mathcal{F} \right\},\end{aligned}\tag{6.15}$$

for all $f \in R_X$.

Note 6.16 (Existence of limit superior and limit inferior). The \limsup and \liminf are called the *limit superior* and the *limit inferior*, respectively. By the completeness of $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$, both of them are well-defined as a number in $\overline{\mathbb{R}}$. ■

Definition 6.17 (Limit under a filter basis). The *limit* of $f \in R_X$ under a filter basis $\mathcal{F} \subset \mathcal{P}(X)$ is

$$\lim_{\mathcal{F}} f = c,\tag{6.18}$$

whenever $\limsup_{\mathcal{F}} f = \liminf_{\mathcal{F}} f = c \in \overline{\mathbb{R}}$.

Note 6.19 (Division by zero and infinity). In this section, we use the conventions that $\alpha/0 = \infty \in \overline{\mathbb{R}}$, for all $\alpha \in \mathbb{R}^{>0}$, and that $\beta/\infty = 0$, for all $\beta \in \mathbb{R}$. ■

Theorem 6.20 (Relation between ratio-limits).

$$\limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} = 1 / \liminf_{\mathcal{F}(F)} \frac{(g|F)}{(f|F)},\tag{6.21}$$

Chapter 6. Local linear dominance

for all $f, g \in R_X$, where $F = \overleftarrow{f}(\mathbb{R}^{>0})$.

Theorem 6.22 (Θ by a limit).

$$\limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} < \infty \iff f \in \Theta_X(g), \quad (6.23)$$

for all $f, g \in R_X$, where $F = \overleftarrow{f}(\mathbb{R}^{>0})$.

Example 6.24 (Asymptotic linear dominance from a limit in $\mathbb{N}^{>0}$).
Consider asymptotic linear dominance $\Theta_{\mathbb{N}^{>0}}$. Let $f \in R_{\mathbb{N}^{>0}}$ be such that $f(n) = n^2|\sin(n)| + n + 3$. Since $f(n) > 0$, and

$$\limsup_{\mathcal{F}(\mathbb{N}^{>0})} \frac{f(n)}{n^2} = \limsup_{n \rightarrow \infty} \frac{f(n)}{n^2} = 1, \quad (6.25)$$

it holds that $f \in \Theta_{\mathbb{N}^{>0}}(n^2)$ by [Θ by a limit] (E.21) \triangle

Theorem 6.26 (\mathfrak{Q} by a limit).

$$\liminf_{\mathcal{F}(G)} \frac{(f|G)}{(g|G)} > 0 \iff f \in \mathfrak{Q}_X(g), \quad (6.27)$$

for all $f, g \in R_X$, where $G = \overleftarrow{g}(\mathbb{R}^{>0})$.

Theorem 6.28 (Θ by a limit).

$$\left(\limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} < \infty \text{ and } \liminf_{\mathcal{F}(G)} \frac{(f|G)}{(g|G)} = 0 \right) \iff f \in \Theta_X(g), \quad (6.29)$$

for all $f, g \in R_X$, where $F = \overleftarrow{f}(\mathbb{R}^{>0})$ and $G = \overleftarrow{g}(\mathbb{R}^{>0})$.

Example 6.30 (Co-asymptotic linear dominance from a limit in \mathbb{N}^2). Consider co-asymptotic linear dominance $\Theta_{\mathbb{N}^2}$. Let $f \in R_{\mathbb{N}^2}$ be such that $f(m, n) = (m + n + 1) \exp(-mn)$. Since $f(m, n) > 0$, $m + n + 1 > 0$, and

$$\begin{aligned} \limsup_{\mathcal{F}(\mathbb{N}^2)} \frac{f(m, n)}{m + n + 1} &= 1, \\ \liminf_{\mathcal{F}(\mathbb{N}^2)} \frac{f(m, n)}{m + n + 1} &= 0, \end{aligned} \tag{6.31}$$

it holds that $f \in \mathfrak{o}_{\mathbb{N}^2}(m + n + 1)$ by $[\mathfrak{o}$ by a limit] (E.30). \triangle

Example 6.32 (Sufficient limit-condition for \mathfrak{o}). In particular,

$$\limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} = 0 \implies f \in \mathfrak{o}_X(g), \tag{6.33}$$

but this is only a sufficient condition. \triangle

Theorem 6.34 (Traditional \mathfrak{o}_X by a limit).

$$\begin{aligned} \forall \varepsilon \in \mathbb{R}^{>0} : \exists A \in \mathcal{F}(X) : (f|A) \leq \varepsilon(g|A) \\ \iff \limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} = 0, \end{aligned} \tag{6.35}$$

for all $f, g \in R_X$, where $F = \overleftarrow{f}(\mathbb{R}^{>0})$.

Example 6.36 (Asymptotic linear dominance from a limit in \mathbb{N}^2). Consider asymptotic linear dominance $\Theta_{\mathbb{N}^2}$. Let $f \in R_{\mathbb{N}^2}$ be such that $f(m, n) = (m + n + 1) \exp(-mn)$. Since $f > 0$, and

$$\limsup_{\mathcal{F}(\mathbb{N}^2)} \frac{f}{1} = 0, \tag{6.37}$$

it holds that $f \in \mathfrak{o}_{\mathbb{N}^2}(1)$ by $[\mathfrak{o}$ by a limit] (E.30). \triangle

Theorem 6.38 ($\hat{\omega}$ by a limit).

$$\left(\limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} = \infty \text{ and } \liminf_{\mathcal{F}(G)} \frac{(f|G)}{(g|G)} > 0 \right) \iff f \in \hat{\omega}_X(g), \quad (6.39)$$

for all $f, g \in R_X$, where $F = \overleftarrow{f}(\mathbb{R}^{>0})$ and $G = \overleftarrow{g}(\mathbb{R}^{>0})$.

Example 6.40 (Sufficient limit-condition for $\hat{\omega}$). In particular,

$$\liminf_{\mathcal{F}(G)} \frac{(f|G)}{(g|G)} = \infty \implies f \in \hat{\omega}_X(g), \quad (6.41)$$

but this is only a sufficient condition. \triangle

Chapter 7

Conclusion

This paper provides a rigorous mathematical foundation for the O -notation and its related notations in algorithm analysis. To find the appropriate definition, an exhaustive list of desirable properties was constructed, and their relations were studied. This revealed that the desirable properties can be reduced to 8 primitive properties which imply the others. It was shown that these primitive properties are equivalent to the definition of the O -notation as linear dominance. Master theorems were shown to hold for linear dominance, and Θ -mappings were defined for easily proving new rules for the Θ -notation. Other existing definitions were studied and compared to each other based on the primitive properties. Some misuses of the O -notation from the literature were pointed out.

We hope this paper to improve the teaching of the topic, and to improve the communication between computer scientists. Computer scientists have used the O -notation correctly intuitively. That intuition is now backed by a solid mathematical foundation.

Bibliography

- [1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, USA, 2nd edition, 2001 (cited on pages 7, 19, 20, 82).
- [2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009 (cited on pages 8, 18, 20, 21, 29, 42–44, 75, 82, 139).
- [3] Paul Bachmann. *Die Analytische Zahlentheorie*. B. G. Teubner, Leipzig, 1894 (cited on pages 15, 16).
- [4] Edmund Landau. *Handbuch der Lehre von der Verteilung der Primzahlen*. Volume 2. B. G. Teubner, Leipzig, 1909 (cited on pages 16, 17).
- [5] Donald E. Knuth. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison-Wesley, 1st edition, 1968 (cited on page 16).
- [6] Donald E. Knuth. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison-Wesley, 2nd edition, 1973 (cited on page 16).
- [7] Edward M. Reingold. On the Optimality of Some Set Algorithms. *J. ACM*, 19(4):649–659, October 1972 (cited on page 16).
- [8] Alfred V. Aho and John E. Hopcroft. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1974 (cited on pages 16–18, 20, 82).
- [9] Donald E. Knuth. Big Omicron and Big Omega and Big Theta. *SIGACT News*, 8(2):18–24, April 1976 (cited on page 17).
- [10] Paul M. B. Vitányi and Lambert Meertens. Big Omega Versus the Wild Functions. *ACM SIGACT News*, 16(4):56–59, April 1985 (cited on page 17).

Bibliography

- [11] Rodney R. Howell. On Asymptotic Notation with Multiple Variables. 2008 (cited on pages 20, 101).
- [12] Kalle Rutanen, Germán Gómez-Herrero, Sirkka-Liisa Eriksson, and Karen Egiazarian. A General Definition of the O-notation for Algorithm Analysis. *Bulletin of the EATCS*, 117, 2015 (cited on pages 22, 82).
- [13] Kalle Rutanen, Germán Gómez-Herrero, Sirkka-Liisa Eriksson, and Karen Egiazarian. A General Definition of the O-notation for Algorithm Analysis. *CoRR*, abs/1309.3210, 2013 (cited on page 22).
- [14] Yuri Gurevich. Sequential Abstract State Machines Capture Sequential Algorithms. *ACM Transactions on Computational Logic*, 1(1):77–111, July 2000 (cited on page 25).
- [15] Andreas Blass and Yuri Gurevich. Abstract State Machines Capture Parallel Algorithms. *ACM Transactions on Computational Logic*, 4:578–651, 2003 (cited on page 25).
- [16] Egon Börger and Robert F. Stärk. *Abstract State Machines: A Method for High-Level System Design and Analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003 (cited on pages 25–27).
- [17] Lenore Blum, Michael Shub, and Stephen Smale. On a Theory of Computation over the Real Numbers; NP Completeness, Recursive Functions and Universal Machines. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*. In SFCS '88. IEEE Computer Society, Washington, DC, USA, 1988, pages 387–397 (cited on pages 27, 29).
- [18] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. Edition, 2008 (cited on page 29).
- [19] Stephen A. Cook and Robert A. Reckhow. Time-bounded Random Access Machines. In *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*. In STOC '72. ACM, Denver, Colorado, USA, 1972, pages 73–80 (cited on page 29).
- [20] Prosenjit Bose, Karim Douïeb, Vida Dujmović, John Howat, and Pat Morin. Fast Local Searches and Updates in Bounded Universes. *Computational Geometry*, 46(2):181–189, 2013 (cited on page 39).
- [21] Arne Andersson and Mikkel Thorup. Dynamic Ordered Sets with Exponential Search Trees. *J. ACM*, 54(3), June 2007 (cited on pages 43, 44).

Bibliography

- [22] Elliott Mendelson. *Introduction to Mathematical Logic*. Chapman & Hall / CRC, 5th edition, 2009 (cited on page 95).

Appendix A

Notation

In this section we provide the definitions for the used notation. This is to avoid ambiguities arising from differing conventions, such as a differing order of composition of functions. We will assume the Zermelo-Fraenkel set-theory with the axiom of choice, abbreviated ZFC.

We refer to theorems in the form [Linear dominance has primitive properties] (4.1) — a short summary followed by a number. We believe this is more useful than Theorem 4.1 when studying the proofs. The numberings in equations, definitions, theorems etc. share the same counter.

Definition A.1 (Numbers). The set of natural numbers, integers, and real numbers are denoted by $\mathbb{N} = \{0, 1, 2, \dots\}$, \mathbb{Z} , and \mathbb{R} , respectively.

Definition A.2 (Subsets of a set). The *set of subsets* of a set X is denoted by $\mathcal{P}(X)$.

Definition A.3 (Finite subsets of a set). The *set of finite subsets* of a set X is denoted by

$$\widehat{\mathcal{P}}(X) := \{D \in \mathcal{P}(X) : |D| < \infty\}. \quad (\text{A.4})$$

Appendix A. Notation

Definition A.5 (Cofinite subsets of a set). The *set of cofinite subsets* of a set X is denoted by

$$\overline{\mathcal{P}}(X) := \{D \in \mathcal{P}(X) : |X \setminus D| < \infty\}. \quad (\text{A.6})$$

Definition A.7 (Cover of a set). A set C is a *cover* of a set X , if

$$X \subset \bigcup C. \quad (\text{A.8})$$

Definition A.9 (Set of functions). The set of functions from a set X to a set Y is denoted by $X \rightarrow Y$, or alternatively by Y^X .

Definition A.10 (Identity function). The *identity function* in a set X is a function $\text{id}_X: X \rightarrow X$ such that

$$\text{id}_X(x) = x. \quad (\text{A.11})$$

Definition A.12 (Composition). The *composition* of $g: Y \rightarrow Z$ and $f: X \rightarrow Y$ is $(g \circ f): X \rightarrow Z$ such that $(g \circ f)(x) = g(f(x))$.

Definition A.13 (Restriction). The *restriction* of $f: X \rightarrow Y$ to $D \subset X$ is $(f|D): D \rightarrow Y$ such that $(f|D)(x) = f(x)$.

Definition A.14 (Inverse of a function). The *inverse* of a function $f: X \rightarrow Y$ is $f^{-1}: Y \rightarrow X$ such that

$$\begin{aligned} f(f^{-1}(y)) &= y, \\ f^{-1}(f(x)) &= x, \end{aligned} \quad (\text{A.15})$$

for all $x \in X, y \in Y$.

Appendix A. Notation

Definition A.16 (Image of a set under a function). The *image* of a set $S \subset X$ under a function $f: X \rightarrow Y$ is

$$\vec{f}(S) = \{f(x) : x \in S\}. \quad (\text{A.17})$$

Definition A.18 (Pre-image of a set under a function). The *pre-image* of a set $S \subset Y$ under a function $f: X \rightarrow Y$ is

$$\overleftarrow{f}(S) = \{x \in X : f(x) \in S\}. \quad (\text{A.19})$$

Definition A.20 (Increasing function). A function $f: \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$ is *increasing*, if $x \leq y \implies f(x) \leq f(y)$, for all $x, y \in \mathbb{R}^{\geq 0}$.

Example A.21 (Non-negative powers are increasing). Let $f: \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$ be such that $f(x) = x^\alpha$, where $\alpha \in \mathbb{R}^{\geq 0}$. Then f is increasing. \triangle

Definition A.22 (Class of sets). A *class of sets* is an unambiguous collection of sets, which may or may not be a set itself.

Definition A.23 (Proper class). A *proper class* is a class of sets which is not a set.

Example A.24 (Classes of sets). Every set is a class of sets. The collection of all sets in ZFC is a proper class. \triangle

Note A.25 (Formalization of proper classes). The proper classes are not formalizable in the ZFC set-theory. This is not a problem for two reasons. First, everything in this paper can be carried through without forming proper classes, by only referring to the class elements and their relationships. Alternatively, we may adopt the von Neumann-Bernays-Gödel set theory [22] — a conservative extension of ZFC which formalizes

Appendix A. Notation

proper classes. ■

Definition A.26 (Set of relations). The set of relations between a set X and a set Y is denoted by $X \leftrightarrow Y$.

Definition A.27 (Reflexive relation). A relation $\sim: X \leftrightarrow X$ is *reflexive*, if $x \sim x$, for all $x \in X$.

Definition A.28 (Transitive relation). A relation $\sim: X \leftrightarrow X$ is *transitive*, if

$$x \sim y \text{ and } y \sim z \implies x \sim z, \quad (\text{A.29})$$

for all $x, y, z \in X$.

Definition A.30 (Preorder). A *preorder* in a set X is a reflexive and transitive relation $\preceq_X: X \leftrightarrow X$.

Definition A.31 (Filtered set). Let X be a set, and $\sim: X \leftrightarrow X$. Then

$$X^{\sim y} := \{x \in X : x \sim y\}, \quad (\text{A.32})$$

for all $y \in X$.

Example A.33. $\mathbb{R}^{\geq 0}$ is the set of non-negative real numbers. △

Definition A.34 (Indicator function). The *indicator function* of $S \subset X$ in a set X is a function $[S]_X: X \rightarrow \mathbb{R}^{\geq 0}$ such that

$$[S]_X(x) = \begin{cases} 1, & x \in S, \\ 0, & x \notin S. \end{cases} \quad (\text{A.35})$$

Appendix A. Notation

Definition A.36 (Non-negative real-valued functions). We are specifically interested in non-negative real-valued functions; we define

$$R_X := (X \rightarrow \mathbb{R}^{\geq 0}), \quad (\text{A.37})$$

where X is a set.

Definition A.38 (Extension of a unary operator to functions). A unary operator $\ominus: \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$ is extended to functions $f \in R_X$ by

$$(\ominus f)(x) = \ominus f(x), \quad (\text{A.39})$$

for all $x \in X$.

Definition A.40 (Extension of a binary operator to functions). A binary operator $\oplus: \mathbb{R}^{\geq 0} \times \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$ is extended to functions $f, g \in R_X$ by

$$(f \oplus g)(x) = f(x) \oplus g(x), \quad (\text{A.41})$$

for all $x \in X$.

Definition A.42 (Extension of a relation to functions). A relation $\sim: Y \leftrightarrow Y$ is extended to functions $f, g: X \rightarrow Y$ by

$$f \sim g :\Longleftrightarrow \forall x \in X : f(x) \sim g(x). \quad (\text{A.43})$$

Example A.44. It holds that $f \geq 0$, for all $f \in R_X$. \triangle

Example A.45. Let $x, y \in \mathbb{R}^d$, where $d \in \mathbb{N}^{>0}$. Then

$$x \geq y \Longleftrightarrow \forall i \in d : x_i \geq y_i. \quad (\text{A.46})$$

\triangle

Appendix A. Notation

Note A.47 (Relation-lifting pitfall). We adopted an implicit convention for lifting relations from sets to functions. This introduces a potential for notational ambiguity.

Consider the formula $f \not\prec g$, where $f, g: X \rightarrow \mathbb{R}$. If the negation refers to the original relation, then the formula is equivalent to

$$\forall x \in X : f(x) \geq g(x). \quad (\text{A.48})$$

However, if the negation refers to the lifted relation, then the formula is equivalent to

$$\exists x \in X : f(x) \geq g(x). \quad (\text{A.49})$$

We avoid this ambiguity by not using negation for lifted relations. In particular, we denote the filtered sets of co-asymptotic linear dominance by $(X \setminus X^{<^y})$ — not by $X \not\prec^y$. ■

Definition A.50 (Unary function for a set of functions). A unary function $\ominus: R_X \rightarrow R_Y$ is extended to $\mathcal{P}(R_X) \rightarrow \mathcal{P}(R_Y)$ by

$$\ominus U = \{\ominus u : u \in U\}. \quad (\text{A.51})$$

Definition A.52 (Binary function for sets of functions). A binary function $\oplus: R_X \times R_X \rightarrow R_Y$ is extended to $\mathcal{P}(R_X) \times \mathcal{P}(R_X) \rightarrow \mathcal{P}(R_Y)$ by

$$U \oplus V = \{u \oplus v : (u, v) \in U \times V\}. \quad (\text{A.53})$$

Definition A.54 (Iteration). The i :th iteration of $f: X \rightarrow X$, where $i \in \mathbb{N}$, is $f^{(i)}: X \rightarrow X$ such that

$$f^{(i)}(x) = \begin{cases} x, & i = 0, \\ f^{(i-1)}(f(x)), & i > 0. \end{cases} \quad (\text{A.55})$$

Definition A.56 (Projection). A *projection* is a function $[\underline{X} \times Y]: X \times$

Appendix A. Notation

$Y \rightarrow X$ such that

$$[\underline{X} \times Y](x, y) = x. \quad (\text{A.57})$$

Definition A.58 (Universe). A *universe* is a class U of sets such that

1. $\mathbb{N}^0 \in U$,
2. $\mathbb{N}^1 \in U$,
3. $\forall X \in U : \mathcal{P}(X) \subset U$,
4. $\forall X, Y \in U : X \times Y \in U$.

Definition A.59 (Sub-universe). A *sub-universe* of a universe U is a subclass $V \subset U$ which is also a universe.

Example A.60 (Examples of universes). The smallest universe is given by

$$\bigcup_{d \in \mathbb{N}} \mathcal{P}(\mathbb{N}^d) \quad (\text{A.61})$$

Every universe contains this set as a sub-universe. The class of all sets is a universe which is a proper class. \triangle

Definition A.62 (Computational problem). A *computational problem* is a function $P: X \rightarrow Y$, where $X, Y \in U$.

Definition A.63 (Set of algorithms). The set of algorithms^a which solve a problem $P: X \rightarrow Y$, under a given model of computation, is denoted by $\mathcal{A}(P)$. The set of all algorithms from X to Y is

$$X \rightarrow Y = \bigcup_{P: X \rightarrow Y} \mathcal{A}(P). \quad (\text{A.64})$$

Appendix A. Notation

^aWe will define the term *algorithm* formally in Section 2.1.

Definition A.65 (Composition of algorithms). The *composition* of algorithms $G: Y \rightarrow Z$ and $F: X \rightarrow Y$ is the algorithm $G \circ F: X \rightarrow Z$, which is obtained by using the output of F as the input of G .

Note A.66. We will sometimes use an algorithm $F: X \rightarrow Y$ as if it were its induced function instead. ■

Definition A.67 (Cost function of an algorithm). The cost function of an algorithm $F \in \mathcal{A}(P)$ is denoted by $f_F \in R_X$.

Appendix B

Howell's counterexample

In this section, we consider Howell's counterexample [11], which shows that asymptotic linear dominance Θ does not satisfy subset-summability.¹

Example B.1 (Howell's counterexample). Let $X = \mathbb{N}^2$, $\hat{g} \in R_X$ be such that

$$\hat{g}(m, n) = \begin{cases} 2^n, & m = 0, \\ mn, & m > 0, \end{cases} \quad (\text{B.2})$$

and $g \in R_X$ be such that $g(m, n) = mn$. Then

$$\begin{aligned} \sum_{i=0}^m \hat{g}(i, n) &= 2^n + m(m+1)n/2 \\ &\not\in \Theta_X(m(m+1)n/2) \\ &= \Theta_X\left(\sum_{i=0}^m g(i, n)\right). \end{aligned} \quad (\text{B.3})$$

△

Note B.4 (Howell's requirements). To be precise, Howell required the following properties from an O-notation:

¹We have fixed the error of having the sum-index i run only to $m-1$.

Appendix B. Howell's counterexample

asymptotic refinement

$$O_X(f) \subset \Theta_X(f) \quad (\text{B.5})$$

reflexivity

$$f \in O_X(f) \quad (\text{B.6})$$

asymptotic order-consistency

$$\left(\exists y \in \mathbb{N}^d : \left(\widehat{f} | X^{\geq y} \right) \leq (f | X^{\geq y}) \right) \implies O_X(\widehat{f}) \subset O_X(f) \quad (\text{B.7})$$

simple subset-summability

$$O_X \left(\sum_{i=0}^{n_k} \widehat{g}(n_1, \dots, n_{k-1}, i, n_{k+1}, \dots, n_d) \right) \in O_X \left(\sum_{i=0}^{n_k} g(n_1, \dots, n_{k-1}, i, n_{k+1}, \dots, n_d) \right), \quad (\text{B.8})$$

where $X = \mathbb{N}^d$, $\widehat{f}, f, g \in R_X$, $\widehat{g} \in O_X(g)$, and $k \in [1, d] \subset \mathbb{N}$. ■

While Howell did not do so, we claim that any sensible definition of O -notation must also satisfy scale-invariance:

$$O_X(\alpha f) = O_X(f), \quad (\text{B.9})$$

for all $f \in R_X$ and $\alpha \in \mathbb{R}^{>0}$. The following theorem then shows that Howell's result only concerns the Θ -notation.

Theorem B.10 (Howell's definition is asymptotic dominance). O has asymptotic order-consistency, scale-invariance, reflexivity, and asymptotic refinement. $\implies O = \Theta$.

Appendix B. Howell's counterexample

Proof. By asymptotic order-consistency, scale-invariance, and reflexivity,

$$\begin{aligned}
 & \hat{f} \in \Theta_X(f) \\
 \implies & \exists c \in \mathbb{R}^{>0}, \exists y \in \mathbb{N}^d : \left(\hat{f} | X^{\geq y} \right) \leq c(f | X^{\geq y}) \\
 \implies & \exists c \in \mathbb{R}^{>0} : O_X(\hat{f}) \subset O_X(cf) \\
 \implies & O_X(\hat{f}) \subset O_X(f) \\
 \implies & \hat{f} \in O_X(f),
 \end{aligned} \tag{B.11}$$

for all $f, \hat{f} \in R_X$. Therefore $\Theta_X(f) \subset O_X(f)$. It follows from asymptotic refinement that $O_X(f) = \Theta_X(f)$. \square

Appendix C

Proofs of implied properties

In this section we will show that the primitive properties imply the rest of the properties.

Definition C.1 (Composite property). A *composite property* is a property which can be equivalently expressed in terms of primitive properties.

Proposition C.2 (Sub-homogeneity in $\mathbb{Q}^{\geq 0}$ is a composite). O_X has sub-homogeneity in $\mathbb{Q}^{\geq 0}$. $\iff O_X$ has sub-homogeneity in \mathbb{N} and sub-homogeneity in $1/\mathbb{N}^{>0}$.

Proof.

\implies

Suppose O_X has sub-homogeneity in $\mathbb{Q}^{\geq 0}$. Then O_X has sub-homogeneity in \mathbb{N} , since $\mathbb{N} \subset \mathbb{Q}^{\geq 0}$, and sub-homogeneity in $1/\mathbb{N}^{>0}$, since $(1/\mathbb{N}^{>0}) \subset \mathbb{Q}^{>0}$.

\impliedby

Suppose O_X has sub-homogeneity in \mathbb{N} and sub-homogeneity in $1/\mathbb{N}^{>0}$. Let $f, g, u \in R_X$ be such that $\overrightarrow{u}(X) \subset \mathbb{Q}^{\geq 0}$. Then there exists $p, q \in R_X$, such that $\overrightarrow{p}(X) \subset \mathbb{N}^{\geq 0}$, $\overrightarrow{q}(X) \subset \mathbb{N}^{>0}$, and $u = p/q$. By sub-homogeneity in \mathbb{N} and sub-

Appendix C. Proofs of implied properties

Property	\leq	T	1	α	L	*	/	\circ	Th
Sub-homogeneity in $\mathbb{Q}^{\geq 0}$						✓	✓		C.2
Sub-homogeneity	✓	✓		✓		✓	✓		C.4
Reflexivity	✓								C.7
Zero-separation	✓	✓	✓			✓			C.8
Orderness	✓	✓							C.10
Zero-triviality	✓	✓	✓			✓	✓	✓	C.11
Injective super-comp.	✓				✓			✓	C.13
Sub-restrictability								✓	C.18
Super-restrictability	✓				✓				C.20
Scalar homogeneity	✓	✓		✓					C.21
Super-homogeneity	✓	✓	✓	✓	✓	✓	✓	✓	C.23
Sub-multiplicativity	✓	✓		✓		✓	✓		C.25
Super-multiplicativity	✓	✓	✓	✓	✓	✓	✓	✓	C.29
Additive consistency	✓	✓			✓			✓	C.47
Maximum consistency	✓	✓			✓			✓	C.50
Multiplicative cons.	✓	✓			✓			✓	C.53
Maximum rule	✓	✓			✓			✓	C.56
Summation rule	✓	✓		✓					C.60
Maximum-sum rule	✓	✓		✓					C.63
Additivity	✓	✓		✓	✓			✓	C.66
Bounded translation-inv.	✓	✓		✓					C.70
Extensibility								✓	C.72
Subset-summability	✓	✓	✓	✓	✓	✓	✓	✓	C.73

Table C.1: The primitive properties that imply a given non-primitive property. The abbreviations are: \leq for order-consistency, T for transitivity, 1 for one-separation, α for scale-invariance, L for locality, * for sub-homogeneity in \mathbb{N} , / for sub-homogeneity in $1/\mathbb{N}^{>0}$, \circ for sub-composability, and Th for theorem.

Appendix C. Proofs of implied properties

homogeneity in $1/\mathbb{N}^{>0}$,

$$\begin{aligned}
 & f \in O_X(g) \\
 \implies & pf \in O_X(pg) \\
 \implies & \frac{p}{q}f \in O_X\left(\frac{p}{q}g\right) \\
 \implies & uf \in O_X(ug).
 \end{aligned} \tag{C.3}$$

□

Proposition C.4 (Sub-homogeneity is implied). O_X has order-consistency, transitivity, scale-invariance, sub-homogeneity in \mathbb{N} , and sub-homogeneity in $1/\mathbb{N}^{>0}$. $\implies O_X$ has sub-homogeneity.

Proof. O_X has sub-homogeneity in $\mathbb{Q}^{\geq 0}$ by [Sub-homogeneity in $\mathbb{Q}^{\geq 0}$ is a composite] (C.2). Let $f, g, u \in R_X$, and $h: \mathbb{R}^{\geq 0} \rightarrow \mathbb{Q}^{\geq 0}$ be such that

$$x \leq h(x) \leq 2x. \tag{C.5}$$

By sub-homogeneity in $\mathbb{Q}^{\geq 0}$, order-consistency, transitivity, and scale-invariance,

$$\begin{aligned}
 & f \in O_X(g) \\
 \implies & (h \circ u)f \in O_X((h \circ u)g) \\
 \implies & uf \in O_X(2ug) \\
 \implies & uf \in O_X(ug).
 \end{aligned} \tag{C.6}$$

□

Proposition C.7 (Reflexivity is implied). O_X has order-consistency $\implies O_X$ has reflexivity.

Proof. By order-consistency, $f \leq f \implies f \in O_X(f)$, for all $f \in R_X$; O_X has reflexivity. □

Proposition C.8 (Zero-separation is implied). O has order-consistency, transitivity, one-separation, and sub-homogeneity in $\mathbb{N} \implies O$ has

Appendix C. Proofs of implied properties

zero-separation.

Proof. Suppose O does not have zero-separation, so that $1 \in O_{\mathbb{N}^{>0}}(0)$. By sub-homogeneity in \mathbb{N} , $n \in O_{\mathbb{N}^{>0}}(0)$. By order-consistency, $0 \in O_{\mathbb{N}^{>0}}(1)$. By transitivity, $n \in O_{\mathbb{N}^{>0}}(1)$. This contradicts one-separation. \square

Proposition C.9 (Orderness is a composite). O_X has reflexivity and transitivity $\iff O_X$ has orderness.

Proof.

\implies

Assume $f \in O_X(g)$. Let $\hat{f} \in O_X(f)$. By transitivity, $\hat{f} \in O_X(g)$, and so $O_X(f) \subset O_X(g)$. Assume $O_X(f) \subset O_X(g)$. By reflexivity, $f \in O_X(f)$. Therefore $f \in O_X(g)$, and so O_X has orderness.

\impliedby

By orderness, $f \in O_X(f) \iff O_X(f) \subset O_X(f)$. Therefore O_X has reflexivity. Let $f \in O_X(g)$, and $g \in O_X(h)$. By orderness, $O_X(g) \subset O_X(h)$. Therefore $f \in O_X(h)$, and so O_X has transitivity. \square

Proposition C.10 (Orderness is implied). O_X has order-consistency and transitivity. $\implies O_X$ has orderness.

Proof. O_X has reflexivity by [Reflexivity is implied] (C.7). O_X has orderness by [Orderness is a composite] (C.9). \square

Proposition C.11 (Zero-triviality is implied). O has order-consistency, transitivity, one-separation, scale-invariance, sub-homogeneity in \mathbb{N} , sub-homogeneity in $1/\mathbb{N}^{>0}$, and sub-composability. $\implies O$ has zero-triviality.

Proof. O has zero-separation by [Zero-separation is implied] (C.8) and sub-homogeneity by [Sub-homogeneity is implied] (C.4). Suppose O does not have zero-triviality, so that there exists $f \in O_X(0)$ such that $f \neq 0$. Then there

Appendix C. Proofs of implied properties

exists $y \in X$ such that $f(y) = c$, for some $c \in \mathbb{R}^{>0}$. Let $s: \mathbb{N}^{>0} \rightarrow X$ be such that $s(x) = y$. By sub-composability and sub-homogeneity,

$$\begin{aligned}
 & f \in O_X(0) \\
 \implies & f \circ s \in O_X(0) \circ s \\
 \implies & c \in O_{\mathbb{N}^{>0}}(0 \circ s) \\
 \implies & 1 \in O_{\mathbb{N}^{>0}}(0)/c \\
 \implies & 1 \in O_{\mathbb{N}^{>0}}(0/c) \\
 \implies & 1 \in O_{\mathbb{N}^{>0}}(0).
 \end{aligned} \tag{C.12}$$

This contradicts zero-separation. □

Proposition C.13 (Injective super-composability is implied). O_X has order-consistency, locality, and injective sub-composability for injective $s: Y \rightarrow X$. $\implies O_X$ has injective super-composability for s .

Proof. Let $f \in R_X$, and $\underline{s}: Y \rightarrow \vec{s}(Y)$ be such that $\underline{s}(y) = s(y)$. Then \underline{s} is bijective. Let $\hat{g} \in O_Y(f \circ s)$ and $\hat{f} \in R_X$ be such that

$$\hat{f}(x) = \begin{cases} (\hat{g} \circ \underline{s}^{-1})(x), & x \in \vec{s}(Y), \\ 0, & x \notin \vec{s}(Y). \end{cases} \tag{C.14}$$

Then $\hat{g} = \hat{f} \circ s$; we show that $\hat{f} \in O_X(f)$. By injective sub-composability

$$\begin{aligned}
 \hat{f}|_{\vec{s}(Y)} &= \hat{g} \circ \underline{s}^{-1} \\
 &\in O_Y(f \circ s) \circ \underline{s}^{-1} \\
 &\subset O_{\vec{s}(Y)}(f \circ s \circ \underline{s}^{-1}) \\
 &= O_{\vec{s}(Y)}(f|_{\vec{s}(Y)}).
 \end{aligned} \tag{C.15}$$

By order-consistency,

$$\hat{f}|_{(X \setminus \vec{s}(Y))} = 0 \in O_{X \setminus \vec{s}(Y)}(f|(X \setminus \vec{s}(Y))). \tag{C.16}$$

By locality,

$$\hat{f} \in O_X(f). \tag{C.17}$$

Therefore O_X has injective super-composability for s . □

Appendix C. Proofs of implied properties

Proposition C.18 (Sub-restrictability is implied). O_X has injective sub-composability. $\implies O_X$ has sub-restrictability.

Proof. Let $D \subset X$, and $s: D \rightarrow X$ be such that $s(x) = x$. Then s is injective. By injective sub-composability

$$\begin{aligned} O_X(f)|D &= O_X(f) \circ s \\ &\subset O_D(f \circ s) \\ &= O_D(f|D), \end{aligned} \tag{C.19}$$

for all $f \in R_X$. \square

Theorem C.20 (Super-restrictability is implied). O has order-consistency and locality. $\implies O$ has super-restrictability.

Proof. Let $D \subset X$, $\bar{D} = X \setminus D$, and $\hat{h} \in O_D(f|D)$. Let $\hat{f} \in R_X$ be such that $(\hat{f}|D) = \hat{h}$ and $(\hat{f}|\bar{D}) = 0$. Then $(\hat{f}|D) \in O_D(f|D)$, and by order-consistency, $(\hat{f}|\bar{D}) \in O_{\bar{D}}(f|\bar{D})$. By locality, $\hat{f} \in O_X(f)$. \square

Theorem C.21 (Scalar homogeneity is implied). O_X has order-consistency, transitivity, and scale-invariance. $\implies O_X$ has scalar homogeneity.

Proof. O_X has orderness by [Orderness is implied] (C.10).

\subseteq

Let $\hat{f} \in O_X(f)$, and $\alpha \in \mathbb{R}^{>0}$. By order-consistency, scale-invariance, orderness, and scale-invariance again,

$$\begin{aligned} \alpha \hat{f} &\in O_X(\alpha \hat{f}) \\ &= O_X(\hat{f}) \\ &\subset O_X(f) \\ &= O_X(\alpha f). \end{aligned} \tag{C.22}$$

Therefore $\alpha \hat{f} \in O_X(\alpha f)$.

Appendix C. Proofs of implied properties

\supseteq

Let $\hat{f} \in O_X(\alpha f)$. By scale-invariance, $\hat{f} \in O_X(f)$. By orderness, $O_X(\hat{f}) \subset O_X(f)$. By scale-invariance, $O_X((1/\alpha)\hat{f}) \subset O_X(f)$. By orderness, $(1/\alpha)\hat{f} \in O_X(f)$. Therefore $\hat{f} \in \alpha O_X(f)$, and so O_X has scalar homogeneity. \square

Theorem C.23 (Super-homogeneity is implied). O_X has order-consistency, locality, transitivity, one-separation, scale-invariance, sub-homogeneity in \mathbb{N} , sub-homogeneity in $1/\mathbb{N}^{>0}$, and sub-composability. $\implies O_X$ has super-homogeneity.

Proof. O_X has sub-restrictability by [Sub-restrictability is implied] (C.18), sub-homogeneity by [Sub-homogeneity is implied] (C.4), and zero-triviality by [Zero-triviality is implied] (C.11). Let $\hat{h} \in O_X(fg)$, $G = \overleftarrow{g}(\mathbb{R}^{>0})$, and $\bar{G} = X \setminus G$. Let $\hat{f} \in R_X$ be such that $(\hat{f}|G) = (\hat{h}|G)/(g|G)$ and $(\hat{f}|\bar{G}) = 0$. By sub-restrictability, and sub-homogeneity,

$$\begin{aligned} (\hat{f}|G) &\in \frac{O_X(fg)|G}{g|G} \\ &\subset \frac{O_G(fg|G)}{g|G} \\ &\subset O_G(f|G). \end{aligned} \tag{C.24}$$

By order-consistency, $(\hat{f}|\bar{G}) \in O_{\bar{G}}(f|\bar{G})$. By locality, $\hat{f} \in O_X(f)$. It holds that $(\hat{h}|\bar{G}) \in O_X(fg)|\bar{G}$. By sub-restrictability, $(\hat{h}|\bar{G}) \in O_{\bar{G}}(0)$. By zero-triviality, $(\hat{h}|\bar{G}) = 0$. Therefore $\hat{f}g = \hat{h}$. \square

Theorem C.25 (Sub-multiplicativity is implied). O_X has order-consistency, transitivity, scale-invariance, sub-homogeneity in \mathbb{N} , and sub-homogeneity in $1/\mathbb{N}^{>0}$. $\implies O_X$ has sub-multiplicativity.

Proof. O_X has orderness by [Orderness is implied] (C.10), and sub-homogeneity by [Sub-homogeneity is implied] (C.4). Let $f, g \in R_X$, and $\hat{f} \in O_X(f)$. By

Appendix C. Proofs of implied properties

sub-homogeneity,

$$\begin{aligned}\widehat{f}g &\in O_X(f)g \\ &\subset O_X(fg).\end{aligned}\tag{C.26}$$

By orderness,

$$O_X(\widehat{f}g) \subset O_X(fg).\tag{C.27}$$

By sub-homogeneity,

$$\begin{aligned}\widehat{f}O_X(g) &\subset O_X(\widehat{f}g) \\ &\subset O_X(fg).\end{aligned}\tag{C.28}$$

Therefore $O_X(f) \cdot O_X(g) \subset O_X(fg)$. \square

Theorem C.29 (Super-multiplicativity is implied). *O has order-consistency, transitivity, one-separation, scale-invariance, sub-homogeneity in \mathbb{N} , sub-homogeneity in $1/\mathbb{N}^{>0}$, locality, and sub-composability. \implies O has super-multiplicativity.*

Proof. O_X has sub-restrictability by [Sub-restrictability is implied] (C.18), sub-homogeneity by [Sub-homogeneity is implied] (C.4), and zero-triviality by [Zero-triviality is implied] (C.11). Let $f, g \in R_X$, and $\widehat{h} \in O_X(fg)$. Let $G = \overleftarrow{g}(\mathbb{R}^{>0})$, and $\overline{G} = X \setminus G$. Let $\widehat{f} \in R_X$ be such that $(\widehat{f}|G) = \frac{(\widehat{h}|G)}{(g|G)}$ and $(\widehat{f}|\overline{G}) = 0$. It holds that $(\widehat{h}|G) \in O_X(fg)|G$. By sub-restrictability, $(\widehat{h}|G) \in O_G(fg|G)$. Similarly, $(\widehat{h}|\overline{G}) \in O_{\overline{G}}(fg|\overline{G}) = O_{\overline{G}}(0)$. By sub-homogeneity

$$\begin{aligned}(\widehat{f}|G) &\in \frac{O_G(fg|G)}{g|G} \\ &\subset O_G\left(\frac{fg|G}{g|G}\right) \\ &= O_G(f|G).\end{aligned}\tag{C.30}$$

By order-consistency, $(\widehat{f}|\overline{G}) \in O_{\overline{G}}(f|\overline{G})$. By locality, $\widehat{f} \in O_X(f)$. By order-consistency, $g \in O_X(g)$. By definition, $(\widehat{f}|G)(g|G) = (\widehat{h}|G)$. By zero-triviality, $(\widehat{h}|\overline{G}) = 0$, and so $(\widehat{h}|\overline{G}) = (\widehat{f}|\overline{G})(g|\overline{G})$. Therefore $\widehat{h} = \widehat{f}g$; O has super-multiplicativity. \square

Appendix C. Proofs of implied properties

Theorem C.31 (Locality is implied). O_X has order-consistency, transitivity, one-separation, scale-invariance, sub-homogeneity in \mathbb{N} , sub-homogeneity in $1/\mathbb{N}^{>0}$, and sub-composability. $\implies O_X$ has locality.

Proof.

Implied properties

O_X has zero-triviality by [Zero-triviality is implied] (C.11), and sub-multiplicativity by [Sub-multiplicativity is implied] (C.25).

Reduction from a finite cover to partition

Let $f, g \in R_X$. We can refine a finite cover $C \subset \mathcal{P}(X)$ of X to a finite partition $\{A_1, \dots, A_m\} \subset \mathcal{P}(X)$ of X . In the definition of locality, we assume that

$$(f|D) \in O_D(g|D), \quad (\text{C.32})$$

for all $D \in C$. By injective sub-composability,

$$(f|A_i) \in O_{A_i}(g|A_i), \quad (\text{C.33})$$

for all $i \in [1, m]_{\mathbb{N}}$. Therefore, we may assume the finite cover of X to be a finite partition of X .

Notation

Let $\{A_1, \dots, A_m\} \subset \mathcal{P}(X)$ be a finite partition of X . Suppose that

$$(f|A_i) \in O_{A_i}(g|A_i), \quad (\text{C.34})$$

for all $i \in [1, m]_{\mathbb{N}}$. Let

$$I = \left\{ i \in [1, m]_{\mathbb{N}} : \vec{f}(A_i) \cap \mathbb{R}^{>0} \neq \emptyset \right\}. \quad (\text{C.35})$$

Let $i \in I$. Let $p_i \in X$ be such that $p_i \in A_i$, and

$$f(p_i) > 0, \quad (\text{C.36})$$

Let $c_i, d_i \in \mathbb{R}^{\geq 0}$ be such that

$$\begin{aligned} c_i &= f(p_i) \\ d_i &= g(p_i). \end{aligned} \quad (\text{C.37})$$

Appendix C. Proofs of implied properties

Let $s_i: X \rightarrow X$ be such that

$$s_i(x) = \begin{cases} x, & x \in A_i, \\ p_i, & \text{otherwise.} \end{cases} \quad (\text{C.38})$$

Positivity

By injective sub-composability,

$$(f|\{p_i\}) \in O_{\{p_i\}}(g|\{p_i\}). \quad (\text{C.39})$$

By zero-triviality, and since $c_i > 0$, it holds that $d_i > 0$.

Lift

By sub-composability,

$$(f|A_i) \circ s_i \in O_X((g|A_i) \circ s_i). \quad (\text{C.40})$$

This is equivalent to

$$(f[A_i]_X + c_i[X \setminus A_i]_X) \in O_X(g[A_i]_X + d_i[X \setminus A_i]_X). \quad (\text{C.41})$$

By order-consistency, and scale-invariance,

$$\left([A_i]_X + \frac{1}{c_i}[X \setminus A_i]_X\right) \in O_X\left([A_i]_X + \frac{1}{d_i}[X \setminus A_i]_X\right). \quad (\text{C.42})$$

By sub-multiplicativity,

$$(f[A_i]_X + [X \setminus A_i]_X) \in O_X(g[A_i]_X + [X \setminus A_i]_X). \quad (\text{C.43})$$

Sum by multiplying

We have that

$$\begin{aligned} \prod_{i \in I} (f[A_i]_X + [X \setminus A_i]_X) &= \sum_{i \in I} f[A_i]_X = f, \\ \prod_{i \in I} (g[A_i]_X + [X \setminus A_i]_X) &= \sum_{i \in I} g[A_i]_X \leq g. \end{aligned} \quad (\text{C.44})$$

By sub-multiplicativity,

$$f \in O_X\left(\sum_{i \in I} g[A_i]_X\right). \quad (\text{C.45})$$

Appendix C. Proofs of implied properties

By order-consistency, and transitivity,

$$f \in O_X(g). \quad (C.46)$$

□

Theorem C.47 (Additive consistency is implied). O_X has order-consistency, transitivity, locality, and injective sub-composability. $\implies O_X$ has additive consistency.

Proof. O_X has sub-restrictability by [Sub-restrictability is implied] (C.18). Let $f, u, v \in R_X$, and $U := \{x \in X : u(x) + v(x) > 0\}$.

\subseteq

Let $\hat{f}, \hat{g} \in O_X(f)$, and $F := \{x \in X : \hat{g}(x) \leq \hat{f}(x)\}$. Let $\hat{h} \in R_X$ be such that

$$\hat{h}(x) = \begin{cases} \frac{u(x)\hat{f}(x) + v(x)\hat{g}(x)}{u(x) + v(x)}, & x \in U \\ 0, & x \notin U. \end{cases} \quad (C.48)$$

Then

$$\begin{aligned} \hat{h}(x) &= \frac{u(x)\hat{f}(x) + v(x)\hat{g}(x)}{u(x) + v(x)} \\ &\leq \frac{u(x)\hat{f}(x) + v(x)\hat{f}(x)}{u(x) + v(x)} \\ &= \hat{f}(x), \end{aligned} \quad (C.49)$$

for all $x \in F \cap U$. Also, $\hat{h}(x) = 0 \leq \hat{f}(x)$, for all $x \in F \cap \bar{U}$. By order-consistency, $(\hat{h}|F) \in O_F(\hat{f}|F)$. By sub-restrictability and transitivity, $(\hat{h}|F) \in O_F(f|F)$. Similarly, $(\hat{h}|\bar{F}) \in O_{\bar{F}}(f|\bar{F})$. By locality, $\hat{h} \in O_X(f)$. In addition, $u\hat{f} + v\hat{g} = (u + v)\hat{h}$. Therefore $u\hat{f} + v\hat{g} \in (u + v)O_X(f)$.

\supseteq

Let $\hat{f} \in O_X(f)$. Then $(u + v)\hat{f} = u\hat{f} + v\hat{f} \in uO_X(f) + vO_X(f)$.

□

Appendix C. Proofs of implied properties

Theorem C.50 (Maximum consistency is implied). O_X has order-consistency, transitivity, locality, and injective sub-composability. $\implies O_X$ has maximum consistency.

Proof. O_X has sub-restrictability by [Sub-restrictability is implied] (C.18).

\subseteq

Let $\hat{f}, \hat{g} \in O_X(f)$, and $F := \{x \in X : \hat{g}(x) \leq \hat{f}(x)\}$. Let $\hat{h} \in R_X$ be such that

$$\hat{h} := \max(\hat{f}, \hat{g}). \quad (C.51)$$

Then

$$\begin{aligned} (\hat{h}|F) &= (\max(\hat{f}, \hat{g})|F) \\ &= (\hat{f}|F). \end{aligned} \quad (C.52)$$

By order-consistency, $(\hat{h}|F) \in O_F(\hat{f}|F)$. By sub-restrictability and transitivity, $(\hat{h}|F) \in O_F(f|F)$. Similarly, $(\hat{h}|\bar{F}) \in O_{\bar{F}}(f|\bar{F})$. By locality, $\hat{h} \in O_X(f)$. In addition, $\max(\hat{f}, \hat{g}) = \hat{h}$. Therefore $\max(\hat{f}, \hat{g}) \in O_X(f)$.

\supseteq

Let $\hat{f} \in O_X(f)$. Then $\hat{f} = \max(\hat{f}, \hat{f}) \in \max(O_X(f), O_X(f))$.

□

Theorem C.53 (Multiplicative consistency is implied). O_X has order-consistency, transitivity, locality, injective sub-composability. $\implies O_X$ has multiplicative consistency.

Proof. O_X has sub-restrictability by [Sub-restrictability is implied] (C.18). Let $f, u, v \in R_X$, and $U := \{x \in X : u(x) + v(x) > 0\}$.

\subseteq

Let $\hat{f}, \hat{g} \in O_X(f)$, and $F := \{x \in X : \hat{g}(x) \leq \hat{f}(x)\}$. Let $\hat{h} \in R_X$ be such that

$$\hat{h}(x) = \begin{cases} \left(\hat{f}(x)^{u(x)} \hat{g}(x)^{v(x)} \right)^{1/(u(x)+v(x))}, & x \in U \\ 0, & x \notin U. \end{cases} \quad (C.54)$$

Appendix C. Proofs of implied properties

Then

$$\begin{aligned}
 \widehat{h}(x) &= \left(\widehat{f}(x)^{u(x)} \widehat{g}(x)^{v(x)} \right)^{1/(u(x)+v(x))} \\
 &\leq \left(\widehat{f}(x)^{u(x)} \widehat{f}(x)^{v(x)} \right)^{1/(u(x)+v(x))} \\
 &= \widehat{f}(x).
 \end{aligned} \tag{C.55}$$

for all $x \in F \cap U$. Also, $\widehat{h}(x) = 0 \leq \widehat{f}(x)$, for all $x \in F \cap \bar{U}$. By order-consistency, $(\widehat{h}|F) \in \mathcal{O}_F(\widehat{f}|F)$. By sub-restrictability and transitivity, $(\widehat{h}|F) \in \mathcal{O}_F(f|F)$. Similarly, $(\widehat{h}|\bar{F}) \in \mathcal{O}_{\bar{F}}(f|\bar{F})$. By locality, $\widehat{h} \in \mathcal{O}_X(f)$. In addition, $\widehat{f}^u \widehat{g}^v = \widehat{h}^{u+v}$ (we assume $0^0 = 1$). Therefore $\widehat{f}^u \widehat{g}^v \in \mathcal{O}_X(f)^{u+v}$.

\supseteq

Let $\widehat{f} \in \mathcal{O}_X(f)$. Then $\widehat{f}^{u+v} = \widehat{f}^u \widehat{f}^v \in \mathcal{O}_X(f)^u \mathcal{O}_X(f)^v$.

□

Theorem C.56 (Maximum rule is implied). \mathcal{O}_X has order-consistency, transitivity, locality, and injective sub-composability. $\implies \mathcal{O}_X$ has the maximum rule.

Proof. \mathcal{O}_X has orderness by [Orderness is implied] (C.10), sub-restrictability by [Sub-restrictability is implied] (C.18), and maximum consistency by [Maximum consistency is implied] (C.50).

\subseteq

Let $f, g \in R_X$. By order-consistency, $f, g \in \mathcal{O}_X(\max(f, g))$. By orderness, $\mathcal{O}_X(f) \subset \mathcal{O}_X(\max(f, g))$ and $\mathcal{O}_X(g) \subset \mathcal{O}_X(\max(f, g))$. By maximum consistency,

$$\begin{aligned}
 \max(\mathcal{O}_X(f), \mathcal{O}_X(g)) &\subset \max(\mathcal{O}_X(\max(f, g)), \mathcal{O}_X(\max(f, g))) \\
 &= \mathcal{O}_X(\max(f, g)).
 \end{aligned} \tag{C.57}$$

\supseteq

Let $\widehat{h} \in \mathcal{O}_X(\max(f, g))$, and $F := \{x \in X : g(x) \leq f(x)\}$. Let $\widehat{f} \in R_X$ be such that

$$\widehat{f} := \widehat{h}[F]_X. \tag{C.58}$$

Appendix C. Proofs of implied properties

By order-consistency, $\hat{f} \in O_X(\hat{h})$. By transitivity, $\hat{f} \in O_X(\max(f, g))$. By sub-restrictability

$$\begin{aligned} (\hat{f}|F) &\in O_X(\max(f, g)|F) \\ &\subset O_F(f|F). \end{aligned} \tag{C.59}$$

Also, $(\hat{f}|\bar{F}) = 0 \leq (f|\bar{F})$. By order-consistency, $(\hat{f}|\bar{F}) \in O_X(f|\bar{F})$. By locality, $\hat{f} \in O_X(f)$. Similarly, let $\hat{g} \in R_X$ be such that $\hat{g} := \hat{h}[\bar{F}]_X$. Then $\hat{g} \in O_X(g)$. In addition, $\hat{h} = \max(\hat{f}, \hat{g})$. Therefore $\hat{h} \in \max(O_X(f), O_X(g))$. \square

Theorem C.60 (Summation rule is implied). O_X has order-consistency, transitivity, and scale-invariance $\implies O_X$ has the summation rule.

Proof. Let $f, g \in R_X$.

\subseteq

It holds that

$$f + g \leq 2 \max(f, g). \tag{C.61}$$

By order-consistency and transitivity, $O_X(f + g) \subset O_X(2 \max(f, g))$. By scale-invariance, $O_X(f + g) \subset O_X(\max(f, g))$.

\supseteq

It holds that

$$\max(f, g) \leq f + g. \tag{C.62}$$

By order-consistency and transitivity, $O_X(\max(f, g)) \subset O_X(f + g)$. \square

Theorem C.63 (Maximum-sum rule is implied). O_X has order-consistency, transitivity, and scale-invariance. $\implies O_X$ has the maximum-sum rule.

Proof.

Appendix C. Proofs of implied properties

\subseteq

Let $\hat{f} \in O_X(f)$ and $\hat{g} \in O_X(g)$. Let $F = \{x \in X : \hat{g}(x) \leq \hat{f}(x)\}$. Let $u \in R_X$ be such that $u := \hat{f}[F]_X$. By order-consistency, $u \in O_X(\hat{f})$. By transitivity, $u \in O_X(f)$. Similarly, let $v \in R_X$ be such that $v := \hat{g}[\overline{F}]_X$. Then $v \in O_X(g)$. In addition, $\max(\hat{f}, \hat{g}) = u + v$. Therefore $\max(\hat{f}, \hat{g}) \in O_X(f) + O_X(g)$.

\supseteq

Let $f, g \in R_X$, $\hat{f} \in O_X(f)$, and $\hat{g} \in O_X(g)$. Let

$$F = \{x \in X : \hat{g}(x) \leq \hat{f}(x)\}. \quad (\text{C.64})$$

Let $u \in R_X$ be such that $u := (\hat{f} + \hat{g})[F]_X$. Then $u \leq (2\hat{f})[F]_X \leq 2\hat{f}$. By order-consistency and transitivity, $O_X(u) \subset O_X(2\hat{f})$. By scale-invariance, $O_X(u) \subset O_X(\hat{f})$. By order-consistency, $u \in O_X(\hat{f})$. By transitivity $u \in O_X(f)$. Similarly, let $v \in R_X$ be such that

$$v := (\hat{f} + \hat{g})[\overline{F}]_X. \quad (\text{C.65})$$

Then $v \in O_X(g)$. In addition, $\max(u, v) = \hat{f} + \hat{g}$. Therefore $\hat{f} + \hat{g} \in \max(O_X(f), O_X(g))$.

□

Theorem C.66 (Additivity is implied). O_X has order-consistency, transitivity, locality, scale-invariance, and injective sub-composability. $\implies O_X$ has additivity.

Proof. Let $f, g \in R_X$. O_X has the summation rule by [Summation rule is implied] (C.60). Therefore,

$$O_X(f + g) = O_X(\max(f, g)). \quad (\text{C.67})$$

O_X has the maximum rule by [Maximum rule is implied] (C.56). Therefore,

$$O_X(\max(f, g)) = \max(O_X(f), O_X(g)). \quad (\text{C.68})$$

Appendix C. Proofs of implied properties

O_X has the maximum-sum rule by [Maximum-sum rule is implied] (C.63). Therefore,

$$\max(O_X(f), O_X(g)) = O_X(f) + O_X(g). \quad (C.69)$$

Therefore $O_X(f + g) = O_X(f) + O_X(g)$. □

Theorem C.70 (Bounded translation-invariance is implied). O_X has order-consistency, transitivity, and scale-invariance $\implies O_X$ has bounded translation-invariance.

Proof. Let $\alpha, \beta \in \mathbb{R}^{>0}$, and $f \in R_X$.

\subseteq

Since $(f + \beta)/\beta \geq 1$,

$$\begin{aligned} (f + \beta) + \alpha &\leq (f + \beta) + \alpha((f + \beta)/\beta) \\ &= (1 + \alpha/\beta)(f + \beta). \end{aligned} \quad (C.71)$$

By order-consistency and transitivity, $O_X(f + \beta + \alpha) \subset O_X((1 + \alpha/\beta)(f + \beta))$.
By scale-invariance $O_X(f + \beta + \alpha) \subset O_X(f + \beta)$

\supseteq

By order-consistency and transitivity, $O_X(f + \beta) \subset O_X(f + \beta + \alpha)$. □

Theorem C.72 (Extensibility is implied). O_X has sub-composability $\implies O_X$ has extensibility.

Proof. This follows from sub-composability by $X = X^*$, $Y = X^* \times Y^*$, and $s = [X^* \times Y^*]$. □

Theorem C.73 (Subset-summability is implied). O_X has primitive properties. $\implies O_X$ has subset-summability.

Proof. This is proved in [Subset-sum is an Θ -mapping] (5.22). □

Appendix C. Proofs of implied properties

Theorem C.74 (Subset-summability implies sub-composability).
 O_X has order-consistency, transitivity and subset-summability. $\implies O_X$ has sub-composability.

Proof. O_X has reflexivity by [Reflexivity is implied] (C.7), and orderness by [Orderness is implied] (C.10).

Let $s: X \rightarrow Y$, and $S_x = \{(s(x), 0) : x \in X\}$. Let $a: \mathbb{N} \rightarrow \mathbb{R}^{\geq 0}$ be such that $a(z) = 1$. Then for $f \in R_Y$,

$$O_X \left(\sum_{(y,z) \in S_x} a(z) f(y) \right) = O_X(f \circ s). \quad (\text{C.75})$$

Similarly, for $\bar{f} \in R_Y$,

$$O_X \left(\sum_{(y,z) \in S_x} a(z) \bar{f}(y) \right) = O_X(\bar{f} \circ s). \quad (\text{C.76})$$

By subset-summability,

$$O_X(f \circ s) \subset O_X(\bar{f} \circ s). \quad (\text{C.77})$$

By orderness,

$$(f \circ s) \in O_X(\bar{f} \circ s). \quad (\text{C.78})$$

□

Appendix D

Proofs of local linear dominance properties

In this section we prove some of the properties of a local linear dominance. We shall apply the following simplification lemma without mentioning it, since it is used in almost every proof.

Lemma D.1 (Simplification lemma for Θ). *Let $X \in U$, I be a finite set, $X_i \subset X$, $f_i \in R_{X_i}$, and $\widehat{f}_i \in \Theta_{X_i}(f_i)$, for all $i \in I$. Then there exists $c \in \mathbb{R}^{>0}$ and $A \in \mathcal{F}(X)$, such that*

$$\left(\widehat{f}_i|(X_i \cap A)\right) \leq c(f_i|(X_i \cap A)), \quad (\text{D.2})$$

for all $i \in I$.

Proof. Since $\widehat{f}_i \in \Theta_{X_i}(f_i)$, there exists $c_i \in \mathbb{R}^{>0}$ and $B_i \in \mathcal{F}(X_i)$, such that

$$\left(\widehat{f}_i|B_i\right) \leq c_i(f_i|B_i), \quad (\text{D.3})$$

for all $i \in I$. By induced sub-structure, there exists $A_i \in \mathcal{F}(X)$ such that $B_i = X_i \cap A_i$, for all $i \in I$. Since $\mathcal{F}(X)$ is \subset -directed, and I is finite, there exists $A \in \mathcal{F}(X)$ such that $A \subset \bigcap_{i \in I} A_i$. Let $c = \max\{c_i : i \in I\}$. Since $(X \cap A) \subset (X \cap A_i)$,

$$\left(\widehat{f}_i|(X_i \cap A)\right) \leq c(f_i|(X_i \cap A)), \quad (\text{D.4})$$

Appendix D. Proofs of local linear dominance properties

for all $i \in I$. □

Note D.5 (Filter basis and simplification lemma). A filter basis in X seems to be the minimal amount of structure needed to prove the simplification lemma. Indeed, we first provided the abstraction of a filter basis solely to prove this lemma in its most general form. It is only later that we noticed that this structure also allows us to define limits. ■

Theorem D.6 (Order-consistency for Θ). *Let $X \in U$, and $f, g \in R_X$. Then*

$$f \leq g \implies \Theta_X(f) \subset \Theta_X(g). \quad (\text{D.7})$$

Proof. Let $\widehat{f} \in \Theta_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$ and $A \in \mathcal{F}(X)$ such that

$$(\widehat{f}|A) \leq c(f|A). \quad (\text{D.8})$$

Since $f \leq g$,

$$(\widehat{f}|A) \leq c(g|A). \quad (\text{D.9})$$

Therefore $\widehat{f} \in \Theta_X(g)$. □

Theorem D.10 (Transitivity for Θ). *Let $X \in U$, and $f, g, h \in R_X$. Then*

$$(f \in \Theta_X(g) \text{ and } g \in \Theta_X(h)) \implies f \in \Theta_X(h). \quad (\text{D.11})$$

Proof. Let $f \in \Theta_X(g)$, and $g \in \Theta_X(h)$. Then there exists $c \in \mathbb{R}^{>0}$ and $A \in \mathcal{F}(X)$, such that

$$\begin{aligned} (f|A) &\leq c(g|A), \\ (g|A) &\leq c(h|A). \end{aligned} \quad (\text{D.12})$$

It follows that

$$\begin{aligned} (f|A) &\leq c(c(h|A)) \\ &= c^2(h|A). \end{aligned} \quad (\text{D.13})$$

Therefore $f \in \Theta_X(h)$. □

Appendix D. Proofs of local linear dominance properties

Theorem D.14 (Locality for Θ). *Let $X \in U$, $f, g \in R_X$, and $C \subset \mathcal{P}(X)$ be a finite cover of X . Then*

$$(\forall D \in C : (f|D) \in \Theta_D(g|D)) \implies f \in \Theta_X(g). \quad (\text{D.15})$$

Proof. Assume $(f|D) \in \Theta_D(g|D)$, for all $D \in C$. Then there exist $c \in \mathbb{R}^{>0}$ and $A \in \mathcal{F}(X)$ such that

$$\begin{aligned} ((f|D)|(D \cap A)) &\leq c((g|D)|(D \cap A)) \\ \iff (f|(D \cap A)) &\leq c(g|(D \cap A)), \end{aligned} \quad (\text{D.16})$$

for all $D \in C$. Since C covers X ,

$$(f|A) \leq c(g|A). \quad (\text{D.17})$$

Therefore $f \in \Theta_X(g)$. \square

Theorem D.18 (One-separation for Θ characterized).

$$(\forall A \in \mathcal{F}(\mathbb{N}^{>0}) : |A| \notin \mathbb{N}) \iff n \notin \Theta_{\mathbb{N}^{>0}}(1). \quad (\text{D.19})$$

Proof.

\implies

Suppose every set in $\mathcal{F}(\mathbb{N}^{>0})$ is infinite. Then for all $c \in \mathbb{R}^{>0}$ and $A \in \mathcal{F}(\mathbb{N}^{>0})$ there exists $n \in A$ such that $n > c1$. Therefore $n \notin \Theta_{\mathbb{N}^{>0}}(1)$.

\impliedby

Suppose some set $A \in \mathcal{F}(\mathbb{N}^{>0})$ is finite. Let $c = \max(A \cup \{1\})$. Then $(n|A) \leq c(1|A)$. Therefore $n \in \Theta_{\mathbb{N}^{>0}}(1)$. \square

Theorem D.20 (Scale-invariance for Θ). *Let $X \in U$, $f \in R_X$, and $\alpha \in \mathbb{R}^{>0}$. Then $\Theta_X(\alpha f) = \Theta_X(f)$.*

Proof.

Appendix D. Proofs of local linear dominance properties

\subseteq

Assume $\widehat{f} \in \Theta_X(\alpha f)$. Then there exists $c \in \mathbb{R}^{>0}$ and $A \in \mathcal{F}(X)$, such that

$$\begin{aligned} (\widehat{f}|A) &\leq c((\alpha f)|A) \\ &= (c\alpha)(f|A). \end{aligned} \tag{D.21}$$

Therefore $\widehat{f} \in \Theta_X(f)$.

\supseteq

Assume $\widehat{f} \in \Theta_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$ and $A \in \mathcal{F}(X)$, such that

$$\begin{aligned} (\widehat{f}|A) &\leq c(f|A) \\ &= (c/\alpha)((\alpha f)|A). \end{aligned} \tag{D.22}$$

Therefore $\widehat{f} \in \Theta_X(\alpha f)$. □

Theorem D.23 (Sub-homogeneity for Θ). *Let $X \in U$, and $f, u \in R_X$. Then*

$$u \Theta_X(f) \subset \Theta_X(uf). \tag{D.24}$$

Proof. Let $\widehat{f} \in \Theta_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$ and $A \in \mathcal{F}(X)$, such that

$$(\widehat{f}|A) \leq c(f|A). \tag{D.25}$$

This implies

$$(u\widehat{f}|A) \leq c(uf|A). \tag{D.26}$$

Therefore $u\widehat{f} \in \Theta_X(uf)$. □

Theorem D.27 (Super-homogeneity for Θ characterized). *Θ_X has super-homogeneity. $\iff \mathcal{F}(X) = \{X\}$.*

Proof.

Appendix D. Proofs of local linear dominance properties

\implies

Suppose $\mathcal{F}(X) \neq \{X\}$. Then there exists $A \in \mathcal{F}(X)$ such that $A \neq X$. Let $\bar{A} = X \setminus A$, and $f \in R_X$ be such that $f = [A]_X$. Let $g \in R_X$, and $\hat{h} \in R_X$ be such that $\hat{h} = fg[A]_X + [\bar{A}]_X$. Then $(\hat{h}|A) \leq (fg|A)$, and so $\hat{h} \in \Theta_X(fg)$. Let $\hat{g} \in \Theta_X(g)$. Then $0 = (f\hat{g}|\bar{A}) \neq (\hat{h}|\bar{A}) = 1$; O_X does not have super-homogeneity.

\impliedby

Suppose $\mathcal{F}(X) = \{X\}$. Let $f, g \in R_X$, and $\hat{h} \in \Theta_X(fg)$. Then there exists $c \in \mathbb{R}^{>0}$ such that

$$\hat{h} \leq cfg. \quad (\text{D.28})$$

Let $F = \overleftarrow{f}(\mathbb{R}^{>0})$, and $\bar{F} = X \setminus F$. Let $\hat{g} \in R_X$ be such that $(\hat{g}|F) = (\hat{h}|F)/(f|F)$ and $(\hat{g}|\bar{F}) = 0$. Then $\hat{g} \leq cg$, and so $\hat{g} \in \Theta_X(g)$. In addition, $\hat{h} = f\hat{g}$. \square

Theorem D.29 (Super-multiplicativity for Θ). *Let $X \in U$, and $f, g \in R_X$. Then*

$$\Theta_X(f) \Theta_X(g) \supset \Theta_X(fg). \quad (\text{D.30})$$

Proof. Let $\hat{h} \in \Theta_X(fg)$. Then there exists $c \in \mathbb{R}^{>0}$ and $A \in \mathcal{F}(X)$, such that

$$(\hat{h}|A) \leq c(fg|A). \quad (\text{D.31})$$

Let $\hat{g} \in R_X$ be such that

$$\hat{g}(x) = \begin{cases} g(x), & x \in A, \\ 1, & x \notin A. \end{cases} \quad (\text{D.32})$$

Then $\hat{g} \in \Theta_X(g)$, since $(\hat{g}|A) = (g|A)$. Let $\hat{G} = \{x \in X : \hat{g}(x) > 0\}$, and let $\hat{f} \in R_X$ be such that

$$\hat{f}(x) = \begin{cases} \hat{h}(x)/\hat{g}(x), & x \in \hat{G} \\ 0, & x \notin \hat{G}. \end{cases} \quad (\text{D.33})$$

Appendix D. Proofs of local linear dominance properties

If $x \in \widehat{G}$, then $\widehat{h}(x) = \widehat{f}(x)\widehat{g}(x)$. If $x \notin \widehat{G}$, then $x \in A$, and $g(x) = 0 = \widehat{g}(x)$. Since $\widehat{h}(x) \leq cf(x)g(x)$, it follows that $\widehat{h}(x) = 0$. Therefore $\widehat{h}(x) = \widehat{f}(x)\widehat{g}(x)$, and $\widehat{h} = \widehat{f}\widehat{g}$. If $x \in A \setminus \widehat{G}$, then $\widehat{f}(x) = 0$, and $\widehat{f}(x) \leq cf(x)$. If $x \in A \cap \widehat{G}$, then $g(x) = \widehat{g}(x) > 0$, and $\widehat{f}(x) = \widehat{h}(x)/g(x) \leq cf(x)$. Therefore $\widehat{f} \in \Theta_X(f)$, and $\widehat{h} \in \Theta_X(f) \Theta_X(g)$. \square

Note D.34 (Super-multiplicativity in another way?). In proving super-multiplicativity, we cannot refer to [Super-multiplicativity is implied] (C.29), since there are local linear dominances which do not satisfy its assumptions. \blacksquare

Theorem D.35 (Sub-composability for Θ for fixed $s: Y \rightarrow X$). *Let $X, Y \in U$, and $s: Y \rightarrow X$. Then*

$$\begin{aligned} & \forall f \in R_X : \Theta_X(f) \circ s \subset \Theta_Y(f \circ s) \\ \iff & \forall A_X \in \mathcal{F}(X), \exists A_Y \in \mathcal{F}(Y) : \overrightarrow{s}(A_Y) \subset A_X. \end{aligned} \tag{D.36}$$

Proof. First notice that

$$\begin{aligned} & \overrightarrow{s}(A_Y) \subset A_X \\ \iff & A_Y \subset \overleftarrow{s}(A_X) \\ \iff & [A_Y]_Y \leq [\overleftarrow{s}(A_X)]_Y \\ \iff & [A_Y]_Y \leq [A_X]_X \circ s, \end{aligned} \tag{D.37}$$

for all $A_X \in \mathcal{F}(X)$, $A_Y \in \mathcal{F}(Y)$. We attempt to prove the statement in this form.

\implies

Let $A_X \in \mathcal{F}(X)$, $f = [A_X]_X$, and $\widehat{f} \in R_X$ be such that $\widehat{f} = 1$. Then $(\widehat{f}|A_X) \leq 1(f|A_X)$, and so $\widehat{f} \in \Theta_X(f)$. By assumption, $\widehat{f} \circ s = 1 \in \Theta_Y(f \circ s)$, and so there exists $d \in \mathbb{R}^{>0}$ and $A_Y \in \mathcal{F}(Y)$ such that

$$\begin{aligned} [A_Y]_Y & \leq [A_Y]_Y d(f \circ s) \\ & = [A_Y]_Y d([A_X]_X \circ s) \\ & \leq d([A_X]_X \circ s). \end{aligned} \tag{D.38}$$

Appendix D. Proofs of local linear dominance properties

Since the functions are indicator functions, this is equivalent to

$$[A_Y]_Y \leq [A_X]_X \circ s. \quad (\text{D.39})$$

\Leftarrow

Let $f \in R_X$ and $\hat{f} \in \Theta_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$ and $A_X \in \mathcal{F}(X)$ such that $[A_X]_X \hat{f} \leq [A_X]_X c f$. By assumption, there exists $A_Y \in \mathcal{F}(Y)$ such that $[A_Y]_Y \leq [A_X]_X \circ s$. Then

$$\begin{aligned} & [A_X]_X \hat{f} \leq [A_X]_X c f \\ \implies & ([A_X]_X \circ s) (\hat{f} \circ s) \leq ([A_X]_X \circ s) c (f \circ s) \\ \implies & [A_Y]_Y (\hat{f} \circ s) \leq [A_Y]_Y c (f \circ s) \\ \implies & (\hat{f} \circ s) \in \Theta_Y(f \circ s). \end{aligned} \quad (\text{D.40})$$

□

Theorem D.41 (Characterization of sub-composability for Θ). Θ has sub-composability if and only if

$$(\forall X \in U : \mathcal{F}(X) = \{X\}) \text{ or } (\forall X \in U : \emptyset \in \mathcal{F}(X)). \quad (\text{D.42})$$

Proof.

\implies

Suppose there exists $X \in U$, such that $\emptyset \notin \mathcal{F}(X)$ and $\mathcal{F}(X) \neq \{X\}$. Let $A \in \mathcal{F}(X)$, $x^* \in X \setminus A$, and $s: X \rightarrow X$ be such that $s(x) = x^*$. Then $\vec{s}(X) \cap A = \emptyset$, and Θ does not have sub-composability under s by [Sub-composability for Θ for fixed $s: Y \rightarrow X$] (D.35). This shows that either $\mathcal{F}(X) = \{X\}$, or $\emptyset \in \mathcal{F}(X)$, for all $X \in U$.

Suppose there exists $X, Y \in U$, such that $\emptyset \in \mathcal{F}(X)$, and $\mathcal{F}(Y) = \{Y\}$. Then sub-composability does not hold for any $s: Y \rightarrow X$, by [Sub-composability for Θ for fixed $s: Y \rightarrow X$] (D.35), because $\vec{s}(Y) \not\subset \emptyset$.

\Leftarrow

Let $X, Y \in U$, and $s: Y \rightarrow X$. Either $\emptyset \in \mathcal{F}(X)$ and $\emptyset \in \mathcal{F}(Y)$, or $\mathcal{F}(X) = \{X\}$

Appendix D. Proofs of local linear dominance properties

and $\mathcal{F}(Y) = \{Y\}$. In either case, sub-composability holds by [Sub-composability for Θ for fixed $s: Y \rightarrow X$] (D.35). \square

Theorem D.43 (Sub-composability for Θ for positive functions).
 $\forall X \in U, \forall A \in \mathcal{F}(X) : |X \setminus A| < \infty \implies \Theta \text{ has sub-composability for positive functions .}$

Proof. Let $\hat{f} \in \Theta_X(f)$, where $f \in R_X^{>0}$. Then there exists $A \in \mathcal{F}(X)$ and $c \in \mathbb{R}^{>0}$, such that

$$\begin{aligned} (\hat{f}|A) &\leq c(f|A) \\ \iff \frac{(\hat{f}|A)}{(f|A)} &\leq c. \end{aligned} \tag{D.44}$$

Since $|X \setminus A| < \infty$, let

$$d = \max \left\{ \frac{\hat{f}|(X \setminus A)}{f|(X \setminus A)} \right\} \cup \{c\}. \tag{D.45}$$

Then $d \in \mathbb{R}^{>0}$, and

$$\begin{aligned} \frac{\hat{f}}{f} &\leq d \\ \iff \hat{f} &\leq df. \end{aligned} \tag{D.46}$$

Let $s: Y \rightarrow X$, where $Y \in U$. Then

$$(\hat{f} \circ s) \leq d(f \circ s). \tag{D.47}$$

Therefore $(\hat{f} \circ s) \in \Theta_X(f \circ s)$. \square

Theorem D.48 (Characterization of Extensibility for Θ). Θ_X has extensibility if and only if

$$\forall A \in \mathcal{F}(X) : \exists B \in \mathcal{F}(X \times Y) : \overrightarrow{[X \times Y]}(B) \subset A. \tag{D.49}$$

Proof. This follows directly from [Sub-composability for Θ for fixed $s: Y \rightarrow X$] (D.35). \square

Appendix D. Proofs of local linear dominance properties

Theorem D.50 (Power-homogeneity for Θ). *Let $X \in U$, $f \in R_X$, and $\alpha \in \mathbb{R}^{>0}$. Then*

$$\Theta_X(f)^\alpha = \Theta_X(f^\alpha). \quad (\text{D.51})$$

Proof.

\subseteq

Let $\hat{f} \in \Theta_X(f)$. Then there exists $A \in \mathcal{F}(X)$, and $c \in \mathbb{R}^{>0}$, such that

$$\left(\hat{f}|A\right) \leq c(f|A). \quad (\text{D.52})$$

Then

$$\left(\hat{f}|A\right)^\alpha \leq c^\alpha (f|A)^\alpha. \quad (\text{D.53})$$

Therefore $\hat{f}^\alpha \in \Theta_X(f^\alpha)$.

\supseteq

Let $\hat{f} \in \Theta_X(f^\alpha)$. Then there exists $A \in \mathcal{F}(X)$, and $c \in \mathbb{R}^{>0}$, such that

$$\left(\hat{f}|A\right) \leq c(f^\alpha|A). \quad (\text{D.54})$$

Then

$$\left(\hat{f}|A\right)^{1/\alpha} \leq c(f|A). \quad (\text{D.55})$$

Let $g \in R_X$ be such that $g = \hat{f}^{1/\alpha}$. Then $g \in \Theta_X(f)$, and $g^\alpha = \hat{f}$. \square

Note D.56 (Power-homogeneity from primitive properties). We would rather want to prove power-homogeneity from primitive properties, as done for other non-primitive properties in Appendix C. However, we were unable to come up with such a proof. \blacksquare

Theorem D.57 (Zero-triviality for Θ characterized). Θ_X has zero-triviality if and only if $\mathcal{F}(X) = \{X\}$.

Proof.

Appendix D. Proofs of local linear dominance properties

\implies

Suppose there exists $A \in \mathcal{F}(X)$ such that $A \neq X$. Then $[X \setminus A]_X \in \mathcal{O}_X(0)$ and zero-triviality does not hold.

\impliedby

Suppose $\mathcal{F}(X) = \{X\}$, and $f \in \mathcal{O}_X(0)$. Then there exists $c \in \mathbb{R}^{>0}$, such that

$$f \leq c0. \quad (\text{D.58})$$

Therefore $f = 0$. \square

Theorem D.59 (Θ is almost equal to \mathcal{O} for cofinite filter sets).
Suppose $|X \setminus A| < \infty$, for all $A \in \mathcal{F}(X)$. Then

$$\Theta_X(g) = \mathcal{O}_X(g), \quad (\text{D.60})$$

for all $g \in R_X^{>0}$.

Proof.

\subseteq

Suppose $f \in \Theta_X(g)$. Then there exists $A \in \mathcal{F}(X)$ and $c \in \mathbb{R}^{>0}$, such that

$$\begin{aligned} (f|A) &\leq c(g|A) \\ \iff \frac{(f|A)}{(g|A)} &\leq c. \end{aligned} \quad (\text{D.61})$$

Since $X \setminus A$ is finite, let

$$d = \max\left(\frac{f}{g}\right)(X \setminus A) \cup \{c\}. \quad (\text{D.62})$$

Then

$$\begin{aligned} \frac{f}{g} &\leq d \\ \iff f &\leq dg. \end{aligned} \quad (\text{D.63})$$

Therefore $f \in \mathcal{O}_X(g)$.

Appendix D. Proofs of local linear dominance properties

\supseteq

Suppose $f \in \mathcal{O}_X(g)$. Then there exists $c \in \mathbb{R}^{>0}$, such that

$$f \leq cg. \tag{D.64}$$

Let $A \in \mathcal{F}(X)$. Then

$$(f|A) \leq c(g|A). \tag{D.65}$$

Therefore $f \in \mathcal{O}_X(g)$. \square

Appendix E

Proofs of limit theorems

In this section we prove the limit theorems for local linear dominance.

Theorem E.1 (Relation between ratio-limits).

$$\limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} = 1 / \liminf_{\mathcal{F}(F)} \frac{(g|F)}{(f|F)}, \quad (\text{E.2})$$

for all $f, g \in R_X$, where $F = \overleftarrow{f}(\mathbb{R}^{>0})$.

Proof.

Notation

Let

$$c := \limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)}, \quad (\text{E.3})$$

and

$$d := \liminf_{\mathcal{F}(F)} \frac{(g|F)}{(f|F)}. \quad (\text{E.4})$$

$$\underline{c = \infty \iff d = 0}$$

Appendix E. Proofs of limit theorems

$$\begin{aligned}
& c = \infty \\
\iff & \forall A \in \mathcal{F}(F) : \sup \left(\overrightarrow{\frac{f|A}{g|A}} \right)(A) = \infty \\
\iff & \forall A \in \mathcal{F}(F) : \exists x \in A : g(x) = 0 \\
\iff & \forall A \in \mathcal{F}(F) : \inf \left(\overrightarrow{\frac{g|A}{f|A}} \right)(A) = 0 \\
\iff & d = 0.
\end{aligned} \tag{E.5}$$

Therefore, if $c = \infty$, or $d = 0$, we have that

$$c = 1/d. \tag{E.6}$$

$$\underline{c = 0} \implies \underline{d = \infty}$$

Suppose $c = 0$. For any $\varepsilon \in \mathbb{R}^{>0}$, there exists $A \in \mathcal{F}(F)$, such that

$$\begin{aligned}
& \sup \left(\overrightarrow{\frac{f|A}{g|A}} \right)(A) \leq \varepsilon \\
\implies & \frac{(f|A)}{(g|A)} \leq \varepsilon \\
\implies & \frac{(g|A)}{(f|A)} \geq \frac{1}{\varepsilon} \\
\implies & \inf \left(\overrightarrow{\frac{g|A}{f|A}} \right)(A) \geq \frac{1}{\varepsilon}
\end{aligned} \tag{E.7}$$

Therefore,

$$d \geq \frac{1}{\varepsilon}. \tag{E.8}$$

Since this holds for all ε ,

$$d = \infty. \tag{E.9}$$

Therefore, if $c = 0$, then $d = \infty$, and

$$c = 1/d. \tag{E.10}$$

$$\underline{c = 0} \iff \underline{d = \infty}$$

Appendix E. Proofs of limit theorems

Suppose $d = \infty$. For $\varepsilon \in \mathbb{R}^{>0}$, there exists $A \in \mathcal{F}(F)$, such that

$$\begin{aligned}
 & \inf \overrightarrow{\left(\frac{g|A}{f|A} \right)}(A) \geq \varepsilon \\
 \implies & \frac{(g|A)}{(f|A)} \geq \varepsilon \\
 \implies & \frac{(f|A)}{(g|A)} \leq \frac{1}{\varepsilon} \\
 \implies & \sup \overrightarrow{\left(\frac{f|A}{g|A} \right)}(A) \leq \frac{1}{\varepsilon}.
 \end{aligned} \tag{E.11}$$

Therefore

$$c \leq \frac{1}{\varepsilon}. \tag{E.12}$$

Since this holds for all ε ,

$$c = 0. \tag{E.13}$$

Therefore, if $d = \infty$, then $c = 0$, and

$$c = 1/d. \tag{E.14}$$

$c \geq 1/d$

Suppose $c, d \in \mathbb{R}^{>0}$. By definition, for any $\varepsilon \in \mathbb{R}^{>0}$, there exists $A \in \mathcal{F}(F)$, such that

$$\begin{aligned}
 & \sup \overrightarrow{\left(\frac{f|A}{g|A} \right)}(A) - c \leq \varepsilon \\
 \implies & \frac{(f|A)}{(g|A)} - c \leq \varepsilon \\
 \implies & \frac{(f|A)}{(g|A)} \leq c + \varepsilon \\
 \implies & \frac{(g|A)}{(f|A)} \geq \frac{1}{c + \varepsilon} \\
 \implies & \inf \overrightarrow{\left(\frac{g|A}{f|A} \right)}(A) \geq \frac{1}{c + \varepsilon}.
 \end{aligned} \tag{E.15}$$

Therefore

$$d \geq \frac{1}{c + \varepsilon}. \tag{E.16}$$

Appendix E. Proofs of limit theorems

Since this holds for all ε ,

$$\begin{aligned} d &\geq \frac{1}{c} \\ \implies c &\geq 1/d. \end{aligned} \tag{E.17}$$

$c \leq 1/d$

Suppose $c, d \in \mathbb{R}^{>0}$. By definition, for any $\varepsilon \in \mathbb{R}^{>0}$ such that $\varepsilon < d$, there exists $A \in \mathcal{F}(F)$, such that

$$\begin{aligned} d - \inf \left(\overrightarrow{\frac{g|A}{f|A}} \right)(A) &\leq \varepsilon \\ \implies d - \frac{(g|A)}{(f|A)} &\leq \varepsilon \\ \implies d - \varepsilon &\leq \frac{(g|A)}{(f|A)} \\ \implies \frac{1}{d - \varepsilon} &\geq \frac{(f|A)}{(g|A)} \\ \implies \frac{1}{d - \varepsilon} &\geq \sup \left(\overrightarrow{\frac{f|A}{g|A}} \right)(A). \end{aligned} \tag{E.18}$$

Therefore

$$c \leq \frac{1}{d - \varepsilon}. \tag{E.19}$$

Since this holds for all ε ,

$$c \leq \frac{1}{d}. \tag{E.20}$$

□

Theorem E.21 (Θ by a limit).

$$\limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} < \infty \iff f \in \Theta_X(g), \tag{E.22}$$

for all $f, g \in R_X$, where $F = \overleftarrow{f}(\mathbb{R}^{>0})$.

Proof.

Appendix E. Proofs of limit theorems

Notation

Let

$$d := \limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)}. \quad (\text{E.23})$$

\implies

Suppose $d < \infty$. By definition, for any $\varepsilon \in \mathbb{R}^{>0}$, there exists $A \in \mathcal{F}(F)$, such that $(g|A) > 0$, and

$$\begin{aligned} & \sup \left(\overrightarrow{\frac{f|A}{g|A}} \right) (A) - d \leq \varepsilon \\ \implies & \frac{(f|A)}{(g|A)} - d \leq \varepsilon \\ \implies & (f|A) \leq (d + \varepsilon)(g|A). \end{aligned} \quad (\text{E.24})$$

Therefore $(f|F) \in \Theta_F(g|F)$. By order-consistency, $(f|\bar{F}) \in \Theta_F(g|\bar{F})$, where $\bar{F} = X \setminus F$. By locality, $f \in \Theta_X(g)$.

\Leftarrow

Suppose $f \in \Theta_X(g)$. By restrictability, $(f|F) \in \Theta_F(g|F)$. Therefore, there exists $A \in \mathcal{F}(F)$ and $c \in \mathbb{R}^{>0}$, such that

$$(f|A) \leq c(g|A) \quad (\text{E.25})$$

In particular, $(g|A) > 0$. Therefore

$$\begin{aligned} & \frac{(f|A)}{(g|A)} \leq c \\ \implies & \sup \left(\overrightarrow{\frac{f|A}{g|A}} \right) (A) \leq c. \end{aligned} \quad (\text{E.26})$$

Therefore

$$d \leq c < \infty. \quad (\text{E.27})$$

□

Appendix E. Proofs of limit theorems

Theorem E.28 (\mathfrak{Q} by a limit).

$$\liminf_{\mathcal{F}(G)} \frac{(f|G)}{(g|G)} > 0 \iff f \in \mathfrak{Q}_X(g), \quad (\text{E.29})$$

for all $f, g \in R_X$, where $G = \overleftarrow{g}(\mathbb{R}^{>0})$.

Proof. This follows from $[\mathfrak{O}$ by a limit] (E.21) and $[\text{Relation between ratio-limits}]$ (E.1). \square

Theorem E.30 (\mathfrak{e} by a limit).

$$\left(\limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} < \infty \text{ and } \liminf_{\mathcal{F}(G)} \frac{(f|G)}{(g|G)} = 0 \right) \iff f \in \mathfrak{e}_X(g), \quad (\text{E.31})$$

for all $f, g \in R_X$, where $F = \overleftarrow{f}(\mathbb{R}^{>0})$ and $G = \overleftarrow{g}(\mathbb{R}^{>0})$.

Proof. By definition,

$$\begin{aligned} & f \in \mathfrak{e}_X(g) \\ \iff & f \in \mathfrak{O}_X(g) \text{ and } \neg(g \in \mathfrak{O}_X(f)) \\ \iff & f \in \mathfrak{O}_X(g) \text{ and } \neg(f \in \mathfrak{Q}_X(g)). \end{aligned} \quad (\text{E.32})$$

The first term is obtained by $[\mathfrak{O}$ by a limit] (E.21), and the second term is obtained by $[\mathfrak{Q}$ by a limit] (E.28). \square

Theorem E.33 (Traditional \mathfrak{o}_X by a limit).

$$\begin{aligned} & \forall \varepsilon \in \mathbb{R}^{>0} : \exists A \in \mathcal{F}(X) : (f|A) \leq \varepsilon(g|A) \\ \iff & \limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} = 0, \end{aligned} \quad (\text{E.34})$$

for all $f, g \in R_X$, where $F = \overleftarrow{f}(\mathbb{R}^{>0})$.

Appendix E. Proofs of limit theorems

Proof. It holds that

$$\begin{aligned}
 & \forall \varepsilon \in \mathbb{R}^{>0} : \exists A \in \mathcal{F}(X) : (f|A) \leq \varepsilon(g|A) \\
 \iff & \forall \varepsilon \in \mathbb{R}^{>0} : \exists A \in \mathcal{F}(F) : (f|A) \leq \varepsilon(g|A) \\
 \iff & \forall \varepsilon \in \mathbb{R}^{>0} : \exists A \in \mathcal{F}(F) : \frac{(f|A)}{(g|A)} \leq \varepsilon \\
 \iff & \forall \varepsilon \in \mathbb{R}^{>0} : \exists A \in \mathcal{F}(F) : \sup \overrightarrow{\frac{(f|A)}{(g|A)}}(A) \leq \varepsilon \tag{E.35} \\
 \iff & \forall \varepsilon \in \mathbb{R}^{>0} : \limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} \leq \varepsilon \\
 \iff & \limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} = 0.
 \end{aligned}$$

□

Theorem E.36 ($\mathring{\omega}$ by a limit).

$$\left(\limsup_{\mathcal{F}(F)} \frac{(f|F)}{(g|F)} = \infty \text{ and } \liminf_{\mathcal{F}(G)} \frac{(f|G)}{(g|G)} > 0 \right) \iff f \in \mathring{\omega}_X(g), \tag{E.37}$$

for all $f, g \in R_X$, where $F = \overleftarrow{f}(\mathbb{R}^{>0})$ and $G = \overleftarrow{g}(\mathbb{R}^{>0})$.

Proof. Similarly to [$\mathring{\bullet}$ by a limit] (E.30). □

Appendix F

Proofs of Master theorems

In this section we will show that various Master theorems hold for the Θ -notation as defined by linear dominance. The theorems are simpler for linear dominance than for asymptotic linear dominance; there are no “regularity” requirements (see [2]).

F.1 Master theorem over powers

Definition F.1 (Master function over powers). Let $a \in \mathbb{R}^{\geq 1}$, $b \in \mathbb{R}^{>1}$, $d \in \mathbb{R}^{>0}$, $B = \{b^i : i \in \mathbb{N}\}$, and $F \in R_B$. A *master function over powers* is a function $T \in R_B$ defined by the recurrence equation

$$T(n) = \begin{cases} aT(n/b) + F(n), & n \geq b, \\ d, & n < b. \end{cases} \quad (\text{F.2})$$

The set of such functions is denoted by $\mathcal{M}_P(a, b, d, F)$.

Theorem F.3 (Logarithm swap). Let $x, y, b \in \mathbb{R}^{>0}$ be such that $b \neq 1$. Then

$$x^{\log_b(y)} = y^{\log_b(x)}. \quad (\text{F.4})$$

Appendix F. Proofs of Master theorems

Proof.

$$\begin{aligned}
 x^{\log_b(y)} &= \left(b^{\log_b(x)}\right)^{\log_b(y)} \\
 &= b^{\log_b(x) \log_b(y)} \\
 &= \left(b^{\log_b(y)}\right)^{\log_b(x)} \\
 &= y^{\log_b(x)}.
 \end{aligned} \tag{F.5}$$

□

Theorem F.6 (Explicit form for a Master function over powers).

Let $T \in \mathcal{M}_P(a, b, d, F)$ be a Master function over powers. Then

$$T(n) = n^{\log_b(a)}d + \sum_{i=0}^{\log_b(n)-1} a^i F(n/b^i). \tag{F.7}$$

Proof.

Pattern

Expanding the recurrence, we find that

$$\begin{aligned}
 T(1) &= d, \\
 T(b) &= ad + F(b), \\
 T(b^2) &= a(ad + F(b)) + F(b^2) \\
 &= a^2d + aF(b) + F(b^2), \\
 T(b^3) &= a(a^2d + aF(b) + F(b^2)) + F(b^3) \\
 &= a^3d + a^2F(b) + aF(b^2) + F(b^3).
 \end{aligned} \tag{F.8}$$

Induction

This suggests the pattern

$$T(b^m) = a^m d + \sum_{i=0}^{m-1} a^i F(b^{m-i}), \tag{F.9}$$

Appendix F. Proofs of Master theorems

where $m \in \mathbb{N}$. We prove this by induction. For the base case, $T(1) = T(b^0) = d$. For the induction step, if $m > 0$, then

$$\begin{aligned}
 T(b^m) &= aT(b^{m-1}) + F(b^m) \\
 &= a \left(a^{m-1}d + \sum_{i=0}^{m-2} a^i F(b^{m-1-i}) \right) + F(b^m) \\
 &= a^m d + \sum_{i=0}^{m-2} a^{i+1} F(b^{m-1-i}) + F(b^m) \\
 &= a^m d + \sum_{i=1}^{m-1} a^i F(b^{m-i}) + F(b^m) \\
 &= a^m d + \sum_{i=0}^{m-1} a^i F(b^{m-i}).
 \end{aligned} \tag{F.10}$$

If $n = b^m$, then $m = \log_b(n)$, and

$$\begin{aligned}
 T(n) &= a^{\log_b(n)} d + \sum_{i=0}^{\log_b(n)-1} a^i F(n/b^i) \\
 &= n^{\log_b(a)} d + \sum_{i=0}^{\log_b(n)-1} a^i F(n/b^i),
 \end{aligned} \tag{F.11}$$

where the last step is by [Logarithm swap] (F.3). □

Theorem F.12 (Difference of powers). *Suppose $\alpha, \beta \in \mathbb{R}^{\geq 0}$ are such that $\alpha < \beta$. Then*

$$(n^\beta - n^\alpha) \in \Theta_B(n^\beta - 1). \tag{F.13}$$

Proof. Since $\alpha < \beta$, we have that $(1 - b^{\alpha-\beta}) > 0$. Then

$$\begin{aligned}
 (n^\beta - 1)(1 - b^{\alpha-\beta}) &\leq (n^\beta - 1)(1 - n^{\alpha-\beta}) \\
 &\leq n^\beta(1 - n^{\alpha-\beta}) \\
 &= n^\beta - n^\alpha \\
 &\leq n^\beta - 1.
 \end{aligned} \tag{F.14}$$

□

Appendix F. Proofs of Master theorems

Lemma F.15 (Master summation lemma). *Let $a \in \mathbb{R}^{\geq 1}$, $b \in \mathbb{R}^{> 1}$, $c \in \mathbb{R}^{\geq 0}$, and $B = \{b^i : i \in \mathbb{N}\}$. Let $S \in R_B$ be such that*

$$S(n) = n^c \sum_{i=0}^{\log_b(n)-1} (a/b^c)^i. \quad (\text{F.16})$$

Then

$$\begin{aligned} \log_b(a) < c &\implies S \in \Theta_B(n^c - 1), \\ \log_b(a) = c &\implies S \in \Theta_B(n^c \log_b(n)), \\ \log_b(a) > c &\implies S \in \Theta_B(n^{\log_b(a)} - 1). \end{aligned} \quad (\text{F.17})$$

Proof.

$$\underline{\log_b(a) = c}$$

This implies $a/b^c = 1$. Then

$$n^c \sum_{i=0}^{\log_b(n)-1} (a/b^c)^i = n^c \log_b(n). \quad (\text{F.18})$$

Therefore $S \in \Theta_B(n^c \log_b(n))$.

$$\underline{\log_b(a) \neq c}$$

Suppose $\log_b(a) \neq c$, and let $\gamma = \frac{1}{(a/b^c)-1}$. Then

$$\begin{aligned} n^c \sum_{i=0}^{\log_b(n)-1} (a/b^c)^i &= n^c \frac{(a/b^c)^{\log_b(n)} - 1}{(a/b^c) - 1} \\ &= \gamma n^c (a^{\log_b(n)} / n^c - 1) \\ &= \gamma (a^{\log_b(n)} - n^c) \\ &= \gamma (n^{\log_b(a)} - n^c). \end{aligned} \quad (\text{F.19})$$

Therefore

$$S(n) = \gamma (n^{\log_b(a)} - n^c). \quad (\text{F.20})$$

$$\underline{\log_b(a) > c}$$

Appendix F. Proofs of Master theorems

This implies $\gamma > 0$. Then $S \in \Theta_B(n^{\log_b(a)} - 1)$ by [Difference of powers] (F.12) and scale-invariance.

$\log_b(a) < c$

This implies $\gamma < 0$. Then $S \in \Theta_B(n^c - 1)$ by [Difference of powers] (F.12) and scale-invariance. \square

Theorem F.21 (Master theorem over powers). *Let $T \in \mathcal{M}_P(a, b, d, F)$ be a Master function over powers, and $F \in \Theta_B(n^c)$, where $c \in \mathbb{R}^{\geq 0}$. Then*

$$\begin{aligned} \log_b(a) < c &\implies T \in \Theta_B(n^c), \\ \log_b(a) = c &\implies T \in \Theta_B(n^c \log_b(bn)), \\ \log_b(a) > c &\implies T \in \Theta_B(n^{\log_b(a)}). \end{aligned} \tag{F.22}$$

If $F \in \Theta_B(n^c)$, then each Θ_B can be replaced with Θ_B .

Proof.

Explicit forms

By [Explicit form for a Master function over powers] (F.6),

$$T(n) = n^{\log_b(a)}d + \sum_{i=0}^{\log_b(n)-1} a^i F(n/b^i), \tag{F.23}$$

Let $S \in R_B$ be such that

$$S(n) = n^c \sum_{i=0}^{\log_b(n)-1} (a/b^c)^i. \tag{F.24}$$

Since $F \in \Theta_B(n^c)$, by subset-summability

$$\begin{aligned} \Theta_B\left(\sum_{i=0}^{\log_b(n)-1} a^i F(n/b^i)\right) &\subset \Theta_B\left(\sum_{i=0}^{\log_b(n)-1} a^i (n/b^i)^c\right) \\ &= \Theta_B\left(n^c \sum_{i=0}^{\log_b(n)-1} (a/b^c)^i\right) \\ &= \Theta_B(S). \end{aligned} \tag{F.25}$$

Appendix F. Proofs of Master theorems

$\log_b(a) < c$

By [Master summation lemma] (F.15),

$$\Theta_B(S) = \Theta_B(n^c - 1). \quad (\text{F.26})$$

By additivity, scale-invariance, and since $\log_b(a) < c$,

$$\begin{aligned} \Theta_B(T) &= \Theta_B\left(n^{\log_b(a)}\right) + \Theta_B\left(\sum_{i=0}^{\log_b(n)-1} a^i F(n/b^i)\right) \\ &\subset \Theta_B\left(n^{\log_b(a)}\right) + \Theta_B(n^c - 1) \\ &= \Theta_B\left(n^{\log_b(a)} + n^c - 1\right) \\ &= \Theta_B(n^c). \end{aligned} \quad (\text{F.27})$$

$\log_b(a) = c$

By [Master summation lemma] (F.15),

$$\Theta_B(S) = \Theta_B(n^c \log_b(n)). \quad (\text{F.28})$$

By additivity and scale-invariance,

$$\begin{aligned} \Theta_B(T) &= \Theta_B(n^c) + \Theta_B\left(\sum_{i=0}^{\log_b(n)-1} a^i F(n/b^i)\right) \\ &\subset \Theta_B(n^c) + \Theta_B(n^c \log_b(n)) \\ &= \Theta_B(n^c + n^c \log_b(n)) \\ &= \Theta_B(n^c \log_b(bn)). \end{aligned} \quad (\text{F.29})$$

$\log_b(a) > c$

By [Master summation lemma] (F.15),

$$\Theta_B(S) = \Theta_B\left(n^{\log_b(a)} - 1\right). \quad (\text{F.30})$$

Appendix F. Proofs of Master theorems

By additivity and scale-invariance,

$$\begin{aligned}
 \Theta_B(T) &= \Theta_B\left(n^{\log_b(a)}\right) + \Theta_B\left(\sum_{i=0}^{\log_b(n)-1} a^i F(n/b^i)\right) \\
 &\subset \Theta_B\left(n^{\log_b(a)}\right) + \Theta_B\left(n^{\log_b(a)} - 1\right) \\
 &= \Theta_B\left(2n^{\log_b(a)} - 1\right) \\
 &= \Theta_B\left(n^{\log_b(a)}\right).
 \end{aligned} \tag{F.31}$$

By order-consistency, $\Theta_B(n^{\log_b(a)}) \subset \Theta_B(T)$. Therefore $\Theta_B(T) = \Theta_B(n^{\log_b(a)})$.

$F \in \Theta_B(n^c)$

Suppose we know that $F \in \Theta_B(n^c)$. Then by subset-summability

$$\Theta_B\left(\sum_{i=0}^{\log_b(n)-1} a^i F(n/b^i)\right) = \Theta_B(S), \tag{F.32}$$

and we get rid of subsets in the above proofs. □

F.2 Master theorem over reals

Definition F.33 (Master function over reals). Let $a \in \mathbb{R}^{\geq 1}$, $b \in \mathbb{R}^{> 1}$, $d \in \mathbb{R}^{> 0}$, and $f \in R_{\mathbb{R}^{\geq 1}}$. A *master function over reals* is a function $t \in R_{\mathbb{R}^{\geq 1}}$ defined by the recurrence equation

$$t(x) = \begin{cases} at(x/b) + f(x), & x \geq b, \\ d, & x < b. \end{cases} \tag{F.34}$$

The set of such functions is denoted by $\mathcal{M}_R(a, b, d, f)$.

Theorem F.35 (Explicit form for a Master function over reals).

Appendix F. Proofs of Master theorems

Let $t \in \mathcal{M}_R(a, b, d, f)$ be a Master function over reals. Then

$$t(x) = a^{\lfloor \log_b(x) \rfloor} d + \sum_{i=0}^{\lfloor \log_b(x) \rfloor - 1} a^i f(x/b^i). \quad (\text{F.36})$$

Proof.

Pattern

Suppose $b^0 \leq x < b^1$. Then

$$t(x) = d. \quad (\text{F.37})$$

Suppose $b^1 \leq x < b^2$. Then

$$t(x) = ad + f(x). \quad (\text{F.38})$$

Suppose $b^2 \leq x < b^3$. Then

$$\begin{aligned} t(x) &= a(ad + f(x/b)) + f(x) \\ &= a^2d + af(x/b) + f(x). \end{aligned} \quad (\text{F.39})$$

Suppose $b^3 \leq x < b^4$. Then

$$\begin{aligned} t(x) &= a(a^2d + af(x/b^2) + f(x/b)) + f(x) \\ &= a^3d + a^2f(x/b^2) + af(x/b) + f(x). \end{aligned} \quad (\text{F.40})$$

Induction

This suggests the pattern

$$t(x) = a^m d + \sum_{i=0}^{m-1} a^i f(x/b^i), \quad (\text{F.41})$$

where $m \in \mathbb{N}$ is such that $b^m \leq x < b^{m+1}$, which is equivalent to $m = \lfloor \log_b(x) \rfloor$. That is,

$$t(x) = a^{\lfloor \log_b(x) \rfloor} d + \sum_{i=0}^{\lfloor \log_b(x) \rfloor - 1} a^i f(x/b^i). \quad (\text{F.42})$$

This can be proved by induction, as with the analogous theorem for powers. \square

Appendix F. Proofs of Master theorems

Theorem F.43 (Master reduction from reals to powers). *Suppose $T \in \mathcal{M}_P(a, b, d, F)$ is a Master function over powers, and $t \in \mathcal{M}_R(a, b, d, f)$ is a Master function over reals. Then*

$$\begin{aligned} f \in \mathfrak{O}_{\mathbb{R} \geq 1} \left(F(b^{\lfloor \log_b(x) \rfloor}) \right) &\implies t \in \mathfrak{O}_{\mathbb{R} \geq 1} \left(T(b^{\lfloor \log_b(x) \rfloor}) \right), \\ f \in \mathfrak{O}_{\mathbb{R} \geq 1} \left(F(b^{\lfloor \log_b(x) \rfloor}) \right) &\implies t \in \mathfrak{O}_{\mathbb{R} \geq 1} \left(T(b^{\lfloor \log_b(x) \rfloor}) \right). \end{aligned} \quad (\text{F.44})$$

Proof.

Explicit forms

We have that

$$\begin{aligned} T(b^{\lfloor \log_b(x) \rfloor}) &= a^{\lfloor \log_b(x) \rfloor} d + \sum_{i=0}^{\lfloor \log_b(x) \rfloor - 1} a^i F(b^{\lfloor \log_b(x) \rfloor} / b^i), \\ t(x) &= a^{\lfloor \log_b(x) \rfloor} d + \sum_{i=0}^{\lfloor \log_b(x) \rfloor - 1} a^i f(x/b^i), \end{aligned} \quad (\text{F.45})$$

by [Explicit form for a Master function over powers] (F.6) and [Explicit form for a Master function over reals] (F.35).

\mathfrak{O} -sets

Note that

$$b^{\lfloor \log_b(x) \rfloor} / b^i = b^{\lfloor \log_b(x/b^i) \rfloor}. \quad (\text{F.46})$$

By additivity and subset-summability,

$$\begin{aligned} \mathfrak{O}_{\mathbb{R} \geq 1}(t) &= \mathfrak{O}_{\mathbb{R} \geq 1} \left(a^{\lfloor \log_b(x) \rfloor} d \right) + \mathfrak{O}_{\mathbb{R} \geq 1} \left(\sum_{i=0}^{\lfloor \log_b(x) \rfloor - 1} a^i f(x/b^i) \right) \\ &\subset \mathfrak{O}_{\mathbb{R} \geq 1} \left(a^{\lfloor \log_b(x) \rfloor} d \right) + \mathfrak{O}_{\mathbb{R} \geq 1} \left(\sum_{i=0}^{\lfloor \log_b(x) \rfloor - 1} a^i F(b^{\lfloor \log_b(x/b^i) \rfloor}) \right) \\ &= \mathfrak{O}_{\mathbb{R} \geq 1} \left(T(b^{\lfloor \log_b(x) \rfloor}) \right). \end{aligned} \quad (\text{F.47})$$

The proof for \mathfrak{O} -sets is similar. □

Appendix F. Proofs of Master theorems

Lemma F.48 (Identity equivalent).

$$x \in \Theta_{\mathbb{R} \geq 1} \left(b^{\lfloor \log_b(x) \rfloor} \right). \quad (\text{F.49})$$

Proof.

$$\begin{aligned} b^{\lfloor \log_b(x) \rfloor} &\leq b^{\log_b(x)} \\ &= x \\ &\leq b^{\lfloor \log_b(x) \rfloor} b, \end{aligned} \quad (\text{F.50})$$

for all $x \in \mathbb{R}^{\geq 1}$. □

Theorem F.51 (Master theorem over reals). *Let $t \in \mathcal{M}_R(a, b, d, f)$ be a Master function over reals, and $f \in \Theta_{\mathbb{R} \geq 1}(x^c)$, where $c \in \mathbb{R}^{\geq 0}$. Then*

$$\begin{aligned} \log_b(a) < c &\implies t \in \Theta_{\mathbb{R} \geq 1}(x^c), \\ \log_b(a) = c &\implies t \in \Theta_{\mathbb{R} \geq 1}(x^c \log_b(bx)), \\ \log_b(a) > c &\implies t \in \Theta_{\mathbb{R} \geq 1}(x^{\log_b(a)}). \end{aligned} \quad (\text{F.52})$$

If $f \in \Theta_{\mathbb{R} \geq 1}(n^c)$, then each $\Theta_{\mathbb{R} \geq 1}$ can be replaced with $\Theta_{\mathbb{R} \geq 1}$.

Proof.

Reduction to powers

Let $T \in \mathcal{M}_P(a, b, d, F)$ be a Master function over powers, where $F(n) = n^c$. By [Identity equivalent] (F.48),

$$\Theta_{\mathbb{R} \geq 1}(x) = \Theta_{\mathbb{R} \geq 1} \left(b^{\lfloor \log_b(x) \rfloor} \right). \quad (\text{F.53})$$

By power-homogeneity,

$$\begin{aligned} \Theta_{\mathbb{R} \geq 1}(x^c) &= \Theta_{\mathbb{R} \geq 1} \left(b^{\lfloor \log_b(x) \rfloor c} \right) \\ &= \Theta_{\mathbb{R} \geq 1} \left(F(b^{\lfloor \log_b(x) \rfloor}) \right). \end{aligned} \quad (\text{F.54})$$

Since $f \in \Theta_{\mathbb{R} \geq 1}(x^c)$ by assumption,

$$f \in \Theta_{\mathbb{R} \geq 1} \left(F(b^{\lfloor \log_b(x) \rfloor}) \right). \quad (\text{F.55})$$

Appendix F. Proofs of Master theorems

By [Master reduction from reals to powers] (F.43),

$$\Theta_{\mathbb{R} \geq 1}(t) \subset \Theta_{\mathbb{R} \geq 1}\left(T(b^{\lfloor \log_b(x) \rfloor})\right). \quad (\text{F.56})$$

$\log_b(a) < c$

By [Master theorem over powers] (F.21),

$$\Theta_B(T) \subset \Theta_B(n^c). \quad (\text{F.57})$$

By subset-summability,

$$\begin{aligned} \Theta_{\mathbb{R} \geq 1}\left(T(b^{\lfloor \log_b(x) \rfloor})\right) &\subset \Theta_{\mathbb{R} \geq 1}\left(\left(b^{\lfloor \log_b(x) \rfloor}\right)^c\right) \\ &= \Theta_{\mathbb{R} \geq 1}(x^c). \end{aligned} \quad (\text{F.58})$$

$\log_b(a) = c$

By [Master theorem over powers] (F.21),

$$\Theta_B(T) \subset \Theta_B(n^c \log_b(bn)). \quad (\text{F.59})$$

By subset-summability,

$$\begin{aligned} \Theta_{\mathbb{R} \geq 1}\left(T(b^{\lfloor \log_b(x) \rfloor})\right) &\subset \Theta_{\mathbb{R} \geq 1}\left(\left(b^{\lfloor \log_b(x) \rfloor}\right)^c \log_b\left(bb^{\lfloor \log_b(x) \rfloor}\right)\right) \\ &= \Theta_{\mathbb{R} \geq 1}(x^c \log_b(bx)). \end{aligned} \quad (\text{F.60})$$

$\log_b(a) > c$

By [Master theorem over powers] (F.21),

$$\Theta_B(T) = \Theta_B\left(n^{\log_b(a)}\right). \quad (\text{F.61})$$

By subset-summability,

$$\begin{aligned} \Theta_{\mathbb{R} \geq 1}(t) &= \Theta_{\mathbb{R} \geq 1}\left(\left(b^{\lfloor \log_b(x) \rfloor}\right)^{\log_b(a)}\right) \\ &= \Theta_{\mathbb{R} \geq 1}\left(x^{\log_b(a)}\right). \end{aligned} \quad (\text{F.62})$$

$f \in \Theta_{\mathbb{R} \geq 1}(n^c)$

If $f \in \Theta_{\mathbb{R} \geq 1}(n^c)$, then the subsets in the above proofs can be replaced with equalities by [Master theorem over powers] (F.21). \square

F.3 Master theorem over integers

Definition F.63 (Master function over integers). Let $a \in \mathbb{R}^{\geq 1}$, $b \in \mathbb{R}^{\geq 2}$, $d \in \mathbb{R}^{>0}$, and $F \in R_{\mathbb{N}^{\geq 1}}$. A *master function over integers* is a function $T \in R_{\mathbb{N}^{\geq 1}}$ defined by the recurrence equation

$$T(n) = \begin{cases} aT(\lceil n/b \rceil) + F(n), & n \geq b, \\ d, & n < b. \end{cases} \quad (\text{F.64})$$

The set of such functions is denoted by $\mathcal{M}_I(a, b, d, F)$.

Note F.65 (Stricter requirement). The requirement $b \geq 2$ is stricter than the requirement $b > 1$ for the other Master theorems; this is needed to avoid the recursion getting stuck to a fixed-point $\geq b$. ■

Definition F.66 (Ceiling division). The *ceiling division* is a function $N: \mathbb{N}^{\geq 1} \rightarrow \mathbb{N}^{\geq 1}$ such that $N(n) = \lceil n/b \rceil$.

Definition F.67 (Ceiling division number). The *ceiling division number* is a function $M: \mathbb{N}^{\geq 1} \rightarrow \mathbb{N}$ such that $M(n) = \min\{i \in \mathbb{N} : N^{(i)}(n) < b\}$.

Theorem F.68 (Fixed points of the ceiling division). Let $b \in \mathbb{R}^{>1}$ and

$$F = \{n \in \mathbb{N}^{\geq 1} : \lceil n/b \rceil = n\}. \quad (\text{F.69})$$

Then

$$F = \{n \in \mathbb{N}^{\geq 1} : 1 \leq n < b/(b-1)\}. \quad (\text{F.70})$$

In addition, $F = \{1\} \iff b \geq 2$.

Appendix F. Proofs of Master theorems

Proof. To characterize the fixed points,

$$\begin{aligned}
 & n \in F \\
 \iff & \lceil n/b \rceil = n \\
 \iff & \exists v \in \mathbb{R} : -b < v \leq 0 \text{ and } nb + v = n \\
 \iff & \exists v \in \mathbb{R} : -b < v \leq 0 \text{ and } v = n(1 - b) \\
 \iff & -b < n(1 - b) \leq 0 \\
 \iff & b/(b - 1) > n \geq 0 \\
 \iff & 0 \leq n < b/(b - 1) \\
 \iff & 1 \leq n < b/(b - 1),
 \end{aligned} \tag{F.71}$$

where we use a version of Euclidean division and the last step follows because $n \in \mathbb{N}^{\geq 1}$. Based on this characterization,

$$\begin{aligned}
 & F = \{1\} \\
 \iff & b/(b - 1) \leq 2 \\
 \iff & b \leq 2(b - 1) \\
 \iff & 0 \leq b - 2 \\
 \iff & b \geq 2.
 \end{aligned} \tag{F.72}$$

Let us also note that

$$\begin{aligned}
 & b/(b - 1) \leq b \\
 \iff & b \leq b(b - 1) \\
 \iff & 1 \leq b - 1 \\
 \iff & b \geq 2.
 \end{aligned} \tag{F.73}$$

□

Theorem F.74 (Explicit form for a Master function over integers).

Let $T \in \mathcal{M}_I(a, b, d, F)$ be a Master function over integers. Then

$$T(n) = a^{M(n)}d + \sum_{i=0}^{M(n)-1} a^i F(N^{(i)}(n)), \tag{F.75}$$

Proof. The function M is well-defined by [Fixed points of the ceiling division] (F.68), since $b \geq 2$. Let $n \in \mathbb{N}^{\geq 1}$.

Appendix F. Proofs of Master theorems

Pattern

Suppose $M(n) = 0$. Then

$$T(n) = d. \quad (\text{F.76})$$

Suppose $M(n) = 1$. Then

$$\begin{aligned} T(n) &= aT(\lceil n/b \rceil) + F(n) \\ &= aT(N^{(1)}(n)) + F(n) \\ &= ad + F(n). \end{aligned} \quad (\text{F.77})$$

Suppose $M(n) = 2$. Then

$$\begin{aligned} T(n) &= aT(N^{(1)}(n)) + F(n) \\ &= a\left(aT\left(\left\lceil N^{(1)}(n)/b \right\rceil\right) + F(N^{(1)}(n))\right) + F(n) \\ &= a^2T(N^{(2)}(n)) + aF(N^{(1)}(n)) + F(n) \\ &= a^2d + aF(N^{(1)}(n)) + F(n). \end{aligned} \quad (\text{F.78})$$

Suppose $M(n) = 3$. Then

$$\begin{aligned} T(n) &= a^2T(N^{(2)}(n)) + aF(N^{(1)}(n)) + F(n) \\ &= a^2\left(aT\left(\left\lceil N^{(2)}(n)/b \right\rceil\right) + F(N^{(2)}(n))\right) + aF(N^{(1)}(n)) + F(n) \\ &= a^3T(N^{(3)}(n)) + a^2F(N^{(2)}(n)) + aF(N^{(1)}(n)) + F(n) \\ &= a^3d + a^2F(N^{(2)}(n)) + aF(N^{(1)}(n)) + F(n). \end{aligned} \quad (\text{F.79})$$

Induction

This suggests the pattern

$$T(n) = a^{M(n)}d + \sum_{i=0}^{M(n)-1} a^i F(N^{(i)}(n)). \quad (\text{F.80})$$

This can be proved by induction, as in the explicit form for powers. □

Appendix F. Proofs of Master theorems

Theorem F.81 (Bounds for N). *Let $n \in \mathbb{N}^{\geq 1}$, and $b \in \mathbb{R}^{\geq 2}$. Then*

$$n/b^i \leq N^{(i)}(n) < n/b^i + 2. \quad (\text{F.82})$$

Proof.

Pattern

It holds that $\lceil x \rceil < x + 1$ for all $x \in \mathbb{R}$. Therefore,

$$\begin{aligned} N^{(1)}(n) &= \lceil n/b \rceil \\ &< (n/b) + 1 \\ N^{(2)}(n) &= \left\lceil N^{(1)}(n)/b \right\rceil \\ &\leq \lceil ((n/b) + 1)/b \rceil \\ &= \lceil n/b^2 + 1/b \rceil \\ &< n/b^2 + 1/b + 1 \\ N^{(3)}(n) &= \left\lceil N^{(2)}(n)/b \right\rceil \\ &\leq \lceil n/b^3 + 1/b^2 + 1/b \rceil \\ &< n/b^3 + 1/b^2 + 1/b + 1. \end{aligned} \quad (\text{F.83})$$

Induction

This suggests the pattern

$$N^{(i)}(n) < n/b^i + \sum_{j=0}^{i-1} (1/b)^j. \quad (\text{F.84})$$

This can be proved by induction. Since $b > 1$,

$$\begin{aligned} N^{(i)}(n) &< n/b^i + \frac{1 - (1/b)^i}{1 - (1/b)} \\ &< n/b^i + \frac{1}{1 - (1/b)}. \end{aligned} \quad (\text{F.85})$$

Since $b \geq 2$,

$$N^{(i)}(n) < n/b^i + 2. \quad (\text{F.86})$$

Similarly, since $x \leq \lceil x \rceil$, for all $x \in \mathbb{R}$, it can be proved that $n/b^i \leq N^{(i)}(n)$. \square

Appendix F. Proofs of Master theorems

Note F.87 (Ceiling division pitfall). It holds that $\lceil \lceil n/a \rceil / b \rceil = \lceil n/(ab) \rceil$, for all $a, b \in \mathbb{N}^{>0}$ and $n \in \mathbb{N}$. A potential pitfall in the proof of [Bounds for N] (F.81) is to assume that this also holds when $a, b \in \mathbb{R}^{>0}$. A counterexample is given by $n = 6$ and $a = b = 5/2$. ■

Theorem F.88 (Bounds for M). Let $n \in \mathbb{N}^{\geq 1}$, and $b \in \mathbb{R}^{\geq 2}$. Then

$$\lfloor \log_b(n) \rfloor \leq M(n) \leq \lfloor \log_b(n) \rfloor + 2. \quad (\text{F.89})$$

Proof. By [Bounds for N] (F.81),

$$\begin{aligned} N^{(\lfloor \log_b(n) \rfloor + 1)}(n) &< \frac{n}{b^{\lfloor \log_b(n) \rfloor + 1}} + 2 \\ &< \frac{n}{b^{\log_b(n)}} + 2 \\ &= \frac{n}{n} + 2 \\ &= 3. \end{aligned} \quad (\text{F.90})$$

This is equivalent to $N^{(\lfloor \log_b(n) \rfloor + 1)}(n) \in \{1, 2\}$. Since $N^{(1)}(2) = \lceil 2/b \rceil = 1$, it follows that $M(n) \leq \lfloor \log_b(n) \rfloor + 2$. Similarly, it follows that $\lfloor \log_b(n) \rfloor \leq M(n)$. □

Theorem F.91 (Multiplicative bounds for $N^{(i)}(n)$). Let $n \in \mathbb{N}^{\geq 1}$, $b \in \mathbb{R}^{\geq 2}$, and $i \in [0, \lfloor \log_b(n) \rfloor + 1]$. Then

$$n/b^i \leq N^{(i)}(n) < 3b(n/b^i). \quad (\text{F.92})$$

Proof. We would like to find $\beta \in \mathbb{R}^{>0}$ such that

$$\begin{aligned} n/b^i + 2 &\leq \beta(n/b^i) \\ \iff \beta &\geq 1 + 2b^i/n, \end{aligned} \quad (\text{F.93})$$

Appendix F. Proofs of Master theorems

For the given argument-sets,

$$\begin{aligned}
 1 + 2b^i/n &\leq 1 + 2b^{\lfloor \log_b(n) \rfloor + 1}/n \\
 &\leq 1 + 2b^{\log_b(n)+1}/n \\
 &= 1 + 2b \\
 &\leq b + 2b \\
 &= 3b.
 \end{aligned} \tag{F.94}$$

Therefore $\beta = 3b$ suffices. Then

$$\begin{aligned}
 n/b^i &\leq N^{(i)}(n) \\
 &< n/b^i + 2 \\
 &\leq 3b(n/b^i),
 \end{aligned} \tag{F.95}$$

by [Bounds for N] (F.81). □

Theorem F.96 (Master theorem over integers). *Let $T \in \mathcal{M}_I(a, b, d, F)$ be a Master function over integers, and $F \in \Theta_{\mathbb{N} \geq 1}(n^c)$, where $c \in \mathbb{R}^{\geq 0}$. Then*

$$\begin{aligned}
 \log_b(a) < c &\implies T \in \Theta_{\mathbb{N} \geq 1}(n^c), \\
 \log_b(a) = c &\implies T \in \Theta_{\mathbb{N} \geq 1}(n^c \log_b(bn)), \\
 \log_b(a) > c &\implies T \in \Theta_{\mathbb{N} \geq 1}\left(n^{\log_b(a)}\right).
 \end{aligned} \tag{F.97}$$

If $F \in \Theta_{\mathbb{N} \geq 1}(n^c)$, then each $\Theta_{\mathbb{N} \geq 1}$ can be replaced with $\Theta_{\mathbb{N} \geq 1}$.

Proof.

Explicit form

By [Explicit form for a Master function over integers] (F.74),

$$T(n) = a^{M(n)}d + \sum_{i=0}^{M(n)-1} a^i F(N^{(i)}(n)), \tag{F.98}$$

First term

Appendix F. Proofs of Master theorems

By [Bounds for M] (F.88),

$$\begin{aligned} & \lfloor \log_b(n) \rfloor \leq M(n) \leq \lfloor \log_b(n) \rfloor + 2 \\ \implies & a^{\lfloor \log_b(n) \rfloor} d \leq a^{M(n)} d \leq a^{\lfloor \log_b(n) \rfloor + 2} d \\ \implies & a^{\lfloor \log_b(n) \rfloor} d \leq a^{M(n)} d \leq a^{\lfloor \log_b(n) \rfloor} a^2 d \end{aligned} \quad (\text{F.99})$$

By order-consistency and scale-invariance,

$$\Theta_{\mathbb{N} \geq 1} \left(a^{M(n)} d \right) = \Theta_{\mathbb{N} \geq 1} \left(a^{\lfloor \log_b(n) \rfloor} \right). \quad (\text{F.100})$$

Sum term

Since $F \in \Theta_{\mathbb{N} \geq 1}(n^c)$, by subset-summability

$$\begin{aligned} & \Theta_{\mathbb{N} \geq 1} \left(\sum_{i=0}^{M(n)-1} a^i F(N^{(i)}(n)) \right) \\ \subset & \Theta_{\mathbb{N} \geq 1} \left(\sum_{i=0}^{M(n)-1} a^i (N^{(i)}(n))^c \right). \end{aligned} \quad (\text{F.101})$$

By [Multiplicative bounds for $N^{(i)}(n)$] (F.91),

$$(n/b^i) \leq N^{(i)}(n) < 3b(n/b^i), \quad (\text{F.102})$$

for all $i \in [0, \lfloor \log_b(n) \rfloor + 1]$. By order-consistency and scale-invariance,

$$\begin{aligned} & \Theta_{\mathbb{N} \geq 1} \left(\sum_{i=0}^{M(n)-1} a^i (N^{(i)}(n))^c \right) \\ = & \Theta_{\mathbb{N} \geq 1} \left(\sum_{i=0}^{M(n)-1} a^i (n/b^i)^c \right). \end{aligned} \quad (\text{F.103})$$

Combined terms

We combine the first term and the sum term by additivity:

$$\Theta_{\mathbb{N} \geq 1}(T) \subset \Theta_{\mathbb{N} \geq 1} \left(a^{\lfloor \log_b(n) \rfloor} + \sum_{i=0}^{M(n)-1} a^i (n/b^i)^c \right). \quad (\text{F.104})$$

Appendix F. Proofs of Master theorems

By [Bounds for M] (F.88),

$$\lfloor \log_b(n) \rfloor \leq M(n) \leq \lfloor \log_b(n) \rfloor + 2. \quad (\text{F.105})$$

Therefore, by order-consistency,

$$\begin{aligned} & \Theta_{\mathbb{N} \geq 1} \left(a^{\lfloor \log_b(n) \rfloor} + \sum_{i=0}^{\lfloor \log_b(n) \rfloor - 1} a^i (n/b^i)^c \right) \\ & \subset \Theta_{\mathbb{N} \geq 1} \left(a^{\lfloor \log_b(n) \rfloor} + \sum_{i=0}^{M(n)-1} a^i (n/b^i)^c \right) \\ & \subset \Theta_{\mathbb{N} \geq 1} \left(a^{\lfloor \log_b(n) \rfloor} + \sum_{i=0}^{\lfloor \log_b(n) \rfloor + 1} a^i (n/b^i)^c \right). \end{aligned} \quad (\text{F.106})$$

The last two terms in the sum are given by

$$\begin{aligned} & a^{\lfloor \log_b(n) \rfloor} (n/b^{\lfloor \log_b(n) \rfloor})^c + a^{\lfloor \log_b(n) \rfloor + 1} (n/b^{\lfloor \log_b(n) \rfloor + 1})^c \\ & = a^{\lfloor \log_b(n) \rfloor} (n/b^{\lfloor \log_b(n) \rfloor})^c (1 + a/b^c) \\ & \in \Theta_{\mathbb{N} \geq 1} \left(a^{\lfloor \log_b(n) \rfloor} \right). \end{aligned} \quad (\text{F.107})$$

Therefore,

$$\begin{aligned} & \Theta_{\mathbb{N} \geq 1} \left(a^{\lfloor \log_b(n) \rfloor} + \sum_{i=0}^{M(n)-1} a^i (n/b^i)^c \right) \\ & = \Theta_{\mathbb{N} \geq 1} \left(a^{\lfloor \log_b(n) \rfloor} + \sum_{i=0}^{\lfloor \log_b(n) \rfloor - 1} a^i (n/b^i)^c \right). \end{aligned} \quad (\text{F.108})$$

Reduction to reals

Let $t \in \mathcal{M}_R(a, b, d, f)$ be a Master function over reals, where $f \in R_{\mathbb{R} > 1}$ is such that $f(x) = x^c$. By [Explicit form for a Master function over reals] (F.35),

$$t(x) = a^{\lfloor \log_b(x) \rfloor} + \sum_{i=0}^{\lfloor \log_b(x) \rfloor - 1} a^i (x/b^i)^c. \quad (\text{F.109})$$

Appendix F. Proofs of Master theorems

Therefore,

$$\Theta_{\mathbb{N} \geq 1}(T) \subset \Theta_{\mathbb{N} \geq 1}(t|\mathbb{N}^{\geq 1}). \quad (\text{F.110})$$

The results follow from [Master theorem over reals] (F.51).

Θ -sets

If $F \in \Theta_{\mathbb{N} \geq 1}(n^c)$, then we can get rid of the subsets in the above proof. \square

Appendix G

Comparison of definitions

In this section we will study various candidate definitions for the O-notation. This highlights the ways in which some familiar candidate definitions fail the primitive properties. The properties of the candidate definitions, proved in this section, are summarized upfront in Table G.1.

G.1 Trivial linear dominance

Definition G.1 (Trivial linear dominance). *Trivial linear dominance* \emptyset is defined by

$$\emptyset_X(f) = R_X, \tag{G.2}$$

for all $f \in R_X$, and all $X \in U$, where U is the class of all sets.

Theorem G.3 (Trivial linear dominance is local linear dominance). *The family $\mathcal{F} = \{\emptyset : X \in U\}$ is a family of filter bases with induced sub-structure.*

Proof.

Directedness

Let $A, B \in \mathcal{F}(X)$, where $X \in U$. Then $A = \emptyset = B$, and we may choose $C := \emptyset \in \mathcal{F}(X)$, so that $C \subset A \cap B$.

Appendix G. Comparison of definitions

Property	\mathcal{O}	\mathcal{G}	\mathcal{H}	\mathcal{I}	\mathcal{J}
Order-consistency	\checkmark D.6				\checkmark G.61
Reflexivity	\checkmark C.7				
Transitivity	\checkmark D.10				\checkmark G.63
Orderness	\checkmark C.10				
Zero-separation	\checkmark C.8				\times G.68
Zero-triviality	\checkmark D.57	\times D.57	\times D.57	\times D.57	\times G.113
One-separation	\checkmark 4.10	\checkmark G.38	\checkmark G.26	\checkmark G.14	\checkmark G.70
Scale-invariance	\checkmark D.20				\checkmark G.72
Scalar homogeneity	\checkmark C.21				
Sub-homogeneity in \mathbb{N}	\checkmark D.23				\times G.82
Sub-hom. in $1/\mathbb{N}^{>0}$	\checkmark D.23				\checkmark G.85
Sub-homogeneity in $\mathbb{Q}^{\geq 0}$	\checkmark C.2				\times G.82
Sub-homogeneity	\checkmark C.4				\times G.82
Super-homogeneity	\checkmark D.27	\times D.27	\times D.27	\times D.27	\times G.92
Power-homogeneity	\checkmark D.50				\checkmark G.98
Additive consistency	\checkmark C.47				
Multiplicative cons.	\checkmark C.53				
Maximum consistency	\checkmark C.50				
Locality	\checkmark D.14				\checkmark G.66
Sub-multiplicativity	\checkmark C.25				\times G.82
Super-multiplicativity	\checkmark D.29				\checkmark G.94
Sub-restrictability	\checkmark C.18				
Super-restrictability	\checkmark C.20				
Maximum rule	\checkmark C.56				
Summation rule	\checkmark C.60				
Maximum-sum rule	\checkmark C.63				
Additivity	\checkmark C.66				
Bounded translation-inv.	\checkmark C.70				
Sub-composability	\checkmark 4.16	\times G.46	\times G.29	\times G.18	\checkmark G.76
Injective sub-comp.	\checkmark 4.16	\checkmark G.49	\times G.29	\times G.18	\checkmark G.76
Injective super-comp.	\checkmark C.13	\checkmark C.13	?	?	\checkmark C.13
Extensibility	\checkmark G.55	\times G.50	\times G.30	\checkmark G.20	\checkmark G.89
Subset-summability	\checkmark 5.22	\times C.74	\times C.74	\times C.74	?
Universe	sets	sets	U	U	sets

Table G.1: Comparison of O-notations. The \checkmark and \times mean that we have proved and disproved, respectively, the property. The following number refers to the corresponding theorem. Primitive properties are marked with a bold face, and $U = \bigcup_{d \in \mathbb{N}} \mathcal{P}(\mathbb{R}^d)$.

Appendix G. Comparison of definitions

Induced sub-structure

Let $D \subset X$, where $X \in U$. Then $\mathcal{F}(D) = \{\emptyset\} = \{A \cap D : A \in \mathcal{F}(X)\}$. \square

Theorem G.4 (One-separation fails for \emptyset). *\emptyset does not have one-separation.*

Proof. It holds that $n \in R_{\mathbb{N}^{>0}} = \emptyset_{\mathbb{N}^{>0}}(1)$. \square

Theorem G.5 (Sub-composability for \emptyset). *\emptyset has sub-composability.*

Proof. The right side of the implication

$$(f \circ s) \in \mathcal{O}_Y(g \circ s) \tag{G.6}$$

holds by the definition of trivial dominance for all $f, g \in R_X$ and $s: Y \rightarrow X$. \square

Note G.7 (Trivial linear dominance has lots of nice properties?). Trivial linear dominance satisfies all of the desirable properties, except those of non-triviality. This underlines the importance of the non-triviality properties. \blacksquare

Note G.8 (Equivalent definitions). Suppose local linear dominance is defined so that $\emptyset \in \mathcal{F}(X)$, for all $X \in U$. Then it is equivalent to trivial linear dominance. \blacksquare

G.2 Asymptotic linear dominance

Recall the definition of asymptotic linear dominance from Chapter 1:

Definition G.9 (Asymptotic linear dominance). *Asymptotic linear dominance \emptyset is defined by $g \in \emptyset_X(f)$ if and only if*

$$\exists c \in \mathbb{R}^{>0}, \exists y \in \mathbb{R}^d : (g|X^{\geq y}) \leq c(f|X^{\geq y}), \tag{G.10}$$

Appendix G. Comparison of definitions

for all $f, g \in R_X$, and all $X \in U$, where $U = \bigcup_{d \in \mathbb{N}_{>0}} \mathcal{P}(\mathbb{R}^d)$.

Theorem G.11 (Asymptotic linear dominance is local linear dominance). *The family $\mathcal{F} = \{\{X^{\geq y} : y \in \mathbb{R}^d\} : X \in U\}$ is a family of filter bases with induced sub-structure.*

Proof.

Directedness

Let $X \subset \mathbb{R}^d$, and $A, B \in \mathcal{F}(X)$. Then there exist $y_1, y_2 \in \mathbb{R}^d$, such that $A = X^{\geq y_1}$, and $B = X^{\geq y_2}$. Let $y := \sup_{\leq} \{y_1, y_2\}$, and $C := X^{\geq y}$. Then $C \in \mathcal{F}(X)$, and $C \subset A \cap B$.

Induced sub-structure

Let $D \subset X \subset \mathbb{R}^d$. Then

$$\begin{aligned} \mathcal{F}(D) &= \{D^{\geq y} : y \in \mathbb{R}^d\} \\ &= \{X^{\geq y} \cap D : y \in \mathbb{R}^d\}. \end{aligned} \tag{G.12}$$

□

Note G.13 (Intrinsic pitfall). Let us consider a variant of asymptotic linear dominance, where $\mathcal{F}(X) = \{X^{\geq y} : y \in X\}$, when $X \subset \mathbb{R}^d$. We call this an *intrinsic* definition — in contrast to the extrinsic ($y \in \mathbb{R}^d$) definition we have given. Then $\mathcal{F}(X)$ is not a filter basis, and the definition fails many properties.

An example is given by $X = (-\infty, 1) \times \mathbb{R} \cup [1, \infty) \times (-\infty, 1)$, $y_1 = (0, 1) \in X$, and $y_2 = (1, 0) \in X$. Then $X^{\geq y_1} \cap X^{\geq y_2} = \emptyset$, but there is no $y_3 \in X$ such that $X^{\geq y_3} = \emptyset$. ■

Theorem G.14 (One-separation for Θ). $n \notin \Theta_{\mathbb{N}_{>0}}(1)$.

Proof. Θ has one-separation if and only if $(\forall A \in \mathcal{F}(\mathbb{N}^{>0}) : |A| \notin \mathbb{N})$ by [One-separation for Θ characterized] (D.18). Let $X := \mathbb{N}^{>0}$ and $A \in \mathcal{F}(X)$. Then

Appendix G. Comparison of definitions

there exists $y \in \mathbb{N}$ such that $A = X^{\geq y}$. Let $f: A \rightarrow \mathbb{N}$ be such that $f(x) = x - y$. Let $x_1, x_2 \in A$. Then

$$\begin{aligned} f(x_1) &= f(x_2) \\ \iff x_1 - y &= x_2 - y \\ \iff x_1 &= x_2. \end{aligned} \tag{G.15}$$

Therefore f is injective. Let $z \in \mathbb{N}$. Then $f(z + y) = (z + y) - y = z$. Therefore f is surjective. Since f is a bijection between A and \mathbb{N} , $|A| = |\mathbb{N}| \notin \mathbb{N}$. Therefore one-separation holds for Θ . \square

Theorem G.16 (Sub-composability for Θ for fixed $s: Y \rightarrow X$). *Let $d_x, d_y \in \mathbb{N}^{>0}$, $X \in U$ be such that $X \subset \mathbb{R}^{d_x}$, $Y \in U$ be such that $Y \subset \mathbb{R}^{d_y}$, and $s: Y \rightarrow X$. Then sub-composability holds for Θ and s if and only if*

$$\forall x^* \in \mathbb{R}^{d_x}, \exists y^* \in \mathbb{R}^{d_y} : \vec{s}(Y^{\geq y^*}) \subset X^{\geq x^*}. \tag{G.17}$$

Proof. Substitute the filter bases of Θ into [Sub-composability for Θ for fixed $s: Y \rightarrow X$] (D.35). \square

Theorem G.18 (Injective sub-composability fails for Θ in \mathbb{N}^2). *Θ does not have injective sub-composability from \mathbb{N}^2 to \mathbb{N} .*

Proof. Let $X := \mathbb{N}^2$, $x^* := (1, 0) \in X$, $Y := \mathbb{N}$, $y^* \in \mathbb{N}$, and $s: Y \rightarrow X$ be such that $s(n) = (0, n)$. Then $\vec{s}(Y^{\geq y^*}) \cap X^{\geq x^*} = \emptyset$. Since $\vec{s}(Y^{\geq y^*})$ is non-empty, $\vec{s}(Y^{\geq y^*}) \not\subset X^{\geq x^*}$. Injective sub-composability fails by [Sub-composability for Θ for fixed $s: Y \rightarrow X$] (G.16). \square

Theorem G.19 (Injective sub-composability fails for Θ in \mathbb{Z}). *Θ does not have injective sub-composability from \mathbb{Z} to \mathbb{N} .*

Proof. Let $X := \mathbb{Z}$, $Y = \mathbb{N}$, $x^* := 0 \in X$, and $s: Y \rightarrow X$ be such that $s(n) = -(n + 1)$. Then $\vec{s}(Y^{\geq y^*}) \cap X^{\geq x^*} = \emptyset$, for all $y^* \in Y$. Since $\vec{s}(Y^{\geq y^*})$ is non-empty, $\vec{s}(Y^{\geq y^*}) \not\subset X^{\geq x^*}$. Injective sub-composability fails by [Sub-composability for Θ for fixed $s: Y \rightarrow X$] (G.16). \square

Appendix G. Comparison of definitions

Theorem G.20 (Extensibility for Θ). *Θ has extensibility.*

Proof. Let $A \in \mathcal{F}(X)$ and $B = A \times Y$. Then $B \in \mathcal{F}(X \times Y)$, and $\overrightarrow{[X \times Y]}(B) = A$. The result follows by [Characterization of Extensibility for Θ] (D.48). \square

G.3 Co-asymptotic linear dominance

Recall the definition of co-asymptotic linear dominance from Chapter 1:

Definition G.21 (Coasymptotic linear dominance). *Coasymptotic linear dominance Θ is defined by $g \in \Theta_X(f)$ if and only if*

$$\exists c \in \mathbb{R}^{>0}, \exists y \in \mathbb{R}^d : (g|(X \setminus X^{<y})) \leq c(f|(X \setminus X^{<y})). \quad (\text{G.22})$$

for all $f, g \in R_X$, and all $X \in U$, where $U = \bigcup_{d \in \mathbb{N}^{>0}} \mathcal{P}(\mathbb{R}^d)$.

Theorem G.23 (Co-asymptotic linear dominance is local linear dominance). *The family $\mathcal{F} = \{\{(X \setminus X^{<y}) : y \in \mathbb{R}^d\} : X \in U\}$ is a family of filter bases with induced sub-structure.*

Proof.

Directedness

Let $X \subset \mathbb{R}^d$, and $A, B \in \mathcal{F}(X)$. Then there exists $y_1, y_2 \in \mathbb{R}^d$, such that $A = (X \setminus X^{<y_1})$ and $B = (X \setminus X^{<y_2})$. Let $y := \sup_{\leq} \{y_1, y_2\}$, and $C := (X \setminus X^{<y})$. Then $C \in \mathcal{F}(X)$ and $C \subset A \cap B$.

Induced sub-structure

Let $D \subset X \subset \mathbb{R}^d$. Then

$$\begin{aligned} \mathcal{F}(D) &= \{(D \setminus D^{<y}) : y \in \mathbb{R}^d\} \\ &= \{(X \setminus X^{<y}) \cap D : y \in \mathbb{R}^d\}. \end{aligned} \quad (\text{G.24})$$

\square

Appendix G. Comparison of definitions

Note G.25 (Intrinsic pitfall). The same intrinsic pitfall as described in Note G.13 for asymptotic linear dominance also holds for co-asymptotic linear dominance. ■

Theorem G.26 (One-separation for Θ). $n \notin \Theta_{\mathbb{N}^{>0}}(1)$.

Proof. The proof is the same as in [One-separation for Θ] (G.14), since co-asymptotic linear dominance is equivalent to asymptotic linear dominance in $\mathbb{N}^{>0}$. □

Theorem G.27 (Composability for Θ for fixed $s: Y \rightarrow X$). Let $d_x, d_y \in \mathbb{N}^{>0}$, $X \in U$ be such that $X \subset \mathbb{R}^{d_x}$, $Y \in U$ be such that $Y \subset \mathbb{R}^{d_y}$, and $s: Y \rightarrow X$. Then composability holds for Θ and s if and only if

$$\forall x^* \in \mathbb{R}^{d_x}, \exists y^* \in \mathbb{R}^{d_y} : \vec{s}\left((Y \setminus Y^{<y^*})\right) \subset (X \setminus X^{<x^*}). \quad (\text{G.28})$$

Proof. Substitute the filter bases of Θ into [Sub-composability for Θ for fixed $s: Y \rightarrow X$] (D.35). □

Theorem G.29 (Injective sub-composability fails for Θ in \mathbb{Z}). Θ does not have injective sub-composability.

Proof. Let $X := \mathbb{Z}$, $Y = \mathbb{N}$, $x^* := 0 \in X$, and $s: Y \rightarrow X$ be such that $s(n) = -(n+1)$. Then $\vec{s}\left((Y \setminus Y^{<y^*})\right) \cap (X \setminus X^{<x^*}) = \emptyset$, for all $y^* \in Y$. Since $\vec{s}\left((Y \setminus Y^{<y^*})\right)$ is non-empty, $\vec{s}\left((Y \setminus Y^{<y^*})\right) \not\subset (X \setminus X^{<x^*})$. Injective sub-composability fails by [Composability for Θ for fixed $s: Y \rightarrow X$] (G.27). □

Theorem G.30 (Extensibility fails for Θ). Θ does not have extensibility.

Proof. Let $A = \mathbb{N}^{>0}$. For any $B \in \mathcal{F}(\mathbb{N}^2)$, $0 \in \overrightarrow{[\mathbb{N} \times \mathbb{N}]}(B)$, but $0 \notin A$. Extensibility fails by [Characterization of Extensibility for Θ] (D.48). □

Appendix G. Comparison of definitions

Theorem G.31 (Θ is equal to Θ in \mathbb{N}^d).

$$\Theta_X = \Theta_X, \quad (\text{G.32})$$

for all $X \subset \mathbb{N}^d$.

Proof. Let $d \in \mathbb{N}^{>0}$, $X \subset \mathbb{N}^d$, $f \in R_X$, and $x \in \mathbb{N}^d$. Then $(X \setminus X^{<x}) \in \bar{\mathcal{P}}(X)$, and therefore $\Theta_X(f) \subset \Theta_X(f)$. Let $A \in \bar{\mathcal{P}}(X)$, and $y = 1 + \sup_{\leq}(X \setminus A)$. Then $(X \setminus X^{<y}) \subset A$, and therefore $\Theta_X(f) \subset \Theta_X(f)$. \square

Theorem G.33 (Injective composability holds for Θ in \mathbb{N}^d). Θ has injective composability in \mathbb{N}^d .

Proof. This follows from $[\Theta \text{ is equal to } \Theta \text{ in } \mathbb{N}^d]$ (G.31) and $[\text{Injective composability for } \Theta]$ (G.49). \square

G.4 Cofinite linear dominance

Recall the definition of cofinite linear dominance from Chapter 1:

Definition G.34 (Cofinite linear dominance). Cofinite linear dominance Θ is defined by $g \in \Theta_X(f)$ if and only if^a

$$\exists c \in \mathbb{R}^{>0}, \exists A \in \bar{\mathcal{P}}(X) : (g|_A) \leq c(f|_A), \quad (\text{G.35})$$

for all $f, g \in R_X$, and all sets X .

$$\text{}^a\bar{\mathcal{P}}(X) = \{A \in \mathcal{P}(X) : |X \setminus A| < \infty\}.$$

Theorem G.36 (Cofinite linear dominance is local linear dominance). The family $\{\bar{\mathcal{P}}(X) : X \in U\}$ is a family of filter bases with induced sub-structure.

Proof.

Appendix G. Comparison of definitions

Directedness

Let $A, B \in \mathcal{F}(X)$, where $X \in U$. Then we may choose $C := A \cap B \in \mathcal{F}(X)$, so that $C \subset A \cap B$.

Induced sub-structure

Let $D \subset X$, where $X \in U$. If $B \in \mathcal{F}(D)$, then $B = D \cap (X \setminus (D \setminus B))$, and $(X \setminus (D \setminus B)) \in \mathcal{F}(X)$. If $A \in \mathcal{F}(X)$, then $A \cap D \in \mathcal{F}(D)$. \square

Note G.37 (Fréchet filter). The filter basis for cofinite linear dominance is also known as the Fréchet filter. ■

Theorem G.38 (One-separation for Θ). Θ has one-separation.

Proof. Θ has one-separation if and only if $(\forall A \in \mathcal{F}(\mathbb{N}^{>0}) : |A| \notin \mathbb{N})$ by [One-separation for Θ characterized] (D.18). Let $X := \mathbb{N}^{>0}$ and $A \in \mathcal{F}(X) = \overline{\mathcal{P}}(X)$. Let $f: A \rightarrow \mathbb{N}$ be such that $f(n) = |A^{<n}|$. Then f is a bijection between A and \mathbb{N} , and so $|A| = |\mathbb{N}| \notin \mathbb{N}$. Therefore one-separation holds for Θ . \square

Theorem G.39 (Sub-composability for Θ for fixed $s: Y \rightarrow X$). Let $X, Y \in U$, and $s: Y \rightarrow X$. Then Θ has sub-composability for s if and only if s is finite-to-one:

$$\forall x \in X : |\overleftarrow{s}(\{x\})| \in \mathbb{N}. \quad (\text{G.40})$$

Proof. The Θ has sub-composability for s if and only if

$$\forall A_X \in \mathcal{F}(X), \exists A_Y \in \mathcal{F}(Y) : \overrightarrow{s}(A_Y) \subset A_X \quad (\text{G.41})$$

by [Sub-composability for Θ for fixed $s: Y \rightarrow X$] (D.35). Substituting the filter bases of Θ , this is equivalent to

$$\forall A_X \in \overline{\mathcal{P}}(X), \exists A_Y \in \overline{\mathcal{P}}(Y) : \overrightarrow{s}(A_Y) \subset A_X. \quad (\text{G.42})$$

We will show equivalence to this formula.

Appendix G. Comparison of definitions

\implies

Suppose there exists $x \in X$ such that $|\overleftarrow{s}(\{x\})| \notin \mathbb{N}$. Then $|Y| \notin \mathbb{N}$, and therefore $\emptyset \notin \overline{\mathcal{P}}(Y)$. Let $A_X = X \setminus \{x\} \in \overline{\mathcal{P}}(X)$, and $A_Y \in \overline{\mathcal{P}}(Y)$. Then $A_Y \neq \emptyset$, and $A_Y \cap \overleftarrow{s}(\{x\}) \neq \emptyset$, which is equivalent to $x \in \overrightarrow{s}(A_Y)$. Therefore $\overrightarrow{s}(A_Y) \not\subset A_X$; Θ does not have sub-composability for s .

\impliedby

Let $A_X \in \overline{\mathcal{P}}(X)$ be such that $X \setminus A_X = \{x_1, \dots, x_n\}$, and $A_Y = \overleftarrow{s}(A_X)$. Then $\overrightarrow{s}(A_Y) = A_X$, and

$$\begin{aligned} |Y \setminus A_Y| &= |Y \setminus \overleftarrow{s}(A_X)| \\ &= |\overleftarrow{s}(X \setminus A_X)| \\ &= |\overleftarrow{s}(\{x_1, \dots, x_n\})| \\ &\leq |\overleftarrow{s}(\{x_1\})| + \dots + |\overleftarrow{s}(\{x_n\})| \\ &\in \mathbb{N}. \end{aligned} \tag{G.43}$$

Therefore $A_Y \in \overline{\mathcal{P}}(Y)$; Θ has sub-composability for s . \square

Theorem G.44 (Sub-composability for Θ for positive $f \in R_X$). *Let $X, Y \in U$, $s: Y \rightarrow X$, and $f \in R_X^{>0}$. Then*

$$\Theta_X(f) \circ s \subset \Theta_X(f \circ s). \tag{G.45}$$

Proof. It holds that $|X \setminus A| < \infty$, for all $A \in \mathcal{F}(X)$, $X \in U$. The claim follows from [Sub-composability for Θ for positive functions] (D.43). \square

Theorem G.46 (Sub-composability fails for Θ). *Θ does not have sub-composability.*

Proof. Let $s: \mathbb{N} \rightarrow \mathbb{N}$ be such that

$$s(n) = \begin{cases} 0, & n \in 2\mathbb{N}, \\ n, & n \in 2\mathbb{N} + 1. \end{cases} \tag{G.47}$$

Then $|\overleftarrow{s}(\{0\})| \notin \mathbb{N}$, and so sub-composability fails for Θ and s by [Sub-composability for Θ for fixed $s: Y \rightarrow X$] (G.39). For example, let $\hat{f}: \mathbb{N} \rightarrow \mathbb{N}$

Appendix G. Comparison of definitions

be such that $\widehat{f}(0) > 0$ and $f: \mathbb{N} \rightarrow \mathbb{N}$ be such that $f(0) = 0$ and $\widehat{f} \in \Theta_{\mathbb{N}}(f)$. Then $(\widehat{f} \circ s) \notin \Theta_{\mathbb{N}}(f \circ s)$. \square

Theorem G.48 (Subset-summability fails for Θ). *Θ does not have subset-summability.*

Proof. Since subset-summability implies sub-composability, and sub-composability does not hold by [Sub-composability fails for Θ] (G.46), neither does subset-summability. \square

Theorem G.49 (Injective composability for Θ). *Θ has injective composability.*

Proof. Let $X, Y \in U$, and $s: Y \rightarrow X$ be injective. Then $|\overleftarrow{s}(\{x\})| \leq 1$, for all $x \in X$. Therefore Θ has sub-composability for s by [Sub-composability for Θ for fixed $s: Y \rightarrow X$] (G.39). Since Θ has order-consistency by [Order-consistency for Θ] (D.6), and locality by [Locality for Θ] (D.14), Θ has injective super-composability by [Injective super-composability is implied] (C.13). \square

Theorem G.50 (Extensibility fails for Θ). *Θ does not have extensibility.*

Proof. Let $A = \mathbb{N}^{>0}$. For any $B \in \mathcal{F}(\mathbb{N}^2)$, $0 \in \overrightarrow{[\mathbb{N} \times \mathbb{N}]}(B)$, but $0 \notin A$. Extensibility fails by [Characterization of Extensibility for Θ] (D.48). \square

G.5 Linear dominance

Recall the definition of linear dominance from Chapter 1:

Definition G.51 (Full linear dominance). *Full linear dominance Θ is defined by $g \in \Theta_X(f)$ if and only if*

$$\exists c \in \mathbb{R}^{>0} : g \leq cf, \tag{G.52}$$

for all $f, g \in R_X$, and all $X \in U$, where U is the class of all sets.

Appendix G. Comparison of definitions

Theorem G.53 (Linear dominance is local linear dominance). *The family $\{\{X\} : X \in U\}$ is a family of filter bases with induced sub-structure.*

Proof.

Directedness

Let $A, B \in \mathcal{F}(X)$, where $X \in U$. Then $A = X = B$, and we may choose $C := X \in \mathcal{F}(X)$, so that $C \subset A \cap B$.

Induced sub-structure

Let $D \subset X$, where $X \in U$. Then $\mathcal{F}(D) = \{D\} = \{A \cap D : A \in \mathcal{F}(X)\}$. □

Note G.54. The primitive properties of linear dominance are proved in Section 4.1. ■

Theorem G.55 (Extensibility for Θ). *Θ has extensibility.*

Proof. Let $A \in \mathcal{F}(X)$ and $B = X \times Y$. Then $A = X$ and $B \in \mathcal{F}(X \times Y)$. Since $\overrightarrow{[X \times Y]}(B) = X = A$, the result follows by [Characterization of Extensibility for Θ] (D.48). □

G.6 Affine dominance

Recall the definition of affine dominance from Chapter 1:

Definition G.56 (Affine dominance). *Affine dominance Θ is defined by $g \in \Theta_X(f)$ if and only if*

$$\exists c \in \mathbb{R}^{>0} : g \leq cf + c, \tag{G.57}$$

for all $f, g \in R_X$, and all $X \in U$, where U is the class of all sets.

Note G.58. Affine dominance is not a local linear dominance. ■

We shall apply the following lemma without mentioning it, since it is used in almost every proof.

Appendix G. Comparison of definitions

Theorem G.59 (Simplification lemma for Θ). *Let I be a finite set, $X_i \in U$, $f_i \in R_{X_i}$, and $\widehat{f}_i \in \Theta_{X_i}(f_i)$, for all $i \in I$. Then there exists $c \in \mathbb{R}^{>0}$, such that*

$$\widehat{f}_i \leq cf_i + c, \quad (\text{G.60})$$

for all $i \in I$.

Proof. There exists $c_i \in \mathbb{R}^{>0}$, such that $\widehat{f}_i \leq c_i f_i + c_i$, for all $i \in I$. Let $c = \max\{c_i : i \in I\}$. Then $\widehat{f}_i \leq cf_i + c$, for all $i \in I$. \square

Theorem G.61 (Order-consistency for Θ). *Let $X \in U$, and $f, g \in R_X$. Then*

$$f \leq g \implies \Theta_X(f) \subset \Theta_X(g). \quad (\text{G.62})$$

Proof. Let $\widehat{f} \in \Theta_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$ such that $\widehat{f} \leq cf + c$. Since $f \leq g$, it follows that $\widehat{f} \leq cg + c$. Therefore $\widehat{f} \in \Theta_X(g)$. \square

Theorem G.63 (Transitivity for Θ). *Let $X \in U$, and $f, g, h \in R_X$. Then*

$$(f \in \Theta_X(g) \text{ and } g \in \Theta_X(h)) \implies f \in \Theta_X(h). \quad (\text{G.64})$$

Proof. Let $f \in \Theta_X(g)$, and $g \in \Theta_X(h)$. Then there exists $c \in \mathbb{R}^{>0}$, such that $f \leq cg + c$ and $g \leq ch + c$. It follows that

$$\begin{aligned} f &\leq c(ch + c) + c \\ &= c^2h + (c^2 + c) \\ &\leq (c^2 + c)h + (c^2 + c). \end{aligned} \quad (\text{G.65})$$

Therefore $f \in \Theta_X(h)$. \square

Theorem G.66 (Locality for Θ). *Let $X \in U$, $f, g \in R_X$, and $C \subset \mathcal{P}(X)$*

Appendix G. Comparison of definitions

be a finite cover of X . Then

$$\forall D \in C : (f|D) \in \Theta_D(g|D) \implies f \in \Theta_X(g). \quad (\text{G.67})$$

Proof. Assume $(f|D) \in \Theta_D(g|D)$, for all $D \in C$. Since C is finite, there exists $c \in \mathbb{R}^{>0}$, such that $(f|D) \leq c(g|D) + c$, for all $D \in C$. Since C covers X , $f \leq cg + c$. Therefore $f \in \Theta_X(g)$. \square

Theorem G.68 (Zero-separation fails for Θ). *Let $X \in U$ be non-empty. Then*

$$\Theta_X(1) = \Theta_X(0). \quad (\text{G.69})$$

Proof. Θ_X has order-consistency by [Order-consistency for Θ] (G.61), transitivity by [Transitivity for Θ] (G.63), and orderness by [Orderness is implied] (C.10). It holds that $(x \mapsto 1) \leq 1(x \mapsto 0) + 1$, for all $x \in X$. Therefore $1 \in \Theta_X(0)$. By orderness, $\Theta_X(1) \subset \Theta_X(0)$. By order-consistency, $0 \in \Theta_X(1)$. By orderness, $\Theta_X(1) \supset \Theta_X(0)$. \square

Theorem G.70 (One-separation for Θ).

$$n \notin \Theta_{\mathbb{N}^{>0}}(1). \quad (\text{G.71})$$

Proof. It holds that $n > c1 + c$, for all $c \in \mathbb{R}^{>0}$, and $n \in \mathbb{N}^{>2c}$. Therefore $n \notin \Theta_{\mathbb{N}^{>0}}(1)$. \square

Theorem G.72 (Scale-invariance for Θ). *Let $X \in U$, $f \in R_X$, and $\alpha \in \mathbb{R}^{>0}$. Then*

$$\Theta_X(\alpha f) = \Theta_X(f). \quad (\text{G.73})$$

Proof.

Appendix G. Comparison of definitions

\subseteq

Assume $\hat{f} \in \Theta_X(\alpha f)$. Then there exists $c \in \mathbb{R}^{>0}$, such that

$$\begin{aligned}\hat{f} &\leq c(\alpha f) + c \\ &\leq \max(c\alpha, c)f + \max(c\alpha, c).\end{aligned}\tag{G.74}$$

Therefore $\hat{f} \in \Theta_X(f)$.

\supseteq

Assume $\hat{f} \in \Theta_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$, such that

$$\begin{aligned}\hat{f} &\leq cf + c \\ &= (c/\alpha)(\alpha f) + c \\ &\leq \max(c/\alpha, c)(\alpha f) + \max(c/\alpha, c).\end{aligned}\tag{G.75}$$

Therefore $\hat{f} \in \Theta_X(\alpha f)$.

□

Theorem G.76 (Composability for Θ). *Let $X, Y \in U$, $f \in R_X$, and $s: Y \rightarrow X$. Then*

$$\Theta_X(f) \circ s \subset \Theta_Y(f \circ s),\tag{G.77}$$

Proof. Let $\hat{f} \in \Theta_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$ such that $\hat{f} \leq cf + c$. This implies $(\hat{f} \circ s) \leq c(f \circ s) + c$. Therefore $(\hat{f} \circ s) \in \Theta_Y(f \circ s)$.

□

Theorem G.78 (Sub-homogeneity for Θ characterized).

$$f \Theta_X(g) \subset \Theta_X(fg) \iff f \in \Theta_X(fg),\tag{G.79}$$

for all $f, g \in R_X$.

Proof. Θ_X has order-consistency by [Order-consistency for Θ] (G.61).

Appendix G. Comparison of definitions

\implies

Suppose $f \notin \Theta_X(fg)$. Let $\widehat{g} \in R_X$ be such that $\widehat{g} = 1$. Then $f \in \Theta_X(f)$ by order-consistency, and $\widehat{g} \in \Theta_X(g)$, since $\widehat{g}(x) = 1 \leq 1g(x) + 1$, for all $x \in X$. Then $f\widehat{g} = f \notin \Theta_X(fg)$. Therefore $f \in \Theta_X(g) \not\subset \Theta_X(fg)$.

\impliedby

Suppose $f \in \Theta_X(fg)$ and $\widehat{g} \in \Theta_X(g)$. Then there exists $c \in \mathbb{R}^{>0}$ such that

$$\begin{aligned} f &\leq cfg + c \\ \widehat{g} &\leq cg + c. \end{aligned} \tag{G.80}$$

Therefore

$$\begin{aligned} f\widehat{g} &\leq f(cg + c) \\ &= cfg + cf \\ &\leq cfg + c(cfg + c) \\ &\leq (c^2 + c)fg + c^2 \\ &\leq (c^2 + c)fg + (c^2 + c). \end{aligned} \tag{G.81}$$

Therefore $f\widehat{g} \in \Theta_X(fg)$. \square

Theorem G.82 (Sub-homogeneity in \mathbb{N} fails for Θ). *There exists $f, g \in R_{\mathbb{N}^{>0}}$ such that $f(\mathbb{N}^{>0}) \subset \mathbb{N}^{>0}$, and*

$$f \in \Theta_{\mathbb{N}^{>0}}(g) \not\subset \Theta_{\mathbb{N}^{>0}}(fg). \tag{G.83}$$

Proof. Let $f \in R_{\mathbb{N}^{>0}}$ be such that $f(n) = n$, and $g \in R_{\mathbb{N}^{>0}}$ be such that $g(n) = 1/n$. Then

$$\begin{aligned} f(\lceil 3c \rceil) &= \lceil 3c \rceil \\ &> c + c \\ &= cf(\lceil 3c \rceil)g(\lceil 3c \rceil) + c, \end{aligned} \tag{G.84}$$

for all $c \in \mathbb{R}^{>0}$. Therefore $f \notin \Theta_X(fg)$, and so sub-homogeneity in \mathbb{N} fails by [Sub-homogeneity for Θ characterized] (G.78). \square

Appendix G. Comparison of definitions

Theorem G.85 (Sub-homogeneity in $1/\mathbb{N}^{>0}$ for Θ). *Let $X \in U$, and $f, u \in R_X$ be such that $\vec{u}(X) \subset 1/\mathbb{N}^{>0}$. Then*

$$u \Theta_X(f) \subset \Theta_X(uf). \quad (\text{G.86})$$

Proof. Let $\hat{f} \in \Theta_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$, such that

$$\hat{f} \leq cf + c. \quad (\text{G.87})$$

This implies

$$\begin{aligned} u\hat{f} &\leq u(cf + c) \\ &= c(uf) + uc \\ &\leq c(uf) + c. \end{aligned} \quad (\text{G.88})$$

Therefore $u\hat{f} \in \Theta_X(uf)$. □

Theorem G.89 (Extensibility for Θ). *Θ has extensibility.*

Proof. Suppose $f \in \Theta_X(g)$. Then there exists $c \in \mathbb{R}^{>0}$ such that

$$f \leq cg + c. \quad (\text{G.90})$$

Then

$$f \circ [\underline{X} \times Y] \leq c(g \circ [\underline{X} \times Y]) + c. \quad (\text{G.91})$$

Therefore Θ has extensibility. □

Theorem G.92 (Super-homogeneity fails for Θ). *There exists $f, g \in R_X$ such that*

$$f \Theta_X(g) \not\subset \Theta_X(fg). \quad (\text{G.93})$$

Proof. Let $f = 0$, and $\hat{h} \in R_X$ be such that $\hat{h} = 1$. Then $\hat{h} \leq 1fg + 1$, and so $\hat{h} \in \Theta_X(fg)$. For all $\hat{g} \in \Theta_X(g)$ it holds that $f\hat{g} = 0 \neq 1 = \hat{h}$. Therefore $f \Theta_X(g) \not\subset \Theta_X(fg)$. □

Appendix G. Comparison of definitions

Theorem G.94 (Super-multiplicativity for Θ). *Let $f, g \in R_X$. Then*

$$\Theta_X(f) \Theta_X(g) \supset \Theta_X(fg). \quad (\text{G.95})$$

Proof. Let $\widehat{h} \in \Theta_X(fg)$. Then there exists $c \in \mathbb{R}^{>0}$ such that

$$\widehat{h} \leq c f g + c. \quad (\text{G.96})$$

Let $\widehat{g} \in R_X$ be such that $\widehat{g}(x) = \max(g(x), 1)$. Since $\max(g(x), 1) \leq 1g(x) + 1$, it holds that $\widehat{g} \in \Theta_X(g)$. Then

$$\begin{aligned} \frac{\widehat{h}}{\widehat{g}} &\leq c f \frac{g}{\widehat{g}} + \frac{c}{\widehat{g}} \\ &\leq c f + c. \end{aligned} \quad (\text{G.97})$$

Let $\widehat{f} \in R_X$ be such that $\widehat{f} = \widehat{h}/\widehat{g}$. By the above, $\widehat{f} \in \Theta_X(f)$. In addition, $\widehat{f}\widehat{g} = \widehat{h}$. Therefore $\widehat{h} \in \Theta_X(f) \Theta_X(g)$. \square

Theorem G.98 (Power-homogeneity for Θ). *Let $X \in U$, $f \in R_X$, and $\alpha \in \mathbb{R}^{>0}$. Then*

$$\Theta_X(f)^\alpha = \Theta_X(f^\alpha). \quad (\text{G.99})$$

Proof. Θ has locality by [Locality for Θ] (G.66).

\subseteq

Let $\widehat{f} \in \Theta_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$ such that

$$\widehat{f} \leq c f + c. \quad (\text{G.100})$$

Let $\alpha \in \mathbb{R}^{>0}$. Since positive powers are increasing,

$$\begin{aligned} \widehat{f}^\alpha &\leq (c f + c)^\alpha \\ &= c^\alpha (f + 1)^\alpha. \end{aligned} \quad (\text{G.101})$$

Let $F = \{x \in X : f(x) \geq 1\}$. Then for $x \in F$,

$$\begin{aligned} (f(x) + 1)^\alpha &\leq (f(x) + f(x))^\alpha \\ &= 2^\alpha f(x)^\alpha. \end{aligned} \quad (\text{G.102})$$

Appendix G. Comparison of definitions

Let $\overline{F} = X \setminus F$. Then for $x \in \overline{F}$,

$$\begin{aligned} (f(x) + 1)^\alpha &\leq (1 + 1)^\alpha \\ &= 2^\alpha. \end{aligned} \tag{G.103}$$

Therefore,

$$\begin{aligned} f &\leq 2^\alpha(f^\alpha + 1) \\ &= 2^\alpha f^\alpha + 2^\alpha. \end{aligned} \tag{G.104}$$

It follows that

$$\widehat{f}^\alpha \leq c^\alpha 2^\alpha (f^\alpha + 1), \tag{G.105}$$

which shows that $\widehat{f}^\alpha \in \Theta_X(f^\alpha)$.

\supseteq

Let $\alpha \in \mathbb{R}^{>0}$, and $\widehat{g} \in \Theta_X(f^\alpha)$. Then there exists $c \in \mathbb{R}^{>0}$ such that

$$\widehat{g} \leq c f^\alpha + c. \tag{G.106}$$

Since positive powers are increasing,

$$\begin{aligned} \widehat{g}^{1/\alpha} &\leq (c f^\alpha + c)^{1/\alpha} \\ &= c^{1/\alpha} (f^\alpha + 1)^{1/\alpha}. \end{aligned} \tag{G.107}$$

Similarly to above, we can prove that $\widehat{g}^{1/\alpha} \in \Theta_X(f)$. Therefore, let $\widehat{f} \in R_X$ be such that $\widehat{f} = \widehat{g}^{1/\alpha}$. Then $\widehat{f} \in \Theta_X(f)$ and $\widehat{f}^\alpha = \widehat{g}$. Therefore $\widehat{g} \in \Theta_X(f)^\alpha$. \square

Theorem G.108 (Subset-sum rule for Θ in some cases). *Let $X, Y, Z \in U$, $S: X \rightarrow \widehat{\mathcal{P}}(Y \times Z)$, $a \in R_Z$, $f \in R_Y$, and $\widehat{f} \in O_Y(f)$. Then subset-summability holds for Θ if*

$$\exists M \in \mathbb{R}^{\geq 0}, \forall x \in X: \sum_{(y,z) \in S_x} a(z) \leq M. \tag{G.109}$$

Proof. Let $T: R_Y \rightarrow R_X$ such that

$$T(f) = \left(y \mapsto \sum_{(y,z) \in S_x} a(z) f(y) \right). \tag{G.110}$$

Appendix G. Comparison of definitions

There exists $c \in \mathbb{R}^{>0}$ such that $\widehat{f} \leq cf + c$. It follows that

$$\begin{aligned} T(\widehat{f}) &\leq cT(f) + c \sum_{(y,z) \in S_x} a(z) \\ &\leq cT(f) + cM \\ &\leq \max(cM, c)T(f) + \max(cM, c). \end{aligned} \tag{G.111}$$

Therefore $T(\widehat{f}) \in \Theta_Y(T(f))$, and so subset-summability holds. \square

Note G.112 (Subset-sum rule for Θ). It is open to us to characterize when exactly subset-summability holds for Θ . By [Affine containment] (G.117) we would expect subset-summability to fail, since subset-summability does not hold for Θ either. \blacksquare

Theorem G.113 (Zero-triviality fails for Θ). Θ does not have zero-triviality.

Proof. It holds that

$$1 \leq 1 \cdot 0 + 1. \tag{G.114}$$

Therefore $1 \in \Theta_{\mathbb{N}^{>0}}(0)$, and $\Theta_{\mathbb{N}^{>0}}(0) \neq \{0\}$. \square

G.7 Containment relations

Analysis of local linear dominance reveals that making the filter basis sets larger increases the number of fulfilled desirable properties — provided that at least one-separation is satisfied.

Trivial linear dominance has the smallest filter basis sets. It satisfies all of the desirable properties except those of non-triviality. Its worst defect is the failure of one-separation. In fact, every function in R_X is equivalent for every $X \in U$.

Asymptotic linear dominance has the next smallest filter basis sets. Its worst defect is the failure of sub-composability in \mathbb{N}^d .

Co-asymptotic linear dominance improves upon asymptotic linear dominance by satisfying injective sub-composability in \mathbb{N}^d . Its worst defects are the failure of sub-composability in \mathbb{N}^d , and the failure of injective sub-composability in \mathbb{Z} .

Appendix G. Comparison of definitions

Cofinite linear dominance improves upon coasymptotic linear dominance by satisfying injective sub-composability in all universes. Its worst defect is the failure of sub-composability in \mathbb{N} .

Linear dominance has the largest filter basis sets, and fulfills all of the desirable properties. We formalize the intuitive size-comparison of the filter basis sets by the following theorem.

Theorem G.115 (Containment in \mathbb{R}^d).

$$\Theta_X(f) \subset \Theta_X(f) \subset \Theta_X(f) \subset \Theta_X(f) \subset \Theta_X(f), \quad (\text{G.116})$$

for all $X \subset \mathbb{R}^d$ and $f \in R_X$.

Proof. Assume $d \in \mathbb{N}^{>0}$ and $X \subset \mathbb{R}^d$.

$$\underline{\Theta_X(f) \subset \Theta_X(f)}$$

Suppose $\hat{f} \in \Theta_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$ such that $\hat{f} \leq cf$. Since $X \in \bar{\mathcal{P}}(X)$, it holds that $(\hat{f}|X) \leq c(f|X)$. Therefore $\hat{f} \in \Theta_X(f)$.

$$\underline{\Theta_X(f) \subset \Theta_X(f)}$$

Suppose $\hat{f} \in \Theta_X(f)$. Then there exists $c \in \mathbb{R}^{>0}$ and $A \in \bar{\mathcal{P}}(X)$ such that $(\hat{f}|A) \leq c(f|A)$. Let $y = \sup_{\leq}(X \setminus A) + 1$. Then $(X \setminus X^{<y}) \subset A$, and so $(\hat{f}|(X \setminus X^{<y})) \leq c(f|(X \setminus X^{<y}))$. Therefore $\hat{f} \in \Theta_X(f)$.

$$\underline{\Theta_X(f) \subset \Theta_X(f)}$$

Suppose $\hat{f} \in \Theta_X(f)$. Then there exist $c \in \mathbb{R}^{>0}$ and $y \in \mathbb{R}^d$ such that $(\hat{f}|(X \setminus X^{<y})) \leq c(f|(X \setminus X^{<y}))$. Since $X^{\geq y} \subset (X \setminus X^{<y})$, it holds that $(\hat{f}|X^{\geq y}) \leq c(f|X^{\geq y})$. Therefore $\hat{f} \in \Theta_X(f)$.

$$\underline{\Theta_X(f) \subset \Theta_X(f)}$$

Suppose $\hat{f} \in \Theta_X(f)$. Then $\hat{f} \in \Theta_X(f)$, since every function is equivalent under trivial linear dominance. \square

Affine dominance has surprisingly good properties; we were especially surprised to be able to prove power-homogeneity for it. Its worst defect is the

Appendix G. Comparison of definitions

failure of sub-homogeneity. In terms of containment, affine dominance has smaller filter bases than cofinite linear dominance.

Theorem G.117 (Affine containment).

$$\Theta_X(f) \subset \Theta_X(f), \quad (\text{G.118})$$

for all $X \in U$ and $f \in R_X$.

Proof. Let $\hat{f} \in \Theta_X(f)$. Then there exists $A \in \overline{\mathcal{P}}(X)$ and $c \in \mathbb{R}^{>0}$ such that $\left(\hat{f}|_A\right) \leq c(f|_A)$. Since A is cofinite, let $d = \max\left(\hat{f}(X \setminus A) \cup \{c\}\right)$. Then $\hat{f} \leq df + d$. Therefore $\hat{f} \in \Theta_X(f)$. \square

Appendix H

Proofs of minimality

In this section we consider additional candidate definitions. These definitions are used — together with those in Appendix G — to prove the minimality of pre-primitive properties in Chapter 4.

H.1 Elementwise dominance

Definition H.1 (Elementwise dominance). *Elementwise dominance* Θ is defined by $f \in \Theta_X(g)$ if and only if

$$f \leq g, \tag{H.2}$$

for all $g \in R_X$, and all $X \in U$, where U is the class of all sets.

Theorem H.3 (Order-consistency for Θ). *Θ has order-consistency.*

Proof. By definition. □

Theorem H.4 (Transitivity for Θ). *Θ has transitivity.*

Proof. Let $f \in \Theta_X(g)$, $g \in \Theta_X(h)$, and $h \in R_X$. Then

$$\begin{aligned} f &\leq g, \\ g &\leq h. \end{aligned} \tag{H.5}$$

Appendix H. Proofs of minimality

Therefore

$$f \leq h. \tag{H.6}$$

□

Theorem H.7 (Scale-invariance fails for Θ). *Θ does not have scale-invariance.*

Proof. Let $f \in \Theta_X(g)$ be such that $f \neq 0$ and $\alpha \in \mathbb{R}^{>0}$ be such that $\alpha < 1$. Let $p \in X$ be such that $f(p) > 0$. Then

$$f(p) > \alpha f(p). \tag{H.8}$$

□

Theorem H.9 (Locality for Θ). *Θ has locality.*

Proof. Let $C \subset \mathcal{P}(X)$ be a finite cover of X , and suppose $(f|D) \in \Theta_X(g|D)$ for all $D \in C$. Then

$$(f|D) \leq (g|D), \tag{H.10}$$

for all $D \in C$. This implies

$$f \leq g. \tag{H.11}$$

□

Theorem H.12 (One-separation for Θ). *Θ has one-separation.*

Proof. It holds that $n > 1$ for $n = 2$.

□

Theorem H.13 (Sub-homogeneity for Θ). *Θ has sub-homogeneity.*

Proof. Let $f \in \Theta_X(g)$, and $g, u \in R_X$. Then

$$f \leq g. \tag{H.14}$$

This implies

$$uf \leq ug. \tag{H.15}$$

□

Theorem H.16 (Sub-composability for Θ). *Θ has sub-composability.*

Proof. Let $f \in \Theta_X(g)$, and $s: Y \rightarrow X$. Then

$$f \leq g. \quad (\text{H.17})$$

This implies

$$f \circ s \leq g \circ s. \quad (\text{H.18})$$

□

H.2 Multiple dominance

Definition H.19 (Multiple dominance). *Multiple dominance Θ is defined by $f \in \Theta_X(g)$ if and only if there exists $c \in \mathbb{R}^{>0}$ such that*

$$f = cg, \quad (\text{H.20})$$

for all $g \in R_X$, and all $X \in U$, where U is the class of all sets.

Theorem H.21 (Order-consistency fails for Θ). *Θ does not have order-consistency.*

Proof. Let $f, g \in R_{\mathbb{N}^{>0}}$ be such that $f = 0$ and $g = 1$. Then $f \leq g$, and

$$0 = f \neq cg = c, \quad (\text{H.22})$$

for all $c \in \mathbb{R}^{>0}$.

□

Theorem H.23 (Transitivity for Θ). *Θ has transitivity.*

Proof. Let $f \in \Theta_X(g)$, $g \in \Theta_X(h)$, and $h \in R_X$. Then there exists $c, d \in \mathbb{R}^{>0}$ such that

$$\begin{aligned} f &= cg, \\ g &= dh. \end{aligned} \quad (\text{H.24})$$

Therefore

$$f = cdh. \quad (\text{H.25})$$

□

Appendix H. Proofs of minimality

Theorem H.26 (Scale-invariance for \odot). *\odot has scale-invariance.*

Proof. Let $f \in \mathcal{O}_X(g)$, and $\alpha \in \mathbb{R}^{>0}$. Then there exists $c \in \mathbb{R}^{>0}$ such that

$$f = cg. \quad (\text{H.27})$$

This implies

$$f = c \frac{1}{\alpha} \alpha g. \quad (\text{H.28})$$

□

Theorem H.29 (Locality fails for \odot). *\odot does not have locality.*

Proof. Let $f \in R_{\mathbb{N}}$ be such that

$$f(n) = \begin{cases} 2, & n > 0, \\ 1, & n = 0. \end{cases} \quad (\text{H.30})$$

Let $g \in R_{\mathbb{N}}$ be such that $g = 4$. Then there is no $c \in \mathbb{R}^{>0}$ such that $f = cg$. □

Theorem H.31 (One-separation for \odot). *\odot has one-separation.*

Proof. It holds that $n > c$ for $n = \lceil c \rceil + 1$, for all $c \in \mathbb{R}^{>0}$. □

Theorem H.32 (Sub-homogeneity for \odot). *\odot has sub-homogeneity.*

Proof. Let $f \in \mathcal{O}_X(g)$, and $g, u \in R_X$. Then there exists $c \in \mathbb{R}^{>0}$ such that

$$f = cg. \quad (\text{H.33})$$

This implies

$$uf = cug. \quad (\text{H.34})$$

□

Appendix H. Proofs of minimality

Theorem H.35 (Sub-composability for \mathbb{O}). \mathbb{O} has sub-composability.

Proof. Let $f \in \mathbb{O}_X(g)$, and $s: Y \rightarrow X$. Then there exists $c \in \mathbb{R}^{>0}$ such that

$$f = cg. \quad (\text{H.36})$$

This implies

$$(f \circ s) = c(g \circ s). \quad (\text{H.37})$$

□

H.3 Non-transitive dominance

Definition H.38 (Non-transitive dominance). *Non-transitive dominance* \mathbb{O} is defined by $f \in \mathbb{O}_X(g)$ if and only if

$$(\exists c \in \mathbb{R}^{>0} : f = cg) \text{ or } (f \leq 2g), \quad (\text{H.39})$$

for all $g \in R_X$, and all $X \in U$, where U is the class of all sets.

Theorem H.40 (Order-consistency for \mathbb{O}). \mathbb{O} has order-consistency.

Proof. Let $f, g \in R_{\mathbb{N}^{>0}}$ be such $f \leq g$. Then

$$f \leq 2g. \quad (\text{H.41})$$

□

Theorem H.42 (Transitivity fails for \mathbb{O}). \mathbb{O} does not have transitivity.

Proof. Let $p \in X$, and $f, g, h \in R_X$ be such that $h = 1$, $g = 2$, and

$$f(x) = \begin{cases} 4 & x \neq p, \\ 3 & \text{otherwise.} \end{cases} \quad (\text{H.43})$$

Then

$$\begin{aligned} f &\leq 2g = 4 \\ g &\leq 2h = 2, \end{aligned} \quad (\text{H.44})$$

so that $f \in \mathbb{O}_X(g)$ and $g \in \mathbb{O}_X(h)$. However, $f(p) > 2h(p)$, and there is no $c \in \mathbb{R}^{>0}$ such that $f = ch$. Therefore $f \notin \mathbb{O}_X(h)$. □

Appendix H. Proofs of minimality

Theorem H.45 (Scale-invariance for \mathbb{O}). \mathbb{O} has scale-invariance.

Proof. Let $\alpha \in \mathbb{R}^{>0}$. Then

$$f = \frac{1}{\alpha} \alpha f. \quad (\text{H.46})$$

□

Theorem H.47 (Locality fails for \mathbb{O}). \mathbb{O} does not have locality.

Proof. Let $f, g \in R_{\mathbb{N}}$ be such that $g = 1$, and

$$f(n) = \begin{cases} 4, & n > 0, \\ 3, & n = 0. \end{cases} \quad (\text{H.48})$$

Then $(f|_{\{0\}}) = 3 = 3(g|_{\{0\}})$, and $(f|_{\mathbb{N}^{>0}}) = 4 = 4(g|_{\mathbb{N}^{>0}})$. Therefore $(f|_{\{0\}}) \in \mathbb{O}_{\{0\}}(g)$ and $(f|_{\mathbb{N}^{>0}}) \in \mathbb{O}_{\mathbb{N}^{>0}}(g|_{\mathbb{N}^{>0}})$. However, $f(0) > 2g(0)$, and there is no $c \in \mathbb{R}^{>0}$ such that $f = cg$. □

Theorem H.49 (One-separation for \mathbb{O}). \mathbb{O} has one-separation.

Proof.

$$\begin{aligned} & n \in \mathbb{O}_{\mathbb{N}^{>0}}(1) \\ \iff & (\exists c \in \mathbb{R}^{>0} : \forall n \in \mathbb{N}^{>0} : n = c) \text{ or } (\forall n \in \mathbb{N}^{>0} : n \leq 2), \end{aligned} \quad (\text{H.50})$$

which is false. □

Theorem H.51 (Sub-homogeneity for \mathbb{O}). \mathbb{O} has sub-homogeneity.

Proof. Let $f \in \mathbb{O}_X(g)$, and $g, u \in R_X$. Suppose there exists $c \in \mathbb{R}^{>0}$ such that

$$f = cg. \quad (\text{H.52})$$

Then

$$uf = cug. \quad (\text{H.53})$$

Suppose $f \leq 2g$. Then

$$uf \leq 2(ug). \quad (\text{H.54})$$

□

Appendix H. Proofs of minimality

Theorem H.55 (Sub-composability for \mathbb{O}). \mathbb{O} has sub-composability.

Proof. Let $f \in \mathbb{O}_X(g)$, and $g, u \in R_X$. Suppose there exists $c \in \mathbb{R}^{>0}$ such that $f = cg$. Then

$$f \circ s = c(g \circ s). \quad (\text{H.56})$$

Suppose $f \leq 2g$. Then

$$f \circ s \leq 2(g \circ s). \quad (\text{H.57})$$

□

H.4 Power dominance

Definition H.58 (Clamped power). The *clamped k -power*, where $k \in \mathbb{N}^{>0}$, is a function $\cdot^{\underline{k}}: \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$ such that

$$x^{\underline{k}} = \begin{cases} x^k & x \geq 1, \\ x & x < 1. \end{cases} \quad (\text{H.59})$$

Definition H.60 (Power dominance). *Power dominance* \mathbb{O} is defined by $f \in \mathbb{O}_X(g)$ if and only if there exists $c \in \mathbb{R}^{>0}$ and $k \in \mathbb{N}^{>0}$ such that

$$f \leq cg^{\underline{k}}, \quad (\text{H.61})$$

for all $g \in R_X$, and all $X \in U$, where U is the class of all sets.

Theorem H.62 (Order-consistency for \mathbb{O}). \mathbb{O} has order-consistency.

Proof. Suppose $f \leq g$. Then

$$f \leq 1g^{\underline{1}} = g. \quad (\text{H.63})$$

□

Appendix H. Proofs of minimality

Theorem H.64 (Transitivity for \mathbb{O}). \mathbb{O} has transitivity.

Proof. Let $f \in \mathbb{O}_X(g)$, $g \in \mathbb{O}_X(h)$, and $h \in R_X$. Then there exists $c \in \mathbb{R}^{>0}$ such that

$$\begin{aligned} f &\leq cg^{\frac{k}{c}}, \\ g &\leq ch^{\frac{k}{c}}. \end{aligned} \tag{H.65}$$

We may assume $c \geq 1$. Then

$$f \leq c(ch^{\frac{k}{c}})^{\frac{k}{c}}. \tag{H.66}$$

Small values

Suppose $x \in X$ is such that $ch(x)^{\frac{k}{c}} < 1$. Then

$$\begin{aligned} f(x) &\leq c^2 h(x)^{\frac{k}{c}} \\ &\leq c^{k+1} h(x)^{\frac{k^2}{c}}. \end{aligned} \tag{H.67}$$

Large values

Suppose $x \in X$ is such that $ch(x)^{\frac{k}{c}} \geq 1$ and $h(x) \geq 1$. Then

$$\begin{aligned} f(x) &\leq c(ch(x)^{\frac{k}{c}})^k \\ &= c^{k+1} h(x)^{k^2} \\ &\leq c^{k+1} h(x)^{\frac{k^2}{c}}. \end{aligned} \tag{H.68}$$

Suppose $x \in X$ is such that $ch(x)^{\frac{k}{c}} \geq 1$ and $h(x) < 1$. Then

$$\begin{aligned} f(x) &\leq c(ch(x)^{\frac{k}{c}})^k \\ &= c^{k+1} h(x)^k \\ &\leq c^{k+1} h(x)^{\frac{k^2}{c}}. \end{aligned} \tag{H.69}$$

□

Theorem H.70 (Scale-invariance for \mathbb{O}). \mathbb{O} has scale-invariance.

Appendix H. Proofs of minimality

Proof. Let $f \in R_X$ and $\alpha \in \mathbb{R}^{>0}$. Then

$$\begin{aligned} f &= \frac{1}{\alpha} \alpha f \\ &= \frac{1}{\alpha} (\alpha f)^{\frac{1}{\alpha}}. \end{aligned} \tag{H.71}$$

□

Theorem H.72 (Locality for \mathbb{O}). \mathbb{O} has locality.

Proof. Let $C \subset \mathcal{P}(X)$ be a finite cover of X , and suppose $(f|D) \in \mathbb{O}_X(g|D)$ for all $D \in C$. Then there exists $c_D \in \mathbb{R}^{>0}$ and $k_D \in \mathbb{N}^{>0}$ such that

$$(f|D) \leq c_D (g|D)^{\underline{k_D}}, \tag{H.73}$$

for all $D \in C$. Let $c = \max\{c_D : D \in C\}$ and $k = \max\{k_D : D \in C\}$. Then

$$f \leq c g^{\underline{k}}. \tag{H.74}$$

□

Theorem H.75 (One-separation for \mathbb{O}). \mathbb{O} has one-separation.

Proof. Let $c \in \mathbb{R}^{>0}$. Then $n > c 1^{\underline{k}}$ for $n = \lceil c \rceil + 1$. □

Lemma H.76 (Clamped power lemma).

$$u g^{\underline{k}} \leq (u g)^{\underline{k}}, \tag{H.77}$$

for all $g \in \mathbb{R}^{\geq 0}$, $u, k \in \mathbb{N}^{>0}$.

Proof.

$u g < 1$

Since $u \geq 1$, it holds that $g < 1$. Then

$$\begin{aligned} u g^{\underline{k}} &= u g \\ &= (u g)^{\underline{k}}. \end{aligned} \tag{H.78}$$

Appendix H. Proofs of minimality

$ug \geq 1$ and $g < 1$

Then

$$\begin{aligned} ug^{\underline{k}} &= ug \\ &\leq (ug)^k \\ &= (ug)^{\underline{k}}. \end{aligned} \tag{H.79}$$

$ug \geq 1$ and $g \geq 1$

Then

$$\begin{aligned} ug^{\underline{k}} &= ug^k \\ &\leq u^k g^k \\ &= (ug)^k \\ &= (ug)^{\underline{k}}. \end{aligned} \tag{H.80}$$

□

Theorem H.81 (Sub-homogeneity in \mathbb{N} for \mathbb{O}). *\mathbb{O} has sub-homogeneity in \mathbb{N} .*

Proof. Let $f \in \mathbb{O}_X(g)$, and $g, u \in R_X$ be such that $\vec{u}(X) \subset \mathbb{N}$. Then there exists $c \in \mathbb{R}^{>0}$ and $k \in \mathbb{N}^{>0}$ such that

$$f \leq cg^{\underline{k}}. \tag{H.82}$$

This implies

$$\begin{aligned} uf &\leq cug^{\underline{k}} \\ &\leq c(ug)^{\underline{k}}, \end{aligned} \tag{H.83}$$

where we used [Clamped power lemma] (H.76). □

Theorem H.84 (Sub-homogeneity in $1/\mathbb{N}^{>0}$ fails for \mathbb{O}). *\mathbb{O} does not have sub-homogeneity in $1/\mathbb{N}^{>0}$.*

Appendix H. Proofs of minimality

Proof. Let $f, g, u \in R_{\mathbb{N}^{>0}}$ be such that

$$\begin{aligned} f(n) &= n^2, \\ g(n) &= n, \\ u(n) &= n. \end{aligned} \tag{H.85}$$

Then $f \leq 1g^2$. Let $c \in \mathbb{R}^{>0}$ and $k \in \mathbb{N}^{>0}$. Then the inequality

$$n = f/u \leq d(g/u)^m = d \tag{H.86}$$

fails for $n = \lceil d \rceil + 1$. □

Theorem H.87 (Sub-composability for \odot). \odot has sub-composability.

Proof. Let $f \in \odot_X(g)$, and $s: Y \rightarrow X$. Then there exists $c \in \mathbb{R}^{>0}$ and $k \in \mathbb{N}^{>0}$ such that

$$f \leq cg^k. \tag{H.88}$$

This implies

$$(f \circ s) \leq c(g \circ s)^k. \tag{H.89}$$

□

Appendix I

Partitioned sets

In this section we develop some theory of partitioned sets. Partitioned sets occur in the theory of O-notation, because the equality of O-sets in a set X — $O_X(f) = O_X(g)$ — is an equivalence relation in R_X .¹

Definition I.1 (Partitioned set). A *partitioned set* is a set X with an associated equivalence $\sim_X: X \leftrightarrow X$ in X .

Note I.2 (Conventions). Let X and Y be partitioned sets. We will often shorten the word *partition* into a single letter p . ■

Definition I.3 (Partition-preserving). A function $f: X \rightarrow Y$ is *p-preserving* if

$$x_1 \sim_X x_2 \implies f(x_1) \sim_Y f(x_2), \quad (\text{I.4})$$

for all $x_1, x_2 \in X$.

Note I.5 (Homomorphisms). The p -preserving functions are the homomorphisms of partitioned sets; they preserve the partition structure. ■

¹An equivalence relation is a reflexive, symmetric, and transitive relation.

Appendix I. Partitioned sets

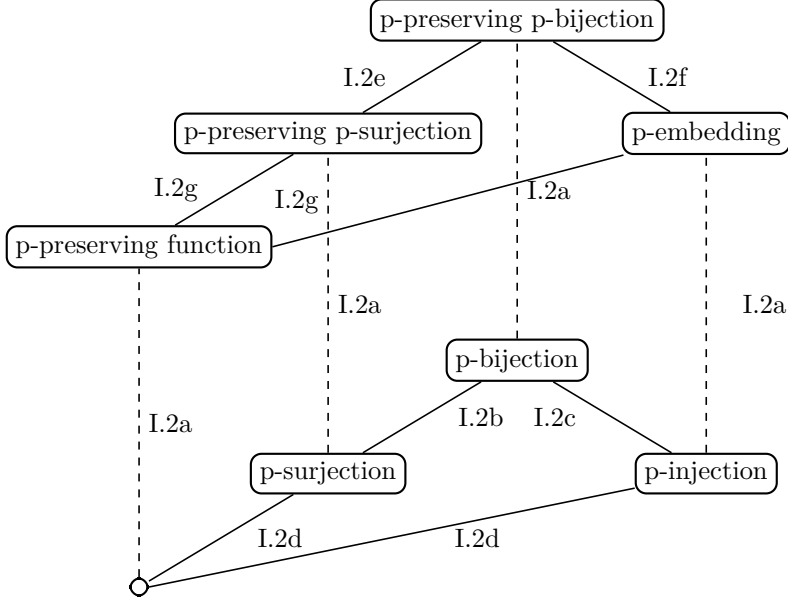


Figure I.1: A Hasse diagram of functions between partitioned sets, ordered by the ‘generalizes’ partial order. The edge labels — which refer to Figure I.2 — show that the generalizations are proper.

Definition I.6 (Partition closure). The p -closure on X is a function $\overline{\cdot} : \mathcal{P}(X) \rightarrow \mathcal{P}(X)$ such that

$$\overline{D} = \{x \in X : \exists d \in D : x \sim_X d\}. \quad (\text{I.7})$$

I.1 Generalized partition-inverse

Definition I.8 (Generalized partition-inverse). A generalized p -inverse of $f : X \rightarrow Y$ is $\hat{f} : Z \rightarrow X$ such that $\vec{f}(X) \subset Z \subset Y$, and

$$f(\hat{f}(f(x))) \sim_Y f(x), \quad (\text{I.9})$$

Appendix I. Partitioned sets

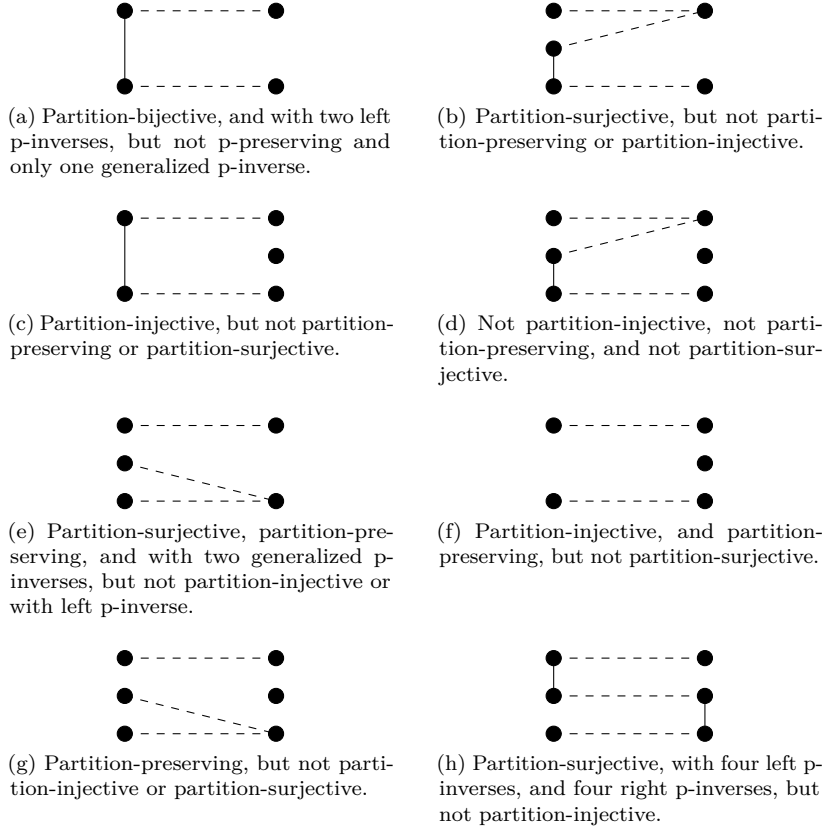


Figure I.2: Diagrams to show that the various definitions of functions on partitioned sets are not equivalent. Equivalent dots are connected with a solid line.

Appendix I. Partitioned sets

for all $x \in X$.

Theorem I.10 (Construction for a generalized p-inverse). *Let $f: X \rightarrow Y$. Then $\hat{f}: Z \rightarrow X$ is a generalized p-inverse of f if and only if*

$$\hat{f}(y) \in \overleftarrow{f}(\overline{\{y\}}), \quad (\text{I.11})$$

for all $y \in \overrightarrow{f}(X)$.

Proof. It holds that

$$\begin{aligned} \hat{f}(f(x)) &\in \overleftarrow{f}(\overline{\{f(x)\}}) \\ \iff f(\hat{f}(f(x))) &\in \overline{\{f(x)\}} \\ \iff f(\hat{f}(f(x))) &\sim_Y f(x), \end{aligned} \quad (\text{I.12})$$

for all $x \in X$. □

Note I.13 (Generalized partition-inverse exists). [Construction for a generalized p-inverse] (I.10) shows that a generalized p-inverse always exists. ■

Note I.14 (Generalized partition-inverse may not be unique). Figure I.2h shows that a function $f: X \rightarrow Y$ can have many non-equivalent generalized p-inverses. ■

I.2 Partition-injectivity and left partition-inverse

Definition I.15 (Partition-injectivity). A function $f: X \rightarrow Y$ is *p-injective*, if

$$f(x_1) \sim_Y f(x_2) \implies x_1 \sim_X x_2, \quad (\text{I.16})$$

for all $x_1, x_2 \in X$.

Appendix I. Partitioned sets

Definition I.17 (Left partition-inverse). A left p -inverse of $f: X \rightarrow Y$ is a function $f^L: Z \rightarrow X$ such that $\vec{f}(X) \subset Z \subset Y$ and

$$f^L(f(x)) \sim_X x, \quad (\text{I.18})$$

for all $x \in X$.

Theorem I.19 (Generalized partition-inverse is a left partition-inverse for a partition-injective function). Let $f: X \rightarrow Y$ be p -injective. Then a generalized p -inverse of f is a left p -inverse of f .

Proof. Let $\hat{f}: Z \rightarrow X$ be a generalized p -inverse of f . By definition,

$$f(\hat{f}(f(x))) \sim_Y f(x), \quad (\text{I.20})$$

for all $x \in X$. Since f is p -injective,

$$\hat{f}(f(x)) \sim_X x, \quad (\text{I.21})$$

for all $x \in X$. Therefore \hat{f} is a left p -inverse of f . □

Note I.22 (Generalized partition-inverse may not be a left partition-inverse). Figure I.2e shows that a generalized p -inverse may not be a left p -inverse. ■

Theorem I.23 (Left partition-inverse is a generalized partition-inverse for a partition-preserving function). Let $f: X \rightarrow Y$ be p -preserving. Then a left p -inverse of f is a generalized p -inverse of f .

Proof. Let $f^L: Z \rightarrow X$ be a left p -inverse of f . By definition,

$$f^L(f(x)) \sim_X x, \quad (\text{I.24})$$

for all $x \in X$. Since f is p -preserving,

$$f(f^L(f(x))) \sim_Y f(x), \quad (\text{I.25})$$

for all $x \in X$. Therefore f^L is a generalized p -inverse of f . □

Note I.26 (Left partition-inverse may not be a generalized partition-inverse). Figure I.2a shows a non-p-preserving function which has two left p-inverses, but only one generalized p-inverse; a left p-inverse may not be a generalized p-inverse. ■

Theorem I.27 (Partition-injectivity is equivalent to having a p-p-reserving left p-inverse). *Let $f: X \rightarrow Y$. Then f is p-injective if and only if f has a p-preserving left p-inverse.*

Proof.

\implies

Let $\hat{f}: \overrightarrow{f}(X) \rightarrow X$ be such that

$$\hat{f}(y) \in \overleftarrow{f}(\overline{\{y\}}). \quad (\text{I.28})$$

Then \hat{f} is a generalized p-inverse of f by [Construction for a generalized p-inverse] (I.10). Since f is p-injective, \hat{f} is a left p-inverse by [Generalized partition-inverse is a left partition-inverse for a partition-injective function] (I.19). It holds that

$$\begin{aligned} & x_1, x_2 \in \overleftarrow{f}(\overline{\{y\}}) \\ \implies & f(x_1), f(x_2) \in \overline{\{y\}} \\ \implies & f(x_1) \sim_Y y \sim_Y f(x_2) \\ \implies & x_1 \sim_X x_2, \quad \vdash f \text{ p-injective} \end{aligned} \quad (\text{I.29})$$

for all $x_1, x_2 \in X$ and $y \in \overrightarrow{f}(X)$. That is, all elements in $\overleftarrow{f}(\overline{\{y\}})$ are equivalent to each other, for all $y \in \overrightarrow{f}(X)$. Let $y_1, y_2 \in \overrightarrow{f}(X)$ be such that $y_1 \sim_Y y_2$. Then

$$\overline{\{y_1\}} = \overline{\{y_2\}}. \quad (\text{I.30})$$

This implies that

$$\hat{f}(y_1), \hat{f}(y_2) \in \overleftarrow{f}(\overline{\{y_1\}}). \quad (\text{I.31})$$

Appendix I. Partitioned sets

However, since all elements in $\overleftarrow{f}(\overline{\{y_1\}})$ are equivalent to each other,

$$\widehat{f}(y_1) \sim_X \widehat{f}(y_2). \quad (\text{I.32})$$

Therefore \widehat{f} is p-preserving.

\Leftarrow

Let $f^L: Z \rightarrow X$ be a p-preserving left p-inverse of f . Then

$$\begin{aligned} & f(x_1) \sim_Y f(x_2) \\ \implies & f^L(f(x_1)) \sim_X f^L(f(x_2)) && \vdash f^L \text{ p-preserving} \\ \implies & x_1 \sim_X x_2, && \vdash f^L \text{ left p-inverse of } f \end{aligned} \quad (\text{I.33})$$

for all $x_1, x_2 \in X$. Therefore f is p-injective. \square

Note I.34 (Non-partition-preserving left p-inverse). Figure I.2h shows that the existence of a non-p-preserving left p-inverse does not imply p-injectivity. ■

Theorem I.35 (Left p-inverses are equivalent on the image). *The left p-inverses of $f: X \rightarrow Y$ are equivalent to each other on $\overrightarrow{f}(X)$.*

Proof. Suppose $g: Z \rightarrow X$ and $h: W \rightarrow X$ are left p-inverses of f . For each $y \in \overrightarrow{f}(X)$, there exists $x_y \in X$ such that $y = f(x_y)$. Since g and h are left p-inverses,

$$g(y) = g(f(x_y)) \sim_X x_y \sim_X h(f(x_y)) = h(y), \quad (\text{I.36})$$

for all $y \in \overrightarrow{f}(X)$. \square

I.3 Partition-surjectivity and right partition-inverse

Definition I.37 (Partition-surjectivity). A function $f: X \rightarrow Y$ is *p-surjective*, if

$$\overline{\overrightarrow{f}(X)} = Y. \quad (\text{I.38})$$

Appendix I. Partitioned sets

Note I.39 (Right partition-inverse). A *right p-inverse* of $f: X \rightarrow Y$ is $f^R: Y \rightarrow X$ such that

$$f(f^R(y)) \sim_Y y, \quad (\text{I.40})$$

for all $y \in Y$. ■

Theorem I.41 (Partition-surjectivity is equivalent to having a right p-inverse). Let $f: X \rightarrow Y$. Then f is *p-surjective* if and only if f has a *right p-inverse*.

Proof.

\implies

Since f is p-surjective,

$$\overleftarrow{f}(\overline{\{y\}}) \neq \emptyset, \quad (\text{I.42})$$

for all $y \in Y$. Let $\hat{f}: Y \rightarrow X$ be such that

$$\hat{f}(y) \in \overleftarrow{f}(\overline{\{y\}}). \quad (\text{I.43})$$

Then

$$\begin{aligned} f(\hat{f}(y)) &\in \overline{\{y\}} \\ \iff f(\hat{f}(y)) &\sim_Y y, \end{aligned} \quad (\text{I.44})$$

for all $y \in Y$. Therefore \hat{f} is a right p-inverse of f .

\impliedby

Let $f^R: Y \rightarrow X$ be a right p-inverse of f . Then

$$f(f^R(y)) \sim_Y y, \quad (\text{I.45})$$

for all $y \in Y$. Therefore f is p-surjective. □

Theorem I.46 (Right p-inverse is order-reflecting for order-preserving). Let $f: X \rightarrow Y$ be order-preserving. Then a right p-inverse of f

Appendix I. Partitioned sets

is order-reflecting.

Proof. Let $f^R: Y \rightarrow X$ be a right p-inverse of f . Then

$$\begin{aligned} & f^R(y_1) \preceq_X f^R(y_2) \\ \implies & f(f^R(y_1)) \preceq_Y f(f^R(y_2)) && \vdash f \text{ order-preserving} \\ \implies & y_1 \preceq_Y y_2, && \vdash f^R \text{ right p-inverse of } f \end{aligned} \tag{I.47}$$

for all $y_1, y_2 \in Y$. Therefore f^R is order-reflecting. \square

I.4 Partition-bijectivity and partition-inverse

Definition I.48 (Partition-bijectivity). A function $f: X \rightarrow Y$ is *p-bijjective*, if it is both p-injective and p-surjective.

Definition I.49 (Partition-inverse). A *p-inverse* of f is a function $\widehat{f}: Y \rightarrow X$ which is both a left p-inverse and a right p-inverse of f .

Theorem I.50 (Partition-bijectivity is equivalent to having a p-p-reserving p-inverse). Let $f: X \rightarrow Y$. Then f is p-bijjective if and only if f has a p-preserving p-inverse.

Proof.

\implies

A p-preserving left p-inverse $f^L: \overrightarrow{f}(X) \rightarrow X$ of f exists by [Partition-injectivity is equivalent to having a p-preserving left p-inverse] (I.27). We may extend f^L to a p-preserving left p-inverse of f on $\overrightarrow{f}(X)$. Since f is p-surjective, $\overrightarrow{f}(X) = Y$. We then notice that the extended f^L satisfies the construction of the right p-inverse f^R in [Partition-surjectivity is equivalent to having a right p-inverse] (I.41).

\impliedby

This follows from [Partition-injectivity is equivalent to having a p-preserving

Appendix I. Partitioned sets

left p-inverse] (I.27) and [Partition-surjectivity is equivalent to having a right p-inverse] (I.41). \square

Theorem I.51 (Partition-bijectivity implies a unique p-inverse).

Let $f: X \rightarrow Y$ be p-bijective. Then f has a unique p-inverse up to an equivalence.

Proof. Since f is p-bijective, there exists a p-inverse $g: Y \rightarrow X$ of f by [Partition-bijectivity is equivalent to having a p-preserving p-inverse] (I.50). Suppose there exists another p-inverse $h: Y \rightarrow X$ of f . Then

$$\begin{aligned} f(g(y)) \sim_Y y \sim_Y f(h(y)) \\ \implies g(y) \sim_X h(y), \qquad \qquad \qquad \vdash f \text{ p-injective} \end{aligned} \tag{I.52}$$

for all $y \in Y$. \square

I.5 Partition-embedding

Definition I.53 (Partition-embedding). A function $f: X \rightarrow Y$ is *p-embedding*, if it is both p-preserving and p-injective.

Theorem I.54 (Partition-embedding by preimages).

Let $f: X \rightarrow Y$. Then f is a p-embedding if and only if

$$\overleftarrow{f} \left(\overline{\{f(x)\}} \right) = \overline{\{x\}}, \tag{I.55}$$

for all $x \in X$.

Proof.

\implies

Appendix I. Partitioned sets

It holds that

$$\begin{aligned}
 & z \in \overleftarrow{f} \left(\overline{\{f(x)\}} \right) \\
 \iff & f(z) \in \overline{\{f(x)\}} \\
 \iff & f(z) \sim_Y f(x) \\
 \iff & z \sim_X x && \vdash f \text{ p-embedding} \\
 \iff & z \in \overline{\{x\}},
 \end{aligned} \tag{I.56}$$

for all $x \in X$.

\Leftarrow

To show that f is p-embedding,

$$\begin{aligned}
 & x_1 \sim_X x_2 \\
 \iff & \overline{\{x_1\}} = \overline{\{x_2\}} \\
 \iff & \overleftarrow{f} \left(\overline{\{f(x_1)\}} \right) = \overleftarrow{f} \left(\overline{\{f(x_2)\}} \right) && \vdash \text{assumption} \\
 \iff & \overline{\{f(x_1)\}} = \overline{\{f(x_2)\}} && \vdash f \text{ p-embedding} \\
 \iff & f(x_1) \sim_Y f(x_2),
 \end{aligned} \tag{I.57}$$

for all $x_1, x_2 \in X$. □

Appendix J

Preordered sets

In this section we develop some theory of preorders, generalizing the theory of partial orders. Preordered sets occur in the theory of O-notation, because $O_X(f)$ is a principal down-set of a preorder in R_X .

Definition J.1 (Preordered set). A *preordered set* is a set X with an associated preorder $\preceq_X: X \leftrightarrow X$.

Note J.2 (Conventions). Let X and Y be preordered sets. ■

Definition J.3 (Order-preserving). A function $f: X \rightarrow Y$ is *order-preserving*, or *monotone*, if

$$x_1 \preceq_X x_2 \implies f(x_1) \preceq_Y f(x_2), \quad (\text{J.4})$$

for all $x_1, x_2 \in X$.

Note J.5 (Homomorphisms). The order-preserving functions are the homomorphisms of preordered sets; they preserve the preorder structure. ■

Appendix J. Preordered sets

Definition J.6 (Order-reflecting). A function $f: X \rightarrow Y$ is *order-reflecting*, if

$$f(x_1) \preceq_Y f(x_2) \implies x_1 \preceq_X x_2, \quad (\text{J.7})$$

for all $x_1, x_2 \in X$.

Definition J.8 (Order-embedding). A function $f: X \rightarrow Y$ is *order-embedding*, if it is both order-preserving and order-reflecting.

Definition J.9 (Order-isomorphism). A function $f: X \rightarrow Y$ is an *order-isomorphism*, if it is p-surjective and order-embedding.

The relationships between such functions is given in Figure J.1, with Figure J.2 showing that the inclusions are proper.

J.1 Order and partitions

Definition J.10 (Induced equivalence). Given a preorder $\preceq_X: X \leftrightarrow X$, the *induced equivalence* is $\sim_X: X \leftrightarrow X$ such that

$$x_1 \sim_X x_2 \iff x_1 \preceq_X x_2 \text{ and } x_2 \preceq_X x_1. \quad (\text{J.11})$$

Note J.12 (Relation to partitioned sets). The induced equivalence \sim_X on a preordered set (X, \preceq_X) partitions X ; the theory of partitioned sets interacts with the theory of preordered sets. ■

Theorem J.13 (Order-preserving is partition-preserving). *Let $f: X \rightarrow Y$ be order-preserving. Then f is p-preserving.*

Proof. Let $x_1, x_2 \in X$ be such that $x_1 \sim_X x_2$. Then

$$x_1 \preceq_X x_2 \implies f(x_1) \preceq_Y f(x_2). \quad (\text{J.14})$$

Similarly,

$$x_2 \preceq_X x_1 \implies f(x_2) \preceq_Y f(x_1). \quad (\text{J.15})$$

Appendix J. Preordered sets

It follows that

$$f(x_1) \sim_Y f(x_2). \quad (\text{J.16})$$

Therefore f is p-preserving. \square

Theorem J.17 (Order-reflecting implies partition-injective). *Let $f: X \rightarrow Y$ be order-reflecting. Then f is partition-injective.*

Proof. It holds that

$$\begin{aligned} & f(x_1) \sim_Y f(x_2) \\ \implies & f(x_1) \preceq_Y f(x_2) \text{ and } f(x_2) \preceq_Y f(x_1) \\ \implies & x_1 \preceq_X x_2 \text{ and } x_2 \preceq_X x_1 \\ \implies & x_1 \sim_X x_2, \end{aligned} \quad (\text{J.18})$$

for all $x_1, x_2 \in X$. \square

J.2 Order-preserving functions and down-sets

Definition J.19 (Generated down-set). The *generated down-set* in X is a function $\downarrow_X(\cdot): \mathcal{P}(X) \rightarrow \mathcal{P}(X)$ such that

$$\downarrow_X(D) = \{x \in X : \exists d \in D : x \preceq_X d\} \quad (\text{J.20})$$

Definition J.21 (Down-set). A subset $D \subset X$ is a *down-set* of X , if $D = \downarrow_X(D)$.

Definition J.22 (Principal down-set). A subset $D \subset X$ is a *principal down-set* of X , if there exists $d \in X$ such that $D = \downarrow_X(\{d\})$.

Theorem J.23 (Alternative definition of down-sets). *Let $D \subset X$.*

Appendix J. Preordered sets

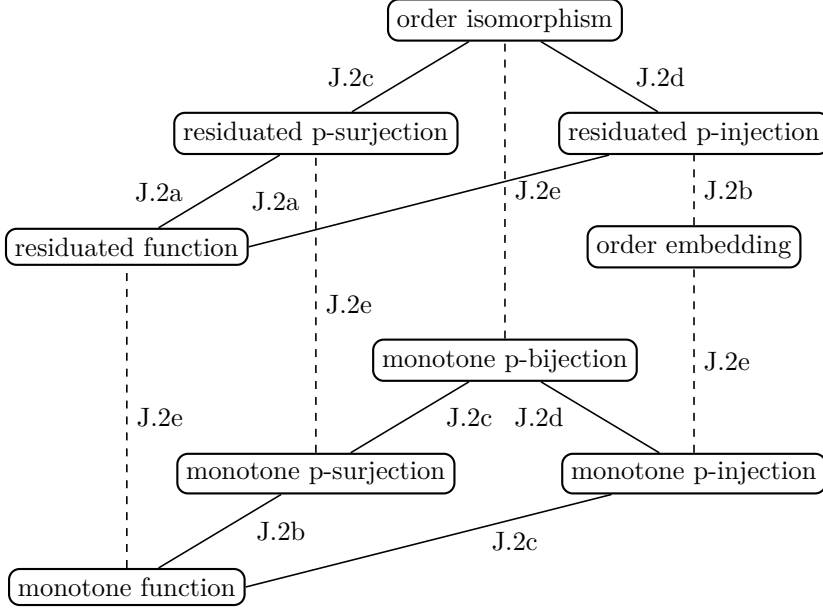


Figure J.1: A Hasse diagram of order-preserving functions between preordered sets, ordered by the ‘generalizes’ partial order. Original figure was created by David Wilding, and is used here with his permission. The edge labels — which refer to Figure J.2 — have been added to show that the generalizations are proper.

Then D is a down-set if and only if

$$\forall x \in X, \forall d \in D : x \preceq_X d \implies x \in D. \quad (\text{J.24})$$

Appendix J. Preordered sets

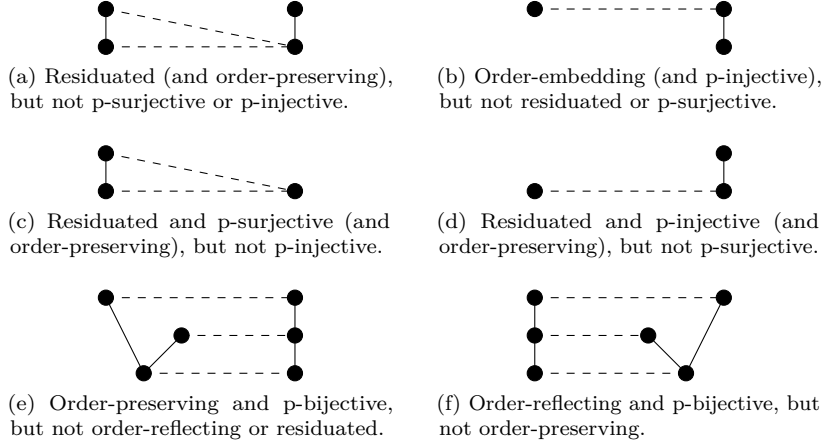


Figure J.2: Hasse diagrams to show that the various definitions of order-preserving functions are not equivalent. Since all orders here are partial, the p-prefix is redundant.

Proof. It always holds that $D \subset \{x \in X : \exists d \in D : x \preceq_X d\}$. Then

$$\begin{aligned}
 & D = \{x \in X : \exists d \in D : x \preceq_X d\} \\
 \iff & D \supset \{x \in X : \exists d \in D : x \preceq_X d\} \\
 \iff & \forall x \in X : (\exists d \in D : x \preceq_X d) \implies x \in D \\
 \iff & \forall x \in X : (\forall d \in D : x \not\preceq_X d) \text{ or } x \in D \\
 \iff & \forall x \in X, \forall d \in D : (x \not\preceq_X d \text{ or } x \in D) \\
 \iff & \forall x \in X, \forall d \in D : x \preceq_X d \implies x \in D.
 \end{aligned} \tag{J.25}$$

□

Theorem J.26 (Order-preservation by preimages). *Let $f: X \rightarrow Y$. Then f is order-preserving if and only if each preimage of a down-set of Y is a down-set of X .*

Proof.

Appendix J. Preordered sets

\implies

Let $S \subset Y$, $D_Y = \downarrow_Y(S)$, and $D_X = \overleftarrow{f}(D_Y)$. Let $x_1 \in X$, and $x_2 \in D_X$ be such that $x_1 \preceq_X x_2$. Since f is order-preserving, $f(x_1) \preceq_Y f(x_2)$. It holds that $f(x) \in D_Y$, for all $x \in D_X$. Since $f(x_1) \preceq_Y f(x_2) \in D_Y$, and D_Y is a down-set, $f(x_1) \in D_Y$. This implies $x_1 \in D_X$. Therefore D_X is a down-set.

\impliedby

Let $x_1, x_2 \in X$. Then there exists $D \subset X$ such that

$$\overleftarrow{f}\left(\downarrow_Y\left(\overrightarrow{f}(\{x_2\})\right)\right) = \downarrow_X(D). \quad (\text{J.27})$$

Let $x \in X$. Then

$$\begin{aligned} x &\in \overleftarrow{f}(\downarrow_Y(\{f(x_2)\})) \\ \iff f(x) &\in \downarrow_Y(\{f(x_2)\}) \\ \iff f(x) &\preceq_Y f(x_2). \end{aligned} \quad (\text{J.28})$$

In particular, $x_2 \in \overleftarrow{f}(\downarrow_Y(f(x_2))) = \downarrow_X(D)$. Then

$$\begin{aligned} x_1 &\preceq_X x_2 \\ \implies x_1 &\in \downarrow_X(D) \\ \implies x_1 &\in \overleftarrow{f}(\downarrow_Y(\{f(x_2)\})) \\ \implies f(x_1) &\in \downarrow_Y(\{f(x_2)\}) \\ \implies f(x_1) &\preceq_Y f(x_2). \end{aligned} \quad (\text{J.29})$$

Therefore f is order-preserving. □

Theorem J.30 (Order-preservation by images). *Let $f: X \rightarrow Y$. Then f is order-preserving if and only if*

$$\overrightarrow{f}(\downarrow_X(S)) \subset \downarrow_Y(\overrightarrow{f}(S)), \quad (\text{J.31})$$

for all $S \subset X$.

Proof.

Appendix J. Preordered sets

\implies

It holds that

$$\begin{aligned}
 & x \in \downarrow_X(S) \\
 \implies & \exists s \in S : x \preceq_X s \\
 \implies & \exists s \in S : f(x) \preceq_Y f(s) \quad \vdash f \text{ order-preserving} \\
 \implies & f(x) \in \downarrow_Y(\overrightarrow{f}(S)),
 \end{aligned} \tag{J.32}$$

for all $x \in X$.

\impliedby

It holds that

$$\begin{aligned}
 & x_1 \preceq_X x_2 \\
 \implies & \downarrow_X(\{x_1\}) \subset \downarrow_X(\{x_2\}) \\
 \implies & \overrightarrow{f}(\downarrow_X(\{x_1\})) \subset \overrightarrow{f}(\downarrow_X(\{x_2\})) \\
 \implies & \overrightarrow{f}(\downarrow_X(\{x_1\})) \subset \downarrow_Y(\{f(x_2)\}) \\
 \implies & f(x_1) \in \downarrow_Y(\{f(x_2)\}) \\
 \implies & f(x_1) \preceq_Y f(x_2),
 \end{aligned} \tag{J.33}$$

for all $x_1, x_2 \in X$. Therefore f is order-preserving. \square

J.3 Residuated functions

Definition J.34 (Residual). A function $\widehat{f}: Y \rightarrow X$ is a *residual* of $f: X \rightarrow Y$, if

$$f(x) \preceq_Y y \iff x \preceq_X \widehat{f}(y), \tag{J.35}$$

for all $x \in X$ and $y \in Y$.

Definition J.36 (Residuated). A function $f: X \rightarrow Y$ is *residuated*, if it has a residual.

Theorem J.37 (Preimage of a down-set under a residuated function). *Let $f: X \rightarrow Y$ be residuated, with a residual $\widehat{f}: Y \rightarrow X$. Then*

$$\overleftarrow{f}(\downarrow_Y(D)) = \downarrow_X(\widehat{f}(D)), \quad (\text{J.38})$$

for all $D \subset Y$.

Proof.

$$\begin{aligned} \overleftarrow{f}(\downarrow_Y(D)) &= \{x \in X : f(x) \in \downarrow_Y(D)\} \\ &= \{x \in X : \exists d \in D : f(x) \preceq_Y d\} \\ &= \{x \in X : \exists d \in D : x \preceq_X \widehat{f}(d)\} \\ &= \{x \in X : \exists x' \in \widehat{f}(D) : x \preceq_X x'\} \\ &= \downarrow_X(\widehat{f}(D)). \end{aligned} \quad (\text{J.39})$$

□

Theorem J.40 (Residuated property by preimages). *Let $f: X \rightarrow Y$. Then f is residuated if and only if for each $y \in Y$, there exists $x_y \in X$ such that*

$$\overleftarrow{f}(\downarrow_Y(\{y\})) = \downarrow_X(\{x_y\}). \quad (\text{J.41})$$

Proof.

\implies

Let f be residuated, and $y \in Y$. Then

$$\overleftarrow{f}(\downarrow_Y(\{y\})) = \downarrow_X(\{\widehat{f}(y)\}), \quad (\text{J.42})$$

by [Preimage of a down-set under a residuated function] (J.37). Therefore each preimage of a principal down-set of Y is a principal down-set of X .

\impliedby

Appendix J. Preordered sets

For each $y \in Y$, there exists $x_y \in X$ such that

$$\begin{aligned}
 & \overleftarrow{f}(\downarrow_Y(\{y\})) = \downarrow_X(\{x_y\}) \\
 \iff & \{x \in X : f(x) \in \downarrow_Y(\{y\})\} = \{x \in X : x \preceq_X x_y\} \\
 \iff & \{x \in X : f(x) \preceq_Y y\} = \{x \in X : x \preceq_X x_y\} \\
 \iff & \forall x \in X : [f(x) \preceq_Y y \iff x \preceq_X x_y].
 \end{aligned} \tag{J.43}$$

Let $\widehat{f}: Y \rightarrow X$ be such that $\widehat{f}(y) = x_y$. By the above, \widehat{f} is a residual of f . Therefore f is residuated. \square

Theorem J.44 (Properties of a residuated function). *Let $f: X \rightarrow Y$. Then f is residuated if and only if f is order-preserving, and there exists order-preserving $\widehat{f}: X \rightarrow Y$, such that*

$$\begin{aligned}
 & \widehat{f}(f(x)) \succeq_X x, \\
 & f(\widehat{f}(y)) \preceq_Y y,
 \end{aligned} \tag{J.45}$$

for all $x \in X, y \in Y$.

Proof.

\implies

For the first relation,

$$\begin{aligned}
 & f(x) \preceq_Y f(x) \\
 \iff & x \preceq_X \widehat{f}(f(x)), \quad \vdash \widehat{f} \text{ residual of } f
 \end{aligned} \tag{J.46}$$

for all $x \in X$. Then

$$\begin{aligned}
 & x_1 \preceq_X x_2 \\
 \implies & x_1 \preceq_X \widehat{f}(f(x_2)) \quad \vdash \text{first relation} \\
 \implies & f(x_1) \preceq_Y f(x_2), \quad \vdash \widehat{f} \text{ residual of } f
 \end{aligned} \tag{J.47}$$

for all $x_1, x_2 \in X$. Therefore f is order-preserving.

For the second relation,

$$\begin{aligned}
 & \widehat{f}(y) \preceq_X \widehat{f}(y) \\
 \iff & f(\widehat{f}(y)) \preceq_Y y, \quad \vdash \widehat{f} \text{ residual of } f
 \end{aligned} \tag{J.48}$$

Appendix J. Preordered sets

for all $y \in Y$. Then

$$\begin{aligned}
 & y_1 \preceq_Y y_2 \\
 \implies & f(\widehat{f}(y_1)) \preceq_Y y_2 && \vdash \text{second relation} \\
 \implies & \widehat{f}(y_1) \preceq_X \widehat{f}(y_2), && \vdash \widehat{f} \text{ residual of } f
 \end{aligned} \tag{J.49}$$

for all $y_1, y_2 \in Y$. Therefore \widehat{f} is order-preserving.

\Leftarrow

It holds that

$$\begin{aligned}
 & f(x) \preceq_Y y \\
 \implies & \widehat{f}(f(x)) \preceq_X \widehat{f}(y) && \vdash \widehat{f} \text{ order-preserving} \\
 \implies & x \preceq_X \widehat{f}(y) && \vdash \text{first relation} \\
 \implies & f(x) \preceq_Y f(\widehat{f}(y)) && \vdash f \text{ order-preserving} \\
 \implies & f(x) \preceq_Y y, && \vdash \text{second relation}
 \end{aligned} \tag{J.50}$$

for all $x \in X, y \in Y$. Therefore \widehat{f} is a residual of f . \square

Theorem J.51 (Residual is essentially unique). *Let $g, h: Y \rightarrow X$ both be residuals of $f: X \rightarrow Y$. Then*

$$g(y) \sim_X h(y), \tag{J.52}$$

for all $y \in Y$.

Proof. By definition,

$$\begin{aligned}
 & g(y) \preceq_X h(y) \\
 \iff & f(g(y)) \preceq_Y y,
 \end{aligned} \tag{J.53}$$

for all $y \in Y$. The latter holds by [Properties of a residuated function] (J.44). Similarly,

$$h(y) \preceq_X g(y), \tag{J.54}$$

for all $y \in Y$. \square

J.4 Residuated functions and supremum

Definition J.55 (Upper-bound). An element $s \in X$ is an *upper-bound* of $S \subset X$, if

$$x \preceq_X s, \quad (\text{J.56})$$

for all $x \in S$.

Definition J.57 (Least upper-bound). An element $s \in X$ is a *least upper-bound* of $S \subset X$, if s is an upper-bound of S , and for any upper-bound $t \in X$ of S ,

$$s \preceq_X t. \quad (\text{J.58})$$

Definition J.59 (Supremum). The *supremum* of a set in a set X is a function $\sup_X: \mathcal{P}(X) \rightarrow \mathcal{P}(X)$, such that

$$\sup_X(S) = \{s \in X : s \text{ is a least upper-bound of } S\}. \quad (\text{J.60})$$

Theorem J.61 (Residuated property by suprema). Let $f: X \rightarrow Y$. Then f is residuated if and only if f is order-preserving,

$$\sup_X \left(\overleftarrow{f}(\downarrow_Y(\{y\})) \right) \neq \emptyset, \quad (\text{J.62})$$

for all $y \in Y$, and

$$\overrightarrow{f}(\sup_X(S)) \subset \sup_Y(\overrightarrow{f}(S)), \quad (\text{J.63})$$

for all $S \subset X$.

Proof.

\implies Order-preservation

The function f is order-preserving by [Properties of a residuated function] (J.44).

Appendix J. Preordered sets

\implies Non-emptiness

By [Residuated property by preimages] (J.40), for each $y \in Y$, there exists $x_y \in X$, such that

$$\overleftarrow{f}(\downarrow_Y(\{y\})) = \downarrow_X(\{x_y\}). \quad (\text{J.64})$$

It follows that

$$\sup_X \left(\overleftarrow{f}(\downarrow_Y(\{y\})) \right) = \overline{\{x_y\}} \neq \emptyset. \quad (\text{J.65})$$

\implies Upper-bound

Let $s \in \sup_X(S)$. Since s is an upper-bound of S ,

$$x \preceq_X s, \quad (\text{J.66})$$

for all $x \in S$. Since f is order-preserving,

$$f(x) \preceq_Y f(s), \quad (\text{J.67})$$

for all $x \in S$. Therefore $f(s)$ is an upper-bound of $\overrightarrow{f}(S)$.

\implies Least upper-bound

Let $\widehat{f}: Y \rightarrow X$ be a residual of f , and $y \in Y$ be an upper-bound of $\overrightarrow{f}(S)$. Then

$$f(x) \preceq_Y y, \quad (\text{J.68})$$

for all $x \in S$. Since f is residuated,

$$x \preceq_X \widehat{f}(y), \quad (\text{J.69})$$

for all $x \in S$. That is, $\widehat{f}(y)$ is an upper-bound of S . Since s is a *least* upper-bound of S ,

$$s \preceq_X \widehat{f}(y). \quad (\text{J.70})$$

Since f is residuated,

$$f(s) \preceq_Y y. \quad (\text{J.71})$$

Therefore $f(s)$ is a least upper-bound of $\overrightarrow{f}(S)$;

$$\overrightarrow{f}(\sup_X(S)) \subset \sup_X \left(\overrightarrow{f}(S) \right). \quad (\text{J.72})$$

Appendix J. Preordered sets

\Leftarrow Order-preservation

Let $\widehat{f}: Y \rightarrow X$ be such that

$$\widehat{f}(y) \in \sup_X \left(\overleftarrow{f}(\downarrow_Y(\{y\})) \right). \quad (\text{J.73})$$

The function \widehat{f} is well-defined, since by assumption

$$\sup_X \left(\overleftarrow{f}(\downarrow_Y(\{y\})) \right) \neq \emptyset, \quad (\text{J.74})$$

for all $y \in Y$. Let $y_1, y_2 \in Y$ be such that $y_1 \preceq_Y y_2$. Since supremum is an upper-bound,

$$\begin{aligned} & \forall x \in \overleftarrow{f}(\downarrow_Y(\{y_2\})) : x \preceq_X \widehat{f}(y_2) \\ \implies & \forall x \in X : \left[f(x) \preceq_Y y_2 \implies x \preceq_X \widehat{f}(y_2) \right] \\ \implies & \forall x \in X : \left[f(x) \preceq_Y y_1 \implies x \preceq_X \widehat{f}(y_2) \right] \\ \implies & \forall x \in \overleftarrow{f}(\downarrow_Y(\{y_1\})) : x \preceq_X \widehat{f}(y_2). \end{aligned} \quad (\text{J.75})$$

That is, $\widehat{f}(y_2)$ is also an upper-bound of $\overleftarrow{f}(\downarrow_Y(\{y_1\}))$. Since $\widehat{f}(y_1)$ is a *least* upper-bound of $\overleftarrow{f}(\downarrow_Y(\{y_1\}))$,

$$\widehat{f}(y_1) \preceq_X \widehat{f}(y_2). \quad (\text{J.76})$$

Therefore \widehat{f} is order-preserving.

\Leftarrow Deflation

Let $y \in Y$. By assumption,

$$\begin{aligned} f(\widehat{f}(y)) & \in \overrightarrow{f} \left(\sup_X \left(\overleftarrow{f}(\downarrow_Y(\{y\})) \right) \right) \\ & \subset \sup_Y \left(\overrightarrow{f} \left(\overleftarrow{f}(\downarrow_Y(\{y\})) \right) \right) \\ & \subset \sup_Y (\downarrow_Y(\{y\})) \\ & = \overline{\{y\}}. \end{aligned} \quad (\text{J.77})$$

Therefore

$$f(\widehat{f}(y)) \preceq_Y y. \quad (\text{J.78})$$

Appendix J. Preordered sets

\Leftarrow Inflation

Since $x \in \overset{\leftarrow}{f}(\downarrow_Y(\{f(x)\}))$,

$$x \preceq_X \widehat{f}(f(x)), \quad (\text{J.79})$$

for all $x \in X$. Therefore f is residuated with residual \widehat{f} by [Properties of a residuated function] (J.44). \square

Theorem J.80 (Residual by supremum). *Let $f: X \rightarrow Y$ be residuated. Then $\widehat{f}: Y \rightarrow X$ is a residual of f if and only if*

$$\widehat{f}(y) \in \sup_X \left(\overset{\leftarrow}{f}(\downarrow_Y(\{y\})) \right), \quad (\text{J.81})$$

for all $y \in Y$.

Proof. This is shown by [Residuated property by suprema] (J.61) and [Residual is essentially unique] (J.51). \square

J.5 Residuated functions and p-inverses

Theorem J.82 (Residual is a generalized p-inverse). *Let $f: X \rightarrow Y$ be residuated. Then*

$$\begin{aligned} f(\widehat{f}(f(x))) &\sim_Y f(x), \\ \widehat{f}(f(\widehat{f}(y))) &\sim_X \widehat{f}(y), \end{aligned} \quad (\text{J.83})$$

for all $x \in X, y \in Y$.

Proof. Let $x \in X$. It holds that

$$f(\widehat{f}(f(x))) \preceq_Y f(x), \quad (\text{J.84})$$

by [Properties of a residuated function] (J.44). By the same theorem,

$$\widehat{f}(f(x)) \succeq_X x. \quad (\text{J.85})$$

Since f is order-preserving by [Properties of a residuated function] (J.44),

$$f(\widehat{f}(f(x))) \succeq_Y f(x). \quad (\text{J.86})$$

Similarly for the second equivalence. \square

Theorem J.87 (Residual is a right p-inverse for p-surjection). *Let $f: X \rightarrow Y$ be a p-surjection. Then a residual of f is a right p-inverse of f .*

Proof. The function f is p-preserving by [Properties of a residuated function] (J.44). A right p-inverse f^R of f exists by [Partition-surjectivity is equivalent to having a right p-inverse] (I.41). Let $\hat{f}: Y \rightarrow X$ be a residual of f . Then

$$\begin{aligned}
 & y \preceq_Y y \\
 \implies & f(f^R(y)) \preceq_Y y && \vdash \text{right p-inverse of } f \\
 \implies & f^R(y) \preceq_X \hat{f}(y) && \vdash f \text{ residuated} \\
 \implies & f(f^R(y)) \preceq_Y f(\hat{f}(y)) && \vdash f \text{ order-preserving} \\
 \implies & y \preceq_Y f(\hat{f}(y)) && \vdash \text{right p-inverse of } f,
 \end{aligned} \tag{J.88}$$

for all $y \in Y$. By [Properties of a residuated function] (J.44),

$$f(\hat{f}(y)) \preceq_Y y, \tag{J.89}$$

for all $y \in Y$. Therefore,

$$f(\hat{f}(y)) \sim_Y y, \tag{J.90}$$

for all $y \in Y$. That is, \hat{f} is a right p-inverse of f . \square

Note J.91. A residuated p-surjection may have many non-equivalent right p-inverses; a residual is a specific version of a right p-inverse. \blacksquare

Theorem J.92 (Residual is a left p-inverse for residuated p-injection). *Let $f: X \rightarrow Y$ be a residuated p-injection. Then a residual of f is a left p-inverse of f .*

Proof. This follows from [Residual is a generalized p-inverse] (J.82) and [Generalized partition-inverse is a left partition-inverse for a partition-injective function] (I.19). \square

Theorem J.93 (Residual is a p-inverse for residuated p-bijection).

Appendix J. Preordered sets

Let $f: X \rightarrow Y$ be a residuated p -bijection. Then a residual of f is a p -inverse of f .

Proof. This follows from [Residual is a left p -inverse for residuated p -injection] (J.92) and [Residual is a right p -inverse for p -surjection] (J.87). \square

Theorem J.94 (Transpose-residuated p -surjective function preserves down-sets). *Let $f: X \rightarrow Y$ be \succeq_X -residuated and p -surjective. Then*

$$\overrightarrow{f}(\downarrow_X(\{S\})) = \downarrow_Y(\overrightarrow{f}(S)), \quad (\text{J.95})$$

for all $S \subset X$.

Proof.

\subseteq

The function f is \succeq_X -order-preserving by [Properties of a residuated function] (J.44), which is the same as \preceq_X -order-preserving. The result follows from [Order-preservation by images] (J.30).

\supseteq

Let $\widehat{f}: Y \rightarrow X$ be a \succeq_X -residual of f , which is a right p -inverse of f by [Residual is a right p -inverse for p -surjection] (J.87). Then

$$\begin{aligned} & y \in \downarrow_Y(\overrightarrow{f}(S)) \\ \implies & \exists x \in S : y \preceq_Y f(x) && \vdash \text{definition of down-set} \\ \implies & \exists x \in S : \widehat{f}(y) \preceq_X x && \vdash \widehat{f} \text{ is } \succeq_X\text{-residual of } f \\ \implies & \widehat{f}(y) \in \downarrow_X(S) && \vdash \text{definition of down-set} \\ \implies & f(\widehat{f}(y)) \in \overrightarrow{f}(\downarrow_X(S)) \\ \implies & y \in \overrightarrow{f}(\downarrow_X(S)), && \vdash \widehat{f} \text{ right } p\text{-inverse of } f \end{aligned} \quad (\text{J.96})$$

for all $S \subset X$. \square

Theorem J.97 (Transpose-residuated surjective function preserves down-sets). *Let $f: X \rightarrow Y$ be \succeq_X -residuated and surjective. Then*

$$\vec{f}(\downarrow_X(\{S\})) = \downarrow_Y(\vec{f}(S)), \quad (\text{J.98})$$

for all $S \subset X$.

Proof. The proof is the same as [Transpose-residuated p-surjective function preserves down-sets] (J.94), except that now we can choose the residual to be both a right inverse and right p-inverse, and use the fact that

$$y = f(\hat{f}(y)). \quad (\text{J.99})$$

□

J.6 Residuated functions and order-embeddings

Theorem J.100 (Residuated p-injection is order-embedding). *Let $f: X \rightarrow Y$ be a residuated p-injection. Then f is order-embedding.*

Proof. Let $\hat{f}: Y \rightarrow X$ be a residual of f . Since f is p-injective, \hat{f} is a left p-inverse of f by [Residual is a left p-inverse for residuated p-injection] (J.92). Then

$$\begin{aligned} & f(x_1) \preceq_Y f(x_2) \\ \implies & x_1 \preceq_X \hat{f}(f(x_2)) && \vdash \hat{f} \text{ residual of } f \\ \implies & x_1 \preceq_X x_2, && \vdash \hat{f} \text{ left p-inverse of } f \end{aligned} \quad (\text{J.101})$$

for all $x_1, x_2 \in X$. Therefore f is order-reflecting. The function f is order-preserving by [Properties of a residuated function] (J.44). Therefore f is order-embedding. □

Theorem J.102 (Residuated p-bijection is an order isomorphism). *Let $f: X \rightarrow Y$. Then f is an order isomorphism if and only if f is a residuated p-bijection.*

Proof.

Appendix J. Preordered sets

\implies

Since f is order-embedding, f is p-injective by [Order-reflecting implies partition-injective] (J.17). Therefore f is p-bijective. Let $f^P: Y \rightarrow X$ be a p-preserving p-inverse of f , which exists by [Partition-bijection is equivalent to having a p-preserving p-inverse] (I.50). Then

$$\begin{aligned} & f(x) \preceq_Y y \\ \iff & f(x) \preceq_Y f(f^P(y)) && \vdash f^P \text{ p-inverse of } f \\ \iff & x \preceq_Y f^P(y), && \vdash f \text{ p-reflecting} \end{aligned} \tag{J.103}$$

for all $x \in X$ and $y \in Y$. Therefore f is a residuated p-bijection.

\Leftarrow

This follows from [Residuated p-injection is order-embedding] (J.100) and because f is p-surjective. \square