III Formulating

> It is easy to study laboratory practices because they are so heavily equipped, so evidently collective, so obviously material, so clearly situated in specific times and spaces, so hesitant and costly. But the same is not true of mathematical practices: notions like … "calculating," "formalism," "abstraction" resist being shifted from the role of indisputable resources to that of inspectable and accountable topics. … We seem to be inevitably contaminated by [these notions], as if abstraction has rendered us abstract as well!
>
> —Latour (2008, 444)

We are not out of the woods yet. We may have a clearer idea about the whys and wherefores of *ground-truthing* (part I) and *programming* (part II), yet we still lack, at this point of the inquiry, one activity that is sometimes crucial to the formation of algorithms in computer science laboratories. Without accounting for these practices, I could only propose an extremely partial constitution of algorithms.

One way to become sensitive to the "missing mass" of our inquiry could be to look at a recent academic paper in computer science. And why not choose the subfield of image processing since it is the empirical ground of this ethnographic venture? While browsing, for example, through a paper entitled "Learning Deep Features for Discriminative Localization" (Zhou et al. 2016), we would encounter many things we are now familiar with. We would read about a specific problem (localizing class-specific image regions) that, according to the paper's authors, is solved satisfactorily by means of a computer program they call CAM, which stands for "class activation mapping." We would see that the problem, CAM, and what this program should retrieve all derive from an already-assembled ground truth (in this case, ImageNet Large Scale Visual Recognition Challenge [ILSVRC] 2014) that has been split into two parts: a training set and an evaluation set. We would also feel, behind the printed words and numbers, the long and fastidious computer programming episodes that were necessary to provide and discuss the paper's results. After all, if the authors did not write lists of instructions capable of triggering electric pulses in meaningful ways, they could not have provided any statistical evaluations of their algorithm's performances.

However, while browsing through this academic paper that presents and tries to convince us about the relevance of a new image-processing algorithm, we would very quickly bump into cryptic passages such as this one:

> By plugging $Fk = \sum x,y\ fk\ (x, y)$ into the class score, Sc, we obtain
>
> $$Sc = \sum_k w_k^c \sum_{x,y} fk\,(x,y) = \sum_{x,y} \sum_k w_k^c fk\,(x,y) \qquad (1)$$
>
> We define Mc as the class activation map for class c, where each spatial element is given by
>
> $$Mc(x,y) = \sum_k w_k^c fk\,(x,y) \qquad (2)$$
>
> Thus, $Sc = \sum_{x,y} Mc\ (x,y)$, and hence $Mc\ (x,y)$ directly indicates the importance of the activation at spatial grid (x,y) leading to the classification of an image to class c. (Zhou et al. 2016, 2923)

Such sentences that mix English words with combinations of Greek and Latin letters divided by equals signs are indeed widely used by computer scientists when they communicate about their algorithms in academic journals. Of course, as grown-up readers, we immediately understand that such an excerpt deals with *mathematics* and that (1) and (2) are proper *formulas* (or *equations* once their variables are replaced by numerical values). But if we only consider the descriptive system developed so far in this inquiry, we have no grip on these mathematical inscriptions. The conceptual apparatus of the inquiry enables us to deal with graphs and numeric values as they refer somehow to both data and targets as defined by ground truths. The inquiry's apparatus also enables us to deal with lines of code as they refer to numbered lists of instructions that trigger electric pulses in desired ways. But what about mathematical formulas? Where do they come from? Why do computer scientists need them, and how are they assembled? At this point, I do not have any other choice. In this last and important part III, I will have to consider the role of mathematics in the formation of algorithms.

The road I am about to take is dangerous; one second of inattention and my action-oriented method will be lost. For intricate reasons that I will cover, mathematical entities such as "theorems," "proofs," or "formulas" are indeed extremely resistant to empirical considerations; even though they certainly are the products of situated activities, they are often considered fundamental ingredients of thoughts. This tenacious habit is frequently the starting point of a downward spiral, itself leading to grand questions such as: "Are mathematics the expressions of abstract structures or individual consciousness?" So many innocent souls have been consumed by such floating interrogations! To avoid digging my own grave in this cemetery of practice, I will have to be extremely cautious and process one small step at a time. But with some patience, the construction of mathematical knowledge as well as its further enrollment in the formation of algorithms may be partially accounted for. Altogether, these efforts to define *formulating practices* will allow me to link both ground-truthing practices (necessary to establish the terms of solvable problems) and programming practices (necessary to make computers compute in desired ways). Within the present constituent effort, what we tend to call "algorithms" may then be described as uncertain products of (at least) these three interrelated activities.

As in part II—and largely for similar reasons—I will require operationalization efforts before diving into ethnographic materials. I will first have to put aside the vast majority of studies on mathematics. Too many topics, too many studies, too many methods; without preliminary cleaning efforts, dealing with mathematics in an action-oriented way is doomed to fail. As we shall see in chapter 5, the only way not to duck will be to start (almost) afresh, from very basic observations and hypotheses. Progressively, these hypotheses—well inspired by several STS on mathematics—will make us realize that mathematical entities such as "theorems," "proofs," or "formulas" are quite akin to more familiar scientific facts. If mathematical knowledge is often considered the expression of some superior reality, it might only be due to its extreme combinability. Once the vascularization of mathematics is put forward, we will realize that its indubitable power also comes from the humble instruments and actions that make nonmathematical topics *mathematicable*. This important point will, in turn, allow me to define *formulating practices* as the empirical process of merging networks that sustain given domains of activity with networks that sustain certified mathematical knowledge. In chapter 6, I will account for a small yet successful formulating effort that took place within the Lab. This third and last case study will underline the centrality of certified mathematical knowledge for the progressive formation of algorithms as it both forces the refinement of ground truths and unfolds scenarios for further programming episodes. It will also allow me to consider recent issues related to machine learning and artificial intelligence in an unconventional way. The last section of chapter 6 will be a brief summary.

This is a section of doi:10.7551/mitpress/12517.001.0001

The Constitution of Algorithms

Ground-Truthing, Programming, Formulating

By: Florian Jaton

Citation:

The Constitution of Algorithms: Ground-Truthing, Programming, Formulating
By: Florian Jaton
DOI: 10.7551/mitpress/12517.001.0001
ISBN (electronic): 9780262363235
Publisher: The MIT Press
Published: 2021

The open access edition of this book was made possible by generous funding and support from Arcadia – a charitable fund of Lisbet Rausing and Peter Baldwin

The MIT Press

The open access edition of this book was made possible by generous funding from Arcadia—a charitable fund of Lisbet Rausing and Peter Baldwin.

ARCADIA
A charitable fund of Lisbet Rausing and Peter Baldwin

This book was set in Stone Serif and Stone Sans by Westchester Publishing Services.

Library of Congress Cataloging-in-Publication Data

Names: Jaton, Florian, author. | Bowker, Geoffrey C., writer of foreword.
Title: The constitution of algorithms : ground-truthing, programming, formulating / Florian Jaton ; foreword by Geoffrey C. Bowker.
Description: Cambridge, Massachusetts : The MIT Press, [2020] | Series: Inside technology | Includes bibliographical references and index.
Identifiers: LCCN 2020028166 | ISBN 9780262542142 (paperback)
Subjects: LCSH: Algorithms--Case studies. | Computer programming--Case studies. | Algorithms--Social aspects. | Mathematics--Philosophy.
Classification: LCC QA9.58 .J38 2020 | DDC 518/.1--dc23
LC record available at https://lccn.loc.gov/2020028166