# Preface

Nowadays, data are more easily accessible than ever, yet support for deriving interesting consequences from base data is often unavailable, too expensive, or too technical for many users. For example, a student may have access to prerequisite listings and expected offering dates of courses but have no way to sieve through possible course sequences unless the college provides a dedicated tool. Similarly, an investor may know the instruments held in his or her mutual fund portfolio but have no easy way to unravel them and reveal his exposure to a specific industry or company. As an additional example, a secretary in a midsize company will typically need to juggle clients and transportation schedules by hand into an itinerary for her boss's forthcoming business trip. In all cases, manually inferring useful information from raw data is time-consuming and error-prone, a situation that often results in bad decisions, suboptimal plans, or missed opportunities.

The problem is not the lack of automated tools: databases are all about relating data, financial service firms will happily use their software to dissect one's portfolio in any desired way, and a plethora of scheduling and workflow applications are available for purchase. Yet individuals, small and medium enterprises, and even small departments in large companies rarely avail themselves of such products: even when the purchase price is affordable, there is an associated financial and cognitive training cost, often quite large. Indeed, the secretary may be unknowingly querying databases all day long, but setting one up may go well beyond her training. There is no simple and general application that empowers users to compute useful inferences on raw data.

This book addresses this problem by drawing inspiration from a type of automated data inference that is immensely popular: the spreadsheet. Applications such as Microsoft Excel and Google Spreadsheet are readily available and allow users to routinely perform complex custom calculations on numerical data. The spreadsheet's clever interface makes it easy to use productively with little or no training and its gentle learning curve allows users to reinforce their skills and acquire new ones in a learning-by-doing fashion. However, none of the above data manipulation problems is expressible in today's spreadsheets. The approach investigated here remedies this situation by extending the spreadsheet paradigm to enable

users to define useful forms of inference among their data. An implementation of this idea will allow the student, for example, to download data about course prerequisites and offerings into his favorite spreadsheet and write a "formula" that will calculate all possible course sequences. The investor will similarly be able to see the individual stocks in his portfolio and determine his actual exposure, and the secretary will have the means to generate workable itineraries and easily choose the least expensive or most convenient. Differently from related efforts, our approach focuses on retaining the cognitive simplicity of the traditional spreadsheet while making these extended functionalities available: like charts in Excel, for example, they should be unobtrusive when not needed and should be intuitively usable when needed. Unlike charts, the proposed extension touches many of the fundamental mechanisms underlying the spreadsheet architecture.

We call this extended paradigm the *deductive spreadsheet*, which we contrast with the *traditional* spreadsheets in use today. This book describes an abstract design for the deductive spreadsheet. As such, it is not tied to any specific commercial application, although we draw examples and motivations from several, and it has not yet been implemented. We adopt an interdisciplinary approach to developing this idea. The design of the deductive support of this application, the part that sits under the hood and that the user rarely thinks about, weaves together techniques and results from several fields within computer science and computational logic, in particular programming language theory, logic programming, databases, proof theory, and model theory. The parallel design of the user interface, the aspect that the user does see and will use to access these extended functionalities, combines research from cognitive psychology and some of the more experimental areas of computer science, in particular human-computer interaction. The result is not only the blueprint for a useful productivity tool but may be more importantly a balanced integration of methodologies from fields that rarely intersect in academic research (although they do in industry).

Expanding somewhat on the design, the internal extension underlying the deductive spreadsheet centers around interpreting a group of columns and rows as a relation, something that users of a traditional spreadsheet commonly do already each time they organize data in tabular form. The key idea is then to provide support for manipulating relations as first-class objects, in particular, to allow us to write formulas that compute to relations and to extend the core mechanisms for evaluation, update, and explanation to these new entities. As a language for relational formulas, we design a variant of the logic programming language Datalog. Datalog has been studied for many years in the context of deductive databases and is now applied in far broader domains, it has excellent computational properties and good expressiveness, and it comes with a number of algorithms that are compatible with a spreadsheet context. We extend its syntax to support the embedding of traditional spreadsheet formulas and to make it usable in a wide range of practical circumstances. These logical formulas are written in cell ranges and, upon evaluation, fill this range with the set of records that satisfy them. In order to do so, they interpret other areas of the spreadsheet as relations and combine portions of those records into their own result. The addition of Datalog-like formulas

significantly extends the expressive power of the traditional spreadsheet, enabling useful classes of new problems to be solved in a spreadsheet environment—in particular all the examples mentioned above.

On the outward-facing side, the user interface of the deductive spreadsheet is designed as a conservative extension of the mostly excellent interface of typical commercial spreadsheets. Indeed, the traditional spreadsheet owes much of its success to a user interface that has evolved to provide intuitive access to the underlying functionalities. In particular, it offers consistent interaction modalities across features and a gentle learning curve that allows users to progress through exposure and simple experimentation. By carefully leveraging recent methodologies from cognitive science, specifically the attention investment model and the cognitive dimensions of notation, we engineered a user interface for the deductive spreadsheet that retains the very same level of usability of current tools. The deductive functionalities are indeed barely noticeable for users who have no need for them and yet accessible in a natural way for those who need their added expressiveness. The new deductive components are made available through intuitive extensions of all common gestures and other standard interaction methodologies. Furthermore, the deductive infrastructure is seamlessly integrated into the traditional spreadsheet so that users not only still have access to the usual functionalities but are able to use them as part of logical inferences.

Looking forward, a future implementation of the design in this book could act as an automated assistant for the daily reasoning and decision-making needs of computer users, in the same way as the traditional spreadsheet assists them every day with calculations simple and complex. It will enable users without formal training in logic or computer science to interactively define logical rules in the same way as they define numerical formulas in today's spreadsheet. This deductive spreadsheet application targets the same segment of computer users as the traditional spreadsheet, estimated at over 50 million users. It has the same potential to boost productivity and help people with their daily tasks that made the traditional spreadsheet so successful.

The book is divided into two main parts: the first half focuses on the deductive engine that underlies this application, the foundations that users do not see. After giving a mathematical model of traditional spreadsheet applications, it extends them with operators to perform a number of relational tasks, akin to the user view of a database but in a spreadsheet context. Expressing this extension in a logic programming framework is a natural stepping stone toward giving it powerful deductive capabilities. The second half of the book deals with the user interface, the part of the application that the user actually interacts with. It reviews the elements of the graphical user interface of traditional spreadsheet applications and describes practical methodologies for designing user interfaces from the field of cognitive psychology. It then proposes a design that conservatively integrates mechanisms for a user to take advantage of the new deductive capabilities. This is followed by the results of some preliminary usability experiments.

The book is expected to appeal to researchers and practitioners in the various areas underlying this work. Researchers will not only find interesting new

developments in most of these areas but also learn how to reach out to adjoining and distant fields to achieve a multidisciplinary focus. Practitioners will find fully developed solutions to numerous problems that are not solvable using a traditional spreadsheet application or not easily solvable with them. They will then be able to either enter them in a forthcoming deductive spreadsheet application or to use the provided insight to program them.

Doha, Qatar
August 2012

Iliano Cervesato

http://www.springer.com/978-3-642-37746-4

The Deductive Spreadsheet
Cervesato, I.
2013, XXI, 406 p. 98 illus., Hardcover
ISBN: 978-3-642-37746-4