International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

Making research methodologies intheoretical computing explicit

Ian Sandersa , Colin Pilkingtona , Laurette Pretoriusa
a School of Computing, University of South Africa

ABSTRACT
All research should be guided by a process that begins with the researcher's philosophical world view and then details why the research has value, how the research was done, and why the particular approach was used. In addition, this process should be made known to the research community. In this paper we argue that theoretical or applied computing research does not normally conform to this common approach of being explicit about the research approach used but that it should do so. We adopt a pragmatic world view and use an argumentation method within a philosophical analysis design to support our position. We then adopt an accepted conceptual model of research methodology structure and use two examples to show that it can be applied across the field of computing. In addition, we propose a set of questions that could guide researchers in assessing the validity of their research. The model and the questions can be introduced to postgraduate students and developing researchers to assist them in planning and reporting their research. In addition to the practical contribution made by the model and questions, this paper adds to the debate about the place of research methodologies in computer science research.

Keywords: research methodology, research design, research methods, conceptual model, computing

INTRODUCTION

Although the term research can have many different meanings determined by the context in which it is used, academic research can been seen as systematic inquiry, based on experience and reasoning, seeking to discover truth, solutions to problems, and new conclusions, leading to a greater understanding of our world (Oates, 2006; O'Leary, 2017; Olivier, 1999; Walliman, 2001). Doing good research involves drawing on other academics' work to focus one's own study and develop research questions or identify research aims; formulating an appropriate strategy to answer the questions or to achieve the aim; doing the research; and then reporting on the outcomes of the study (Mouton, 2001; Oates, 2006). Choosing an appropriate strategy

for, or approach to, addressing the research problem (questions and/or aims), often termed the research methodology, is a crucial part of doing research that is accepted in the academic community (O'Leary, 2017).

We believe that all research should follow a rigorous process that clarifies the researcher's standpoint and details why the research has value, how the research was done, and why the particular approach was used, and that this process should be made known to the research community. This is in agreement with Derntl (2014) who, in his article on writing for publication, is clear that the how of doing the research (methods, materials, theories) should be explicitly stated in the body of a research article. Furthermore, we concur with Thorne (2016) that academic researchers need to have a solid grasp of "why" the research adds to a body of knowledge and disciplinary understanding, and that just because there is a gap in the field does not mean that it necessarily has to be studied and filled.

In this paper we argue that all PhD graduates should have a broad understanding of re- search methodologies and deeper knowledge in the specific areas that are relevant to their own field.

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

This means that such researchers should know about the most important philosoph- ical world views and how a chosen view influences their research. They should understand research designs and be able to argue the strengths and weaknesses of these. In addition, they should be able to discuss the methods that are often associated with different designs. In our experience, from many years of reviewing and examining, researchers and students in comput- ing, especially those in more theoretical or applied areas, tend not to report on the research methodology they have used in their work in articles or theses. In the more socially oriented research areas, reporting on one's methodology is expected and sometimes even required. It is our belief that the methodologies used in theoretical or applied computing research should be more visible, if only as a guide to others new to the field and present a possible approach that can be used. We believe that explicitly presenting and arguing about one's research methodology will get the "philosophy" back into PhDs in theoretical or applied computing research and make for better future academics rather than technicians who simply carry out projects.
In Section 2 we clarify our use of terms, set the context of our research, and further develop our argument. In Section 3 we present the conceptual model we have selected for this paper and show how typical theoretical or applied research projects may employ this model and be given authenticity by an explicit statement of the research approach used (Section 4). Section 5 discusses the contributions of this paper.

1 CONTEXT AND MOTIVATION

1.1 The computing disciplines

The Association of Computing Machinery (ACM) defines computing as "any goal-oriented activity requiring, benefiting from, or creating computers" (Shackelford et al., 2005, p. 9). Computing as a discipline covers a wide range of topics/subjects and the ACM overview re- port divides the computing field into five broad disciplines – computer engineering, computer science, information systems, information technology, and software engineering (Shackelford et al., 2005). In this paper we adopt the relatively common South African categorisation of information systems (IS) and computer science (CS) as being the two "arms" of the computing discipline and which are often located in two different departments (sometimes in different faculties) in universities (Sanders & Alexander, 2015). IS could be considered to cover the more socially oriented aspects of the discipline while CS is generally more technical and science oriented. We do not specifically discuss other fields in the discipline (for example, Software Engineering or Information Technology) – we mainly differentiate between the more social side and the more theoretical side and leave it for the interested reader to determine where their own sub-discipline resides. We also acknowledge that researchers in the computing field are not necessarily locked into their areas of research – some CS researchers do socially ori- ented research as well as theoretical or applied research and some IS researchers do research which is theoretical or applied. For convenience, in the remainder of the paper, when we refer to CS research we mean the theoretical or applied types of research more commonly done by computer scientists.

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

Researchers in computing work in diverse fields and publish in a broad range of conferences and journals. Olivier (1999) notes that there are three main strategies that could be used in computing research: novel ways of combining or blending current disparate knowledge into a more coherent whole; finding a solution to a problem for which there is no current solution; and finding a better solution to a problem that has already been solved, or solved incompletely. He further sees the quality of this research in the "beauty" of the solution – its "simplicity, originality and generality" (Olivier, 1999, p. 4).

IS researchers typically study how information systems are designed, developed, adopted and used by individuals, groups, companies, institutions, and society in general (Oates, 2006). CS research, in contrast, reflects its origin in both formal approaches like mathematics and technical ones like engineering. Hartmanis (1993) notes that CS is different from other sciences and argues that theories in CS are used to develop models, measures and evaluation methods for various topics. CS research is also about efficiency and optimality, and has a "how" motivation rather than a "why" motivation. Ramesh et al. (2004) studied 628 CS research papers over a five-year period and concluded that CS research generally focuses on five major topic areas: problem-solving concepts, computer concepts, systems/software concepts, data/information concepts and problem-domain-specific concepts. The majority of papers they studied focused on formulating a process, method, or an algorithm. This largely agrees with Hartmanis's stance.

1.2 Research methodologies in computing

As there is no universally agreed upon understanding of what research methodology is, it is necessary that we be clear on what we mean, especially as it relates to the conceptual model that we discuss in section 3. A research methodology can be seen as the conceptual work that forms the logic of a research endeavour – the research goal and questions, the contribution to the discipline's knowledge, which approaches, designs, and methods are to be used, and why these are appropriate (Verschuren & Doorewaard, 2010). It, thus, provides the justification or rationale for the purpose of the research and the choices concerning research designs and methods. How the research is then actually implemented forms the more technical part of a research endeavour.

In IS research projects, methodology is usually explicit and generally expected to form a key part of a research article. There are many online resources (see, for example Association for Information Systems (2019)), books (Oates, 2006; Olivier, 1999; Whitman & Woszczynski, 2004; Willcocks et al., 2015, 2016), and research publications (Galliers & Stein, 2018; Myers, 1997, 2018; Nunamaker Jr et al., 2015; Petter et al., 2018; Walsham, 2018) that elucidate or debate the nature of IS research and how it is performed. However, in most CS research articles/theses, methodology is often implicit and scarcely visible, and sometimes completely ignored. Holz et al. (2006, Table 1, p. 99) searched for "research method" in research art- icles published by the ACM SIGs and show that research areas on the "social science" side (for example, management information systems and information technology education) of the computing discipline are more likely to use that term than research areas on the other extreme (for example, computer architecture, operating systems, algorithms and computation theory). This rather limited study seems to validate the current expectation in the computing research domain that research in the more theoretical fields does not require (or even benefit explicit statement of the research methodology utilised.

CS has developed research methodologies that borrow from the two broad fields of math-

ematics and engineering (Demeyer, 2011). Hassani (2017) ascribes the fact that CS does not have well defined and accepted methods for research to its relative youth as a discipline together with its extensive overlap with other disciplines. It should be noted that while there are many computing research methodology textbooks (for example, Oates (2006)) these tend to concentrate on IS type research projects with much less concentration on theoretical or applied computing type projects.

Connolly (2020) recently argued that computing is moving in the direction of the more methodologically pluralist social sciences due to the ubiquity of computing in everyday life, and that computing research is expanding beyond traditional computing fields into areas such as health care, education, and social computing. Such a multidisciplinary approach is going to require a broader recognition and understanding of how research may be conducted in these wider fields of enquiry, which will improve computing research approaches.

The contrast between how research methodologies are handled in IS and CS research is reflected in the training of students in the two disciplines. In current IS student dissertations, theses, or research proposals, an explicit discussion (at least a chapter) of the chosen research methodology is required (Mouton, 2001). Many CS theses have no such chapter (see for example Ade-Ibijola (2016), Motara (2018) and Sanders (2002) and many other theses1).

Furthermore, although it is common to teach research methods in the social sciences, it is less common in the computer sciences (Punyabukkana, 2017). There are many advocates for research methodology courses for IS students but fewer for CS students. For example, in 2004, Caplinskas and Vasilecas (2004) argued that IS was a developing discipline and that students should be taught about research methodologies so that appropriate discussions of the methodologies that they had used could be included in their theses. As the output from an ACM ITiCSE 2006 working group, Holz et al. (2006) presents a concept map for the teaching of Computer Research Methods (CRM). They suggest that formalising CRM could be a useful starting point for developing a methodology course, but even this structure seems to favour IS-type research. More recently, Håkansson (2013, p. 67) states that in the computer sciences "methods and methodologies are commonly neglected, or avoided, and regarded as 'necessarily vicious'." She argues that, if research methodologies are applied, this is often done after the research is conducted and that students misunderstand and misapply them. The portal she has developed aims to assist students conducting research or degree projects to choose appropriate methods and methodologies to steer their work to achieve proper, correct and well-founded results. In addition, Hassani (2017) advocates that CS researchers and lecturers should continue to polish their ideas on methodologies and to share these ideas with students and emerging researchers.

1.3 Local context

In the School of Computing at the University of South Africa (Unisa) we have both IS and CS students, although there are many more IS students. Postgraduate students have to complete a compulsory one year proposal module before being allowed to register for their thesis. The documentation for the Master's and Doctoral proposal module is largely focused on a preliminary literature review to find a research gap and associated research questions as well as the appropriate research methodologies for answering the research questions. All students are expected to produce a proposal with a "research methodology" chapter, however supervisors of theoretical or applied computing topics often do not require it of their students. The effect of the difference in the way research methodologies are taught in IS and CS means that CS students

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
(and in some cases, their supervisors) often do not know how to deal with these requirements. Sometimes, the students, with limited guidance from their super- visors, try to define research methodologies without a real understanding of those used in the discipline. The benefits of using accepted theories for developing an approach, which incor- porates a philosophical world view, research design and appropriate methods, towards doing
the research are lost, and students essentially begin their research projects "rudderless".
Our experience, as external examiners, shows that the tendency to ignore methodologies in

research work that is deemed to be "more theoretical" is not only prevalent at our university. The absence of an explicit statement of research methodologies in CS and how they are used to structure and validate one's research, can be seen in dissertations, theses and research papers from other institutions as well. Some students manage to produce adequate or even good research, in spite of the seeming lack of a considered approach, but other students produce documents that show that they do not really understand the process of doing research.

1.4 Motivation

We agree with other researchers (Caplinskas & Vasilecas, 2004; Demeyer, 2011; Håkansson, 2013; Holz et al., 2006) that it is important that all computing PhD students should be taught about research methodologies and encouraged to be explicit about the use of these in their work. As additional support for our argument, we note that many guidelines to examiners of Master's dissertations and PhD theses include a statement that says the dissertation/thesis must show that candidates are acquainted with the methods or techniques of research (University of KwaZulu-Natal, 2020; University of South Africa, 2014; University of the Western Cape, 2019). These guidelines sometimes expand this statement to say that candidates "should understand the nature and purpose of their investigations" (University of KwaZulu-Natal, 2020, p. 1). We believe that this means that universities require their candidates to know about and apply relevant methodologies in their research. Students who understand the various philosophies and approaches that can be adopted in doing research and are able to articulate their own philosophies and approaches in their documents are much more likely to do good research. They are also more likely to be able to convince examiners that they meet the requirements for the degree.
Other drivers for the explicit statement of a research methodology for any research are the increasing demand for researchers to consider the ethical implications of their research, the increasing prevalence of multidisciplinary research (Connolly, 2020), and issues relating to reproducibility in research (where it is not only the data that has to be available but also exactly how it was used – the methodology).

1.5 The philosophy of research

Our view is that CS research should be formalised by rigorous approaches that guide, and enable, researchers to do quality research. Researchers should also be able to argue about the validity and relevance of their research. As much as Thorne decried the loss of a solid philosophical understanding of research in the nursing field in her editorial entitled "PhD without the Ph?", we believe there may be a similar loss in CS. Prather et al. (2009) argue for putting the "Ph" back into "PhD" in ecological sciences. They say that PhD students sometimes lose sight of how their research fits into the current theories of the discipline and pose a number

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

of questions that they believe students should bear in mind in the course of their PhD work. Inspired by these researchers we have defined our own list of questions to guide CS researchers, especially PhD students, in doing good research.

- What explicit theoretical dilemma and conceptual underpinning (not just a gap) in CS has inspired my research?
- How would truth (and/or fact) be evaluated in the research?
- Which concepts/models/assumptions from the discipline are integrated in my methodology?
- What are the implications of my results for the wider field of CS? How do they relate to this wider field? In which scope (space, time, and processes) could the results apply?
- How do my results challenge or support existing approaches to truth/fact and theory?

We believe that if CS students (and researchers) develop a good understanding of research methodology, then they will be far more able to answer questions like those above and they will be better able to argue the relevance and value of their research. This is in line with the view expressed in Prather et al. (2009, p. 390), that "our dissertations have the ability to modify existing theories, but only if we understand how our hypotheses were derived from theories." In addition, in response to the Prather *et al.* position statement, Crowl says (Prather et al., 2009, p. 390): "Putting the data before the theoretical drivers is tantamount to 'putting the cart before the horse' and will surely lead to little or no theoretical or even corrective improvements." In the section below we describe our efforts to make research methodology in CS more explicit and useful.

2 A CONCEPTUAL MODEL

A conceptual model of research methodologies in computing fields of study has been proposed (Pilkington & Pretorius, 2015). Note that the model is not significantly different from other models (for example, Creswell (2014), Oates (2006) and Saunders et al. (2009) and others) but it focuses on the structure of the designs and methods typically used in the computing field (while the other models do not necessarily have that focus). This makes the model appropriate for our study and we expand on it here, focusing on CS research (see Figure 1 for a simplified UML class diagram).

A research endeavour, including those in CS, is underpinned by a philosophical world view and has one or more research designs that guide the process of the research, where each research design uses one or more research methods in carrying out the design and gathering data. Simplified descriptions of the various world views, designs, and methods are provided below only to sketch the basic background; researchers should read up further on their selected approaches when preparing for, and presenting, their research. The intention here is only to provide a structure in which a researcher can locate her or his work rather than a rigid framework that directs it.

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

Philosophical world view

The philosophical world view forms the basic belief about what is reality, truth, and knowledge, and guides the choices that are made in a research endeavour. Although this is often implicit in CS research, it is argued that it is nonetheless present, even if undeclared (Wahyuni, 2012), and that rigour can be improved in such research by acknowledging the philosophical perspective taken by a researcher.

Typical world views prevalent in CS research are the following, although these are certainly not the only world views that are common in computing research.

- Postpositivism: This is the notion that there is a shared, independent reality that exists that we can only partly know to some level of probability (Walliman, 2001). It aims to find deterministic, causal relationships between various variables and their associated phenomena (Creswell, 2014). Deductive logic and testing of hypotheses via experimentation are used to build reliable knowledge. It is largely quantitative and empirical in nature, based on careful observation and measurement (Walliman, 2001). The role of the researcher is to be an objective seeker of truth.

- Pragmatism: For pragmatists, truth is that which can be practically applied, and so ideas and theories are those that have practical value as solutions to problems (Creswell, 2014). For research, it means that all approaches available and useful to understanding a problem are used, often leading to pluralist, mixed method type research. Pragmatism does not hold to any one particular philosophy of truth or reality (Creswell, 2014), allowing the researcher the freedom to choose the best approach that will lead to the intended outcomes.

- Interpretivism: This view holds that the researcher's task is to find and make sense of the meaning that other people give to their experience, inductively developing understanding and theory based on interpreted patterns discerned in such meaning (Creswell, 2014). The aim, thus, is not causal explanation of phenomena, but understanding, which is in keeping with a view of the world as not being mechanistic but organic (Walliman, 2001). In this approach, neither the participant in the research, nor the researcher, are neutral entities independent of the context that is being investigated.

- Constructivism: For constructivists, the meaning that people ascribe to their experiences is socially constructed through interaction with other people (Creswell, 2014). Knowledge is constructed, not discovered, and researchers attempt to understand multiple social constructions of meaning that research participants have. Knowledge, thus, is a construction by the researcher of an understanding of the constructions of the research participants. As much as meaning is negotiated among research participants in their interactions, so also the researcher co-constructs meaning and understanding of the context.

- Transformative: Such a world view takes a more political or critical view of research, and, in working with marginalised communities, rejects the idea of a powerful academic

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

researching the powerless participants. It consciously seeks to explore the historical, contextual, and social justice factors that shape experience, and acknowledges that there are multiple realities based on various backgrounds and values: social, cultural, gender, ethnic, and economic (Creswell, 2014). It advocates for change in the lives of the participants, dealing with issues of oppression, inequality, alienation, and empowerment. The transformative researcher often plays a participative role, actively involving participants in the research process.

It is important that the research design used be consistent with the philosophical world view taken by the researcher. For example, it would not make sense to hold to interpretivist approaches focused on the experiences of participants and use experimental research designs aimed at determining cause and effect.

2.2 Research design

The research design encapsulates the type of study to be undertaken (Mouton, 2001), and is sometimes confusingly termed the research methodology. A research scheme will have at least one such design guiding and structuring the study, although more than one design may

be used where further designs may have equal or subsidiary status and may be run concurrently or sequentially. Two basic types of research designs are recognised: empirical and non-empirical (or theoretical) designs (Mouton, 2001). Empirical research is that research which seeks to gather empirical (that is, observed or experienced) data, which is then interpreted. This division into two design types is not a universal approach, and a qualitative/quantitative approach is sometimes also used, although it has been convincingly argued that the research designs themselves are rarely intrinsically one or the other (Biggam, 2011).

Typical research designs prevalent in CS research are the following, where again these are not the only designs that are common in computing research.

- Empirical research designs
  - Design science research: Design science focuses on the development and performance evaluation of an artefact, thus leading to a better understanding of the research problem for which the artefact was built. There are two main streams in design science, one with the focus on design theory and the other focusing on pragmatic design of the artefact. Merging these two emphases leads to both and adds to the design theory body of knowledge. Artefacts could be conceptual constructs, models, methods and algorithms, instantiations of a system, or better design theory. Core texts on design science research include Hevner et al. (2004) and Gregor and Hevner (2013). It has been argued that design science research is computer science's primary research design (Baskerville, 2008).
  - Experimental: The basic idea behind experiments is to try something and note its effects (Olivier, 1999). Three basic experimental goals are explorative (discover effects), testing (confirming theory in specific cases), and proving a theory correct. Many experiments compare two or more experimental cases to help find an effect

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

of a certain variable (keeping all other variables constant, and changing only the experimental variable). Experiments can be done in controlled, laboratory environments or in naturalistic, field settings.

- Statistical modelling: Statistical modelling is aimed at developing accurate models and representations that approximate phenomena in the real world through the use of statistical techniques (Mouton, 2001). The values projected by the model are then validated against the actual values found in the real world. Such research designs can be used to explore phenomena, as well as find causal links, and may be used in predictive studies. The modelling may be deterministic or stochastic (where some randomness is included).

• Non-empirical (theoretical) research designs

- Algorithms: Research could be designed around finding a new algorithm to solve some problem (which will have to be demonstrably better than any current algorithms), or finding an algorithm for a problem that does not yet have a solution

  (Olivier, 1999). Using an algorithmic approach is a good way to unambiguously and concisely express the detail of how a process can be automated, and of convincing readers that the problem has indeed been solved. Accompanying the algorithm will have to be some analysis of the functionality of the proposed algorithm (Olivier, 1999).
- Model/theory building: Models may be used to simplify the many complex details ofa situation by focusing on the essential parts that are relevant to the problem under consideration. Models can be both useful in proposing new ideas (thus tentatively showing its value or clarifying initial thinking) or to explore and analyse new ideas about existing models and theories (Olivier, 1999). Models can be of different types (either abstracting out the essential properties or reducing the scale of the problem area), and can be presented formally (often mathematically) or informally (often with diagrams to assist understanding) (Olivier, 1999). Note that the model is not the point of the research; modelling is a way of structuring the research, and the model represents an understanding of a particular area of knowledge, which is what the research is aiming at.
- Mathematical proof: This is one of the few means of definitively showing the of a statement (Olivier, 1999). The sorts of approaches used include the use of propositional/predicate logic and automata/formal grammars.

2.3 Research methods

Each research design would use one or more methods in carrying out the research and gathering data, and these could be of three different types. Note that these methods are generally either quantitative or qualitative, although a category to cover more theoretical methods has been included. A design that uses more than one method could be termed a mixed method approach, and is often used to purposely triangulate data to reach a deeper understanding of

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
the field under investigation (Hammond & Wellington, 2013).

Typical research methods prevalent in CS research are listed below. Note that qualitative research methods are not often used in CS research, and are thus not explored here.

- Quantitative research methods
  - Measurement: These are data collection methods used for structuring experimental research designs and other designs that use some form of organised measurement. Typically, data may be collected in single group, experimental/control groups, or blind/double blind approaches.
- Theoretical research methods
  - Argumentation: Such research methods use various forms of argument (deductive, inductive, or both) to present an assertion, contention, or discussion using some form of textual, symbolic, or mathematical notation. Premises are used to develop reasoned conclusions, and challenging (or evaluating) an argument is then based on showing that a premise is not true, or that a conclusion does not follow logically (Olivier, 1999).
  - Prototype: This is a simplified version of a system that may serve several purposes, from being used to show that a concept is implementable (a "proof-of-concept" prototype), a guide to show concretely what a final implementation could do, and an implementation with which to experiment (Olivier, 1999). They are constructed to have the some of the features of a final implemented system, but ignore other characteristics.
  - Simulation: A simulation is a working, running model used to investigate systems and measure results under different scenarios (Olivier, 1999).

2.4 Positioning this work

To give an example of how this conceptual model can be used, this article will be placed within the structure of a research methodology that is presented above. The position that we are taking is based on a pragmatic view of knowledge, in that we believe that there is value in the action of making research's methodology more visible and considered, and that it lead to the better presentation of research.

The research design that we are following could be seen to be philosophical analysis. This design allows us to argue for a particular position in a clear and reasoned manner. The research method being applied is argumentation, where we are using textual arguments to present and discuss our position. Thus, the argumentation method is used within a philosophical analysis design to offer a reasoned discourse leading to conclusions that we trust will be considered, if not convincing.

We can also use the questions posed in section 2.6 to argue that this paper has relevance and makes a contribution in the field.

- *What explicit theoretical dilemma and conceptual underpinning (not just a gap) in CS has*

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

inspired my research?

Various authors (Verschuren & Doorewaard, 2010) have argued that research methodologies are important for doing good research. There are no research articles that argue that good research can be executed without any planning although some ideas for research could be arrived at serendipitously.

In most disciplines research methodologies are made explicit in articles, theses and dissertations. In the computing field, this is clearly the case for IS research but does not always apply for theoretical or applied CS research – see section 2.3 for discussion of this.

When reviewing other typical approaches to research methodology, there is often no consideration of the methods that CS researchers use, which could be why these are not included in CS research papers and theses. The conceptual model presented here specifically includes such designs and methods in an attempt to broaden the wider understanding of research.

- *How would truth (and/or fact) be evaluated in the research?*

  Consideration of research articles, theses and dissertations in computing shows that there is such a difference – more theoretical or applied computing research tends not to have an explicit statement of research methodologies. This could be seen as a reflection of the actions of CS researchers – they do what is needed to solve the problem. That is, they implicitly adopt a pragmatist view. We argue that it would be useful to other researchers if this viewpoint was made explicit.

  A survey of computing professionals may establish why this difference exists but we adopt the approach in this paper that all research should be explicit about research methodologies.

- *Which concepts/models/assumptions from the discipline are integrated in my methodology?*

  In this work, we have made the decision that there is no fundamental dichotomy between IS and CS research. All research is based on appropriate research methodologies whether these are explicitly stated or not.

- *What are the implications of my results for the wider field of CS?* We hope that this paper will lead to better presentation of research done in the field of CS and to more well informed researchers.

- *How do my results challenge or support existing approaches to truth/fact and theory?*

  The paper supports the ideas published by other researchers that research methodologies are important, should be well understood by all researchers and will lead to better research.

In the section below we describe how the approach can be applied to theoretical or applied CS research.

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

3 EXAMPLES OF COMPUTER SCIENCE RESEARCH

There are different types of research that one could undertake in computer science. Ramesh et al. (2004) provides a detailed characterisation of research in the computing field (CS, IS and Software Engineering). The paper suggests some categories of research topics but does not really give details of individual topics at a level that would be useful for this paper. Sanders and Alexander (2015) present a word cloud of CS thesis topics in South Africa over the period of 1978 to 2014 and relate these topics to the 2012 ACM classification scheme (ACM, 2012).

Again, these topics are not at the correct level of detail for us to show the application of the conceptual model in Section 3.

To illustrate the use of the conceptual model we consider the pervasive area of *algorithms* in computing (the term "algorithm" appears 64 times in ACM (2012)). We chose this area as any computing student or researcher will have had significant exposure to the area and the types of problems that have relevance there.

3.1 Algorithms

In Table 1 we present examples of different types of research in the area of algorithms that computer scientists may undertake. These examples range from purely theoretical research, through more experimental and applied topics, to the more socially oriented educational research that some computer scientists may undertake. Similar varieties of research topics can be identified for other fields in CS. For each example in Table 1 we show what research approaches (based on Figure 1) could be applicable to the topic. Note that research topics 6 and 7 are more on the "soft" side of computing (they are much more like doing social sciences research) and so the design and methods used there are not covered in Figure 1 but can be seen in Pilkington and Pretorius (2015).

In addition to the selection of an appropriate approach (methodology), we argue that prior contemplation of one's research will allow one to consider the merits and potential impacts of one's research. If we consider the questions posed in Section 2 and apply these to problem 1 in Table 1 then we can argue the relevance of that research.

- *What explicit theoretical dilemma and conceptual underpinning (not just a gap) inspired my research?*

  In CS there is well established theory (conceptual understanding) of complexity classes (see for example Garey and Johnson (1979)). There is also the well known theoretical dilemma P = NP?

  If the problem can be shown to be in P then that theory becomes applicable.

  If the problem seems to be "hard" then the conceptual underpinning would be that there are a few fundamental problems that have been shown from basic argumentation to be "NP-Complete" and an argument has been made that if a given problem can be reduced to one of these fundamental problems then, the two problems are as "hard as each other".

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

The research is thus inspired by the opportunity to add to knowledge about what types of problems belong in what complexity classes.

- *How would truth (and/or fact) be evaluated in the research?*

  The truth can be evaluated by determining if the researcher has developed a valid argument (a correct proof) based on the existing theories. *Which concepts/models/assumptions from the discipline are integrated in my methodology?*

  The philosophical world view in this research is pragmatism, the design is mathematical proof and the method is argumentation.

  The current assumption/existing theory in the discipline is that P = NP. This research is not about disproving the theory (that would be much harder to do) but is more about adding to the knowledge of P or NP type problems.

  The researcher could start from the basis of trying to understand the problem and then either find a polynomial time algorithm or show that the problem is "as hard" as other known hard problems.

  In the first situation, the researcher could explore various algorithm design approaches to see if these can lead to an algorithm for the new problem.

  In the second case, the researcher's assumption could be that the new problem is NP-Complete. This implies that he/she can essentially use whatever proof techniques that would be considered suitable in order to be able to reduce the new problem to one of the fundamental problems.

- *What are the implications of my results for the wider field of CS? How do they relate to this wider field? In which scope (space, time, and processes) could the results apply?*

  The implications of the results are more knowledge about the classification of problems into complexity classes.

  One outcome could be that there is another problem that is NP-Complete and this could be used for other reductions as it has been shown to be "as hard" as the fundamental problems.

  A different outcome could be that the problem can be solved in polynomial time.

- *How do my results challenge or support existing approaches to truth/fact and theory?*

  The results of the research could support the existing theory that P = NP or could add knowledge about the problems that can be solved in reasonable time.

If the researcher considers these questions before starting their research and then returns to reconsider them after completing the more technical parts of the research then we argue that he/she would be in a much better position to argue that his/her research was a systematic investigation based on existing theories and has made a contribution in the field. In this case showing that he/she understands the theory behind NP-Completeness and what would be required to show a new problem is NP-Complete would help structure the argumentation process, would show that the objective has been achieved and would strengthen the argument

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

that a contribution has been made.

If we now consider problem 5 in Table 1, then it is clear that a researcher could "build a tool", see whether that solves the problem to some extent and then write that up. We would, however, argue that the researcher would do better research if they thought about the motivation for doing the research, the approach that could be taken to do the research, and the implications of doing the research before simply doing it. If we now consider the questions posed in Section 2, then the researcher could position their research as below.

- *What explicit theoretical dilemma and conceptual underpinning (not just a gap) in CS has inspired my research?*

  Research has shown that novices have difficulty in designing algorithms to solve various computing problems and many approaches have been developed to teaching algorithm design (see for example Kerren et al. (2006)). One argument for this is that novices do not develop the same plans as experts and cannot merge plans to solve new problems (Soloway & Ehrlich, 1984).

- *How would truth (and/or fact) be evaluated in the research?*

  In this case, the researcher is taking a pragmatic approach to the research so the tool will be considered to be useful if it can perform as well as an expert in finding the novices' semantic errors.

- *Which concepts/models/assumptions from the discipline are integrated in my methodology?*

  Experts develop plans to solve problems. Novices do not have these plans. We can use experts' plans to find the errors in novices' solutions.

- *What are the implications of my results for the wider field of CS? How do they relate to this wider field? In which scope (space, time, and processes) could the results apply?*

  The first (most obvious) implication is that if the tool works, then it could be used in the teaching of algorithms to novices. A more subtle implication is that the design approach used might be useful for developing tools in other areas of CS.

- *How do my results challenge or support existing approaches to truth/fact and theory?*

  The results could support the idea of experts having plans and being able to adapt these as necessary. It is quite possible that the process of developing and evaluating the tool would also lead to new insights about teaching algorithms that had not been considered before.

The other topics in Table 1 can be addressed in a similar fashion.

We can also apply the approach in more focused research areas. In section 4.2 we show how to apply the approach in *natural language processing* (NLP) (cf. Table 2), which is more contained and is classified under *artificial intelligence*. Here we consider the nature of the field in a little more detail and again propose world views, designs and methods that would be appropriate. Natural language processing

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)

Research in NLP is often classified, as in ACM (2012), in terms of broad problem areas such as information extraction, machine translation, natural language generation, speech recognition, and language phonology/morphology. Since human languages exist independently of being "processed" computationally and did not evolve according to prescribed formal rules, as is the case with programming languages, the notion of a *language model* lies at the core of NLP. Its purpose is essentially to assist a computer in deciding whether a given language input is valid in the human language being modelled, and the coverage and quality of the model is often a determining factor in the quality of the downstream language technology artefact or solution.

Over the past more than 60 years two important approaches to building language models have emerged and are still prevalent in NLP research today: the *symbolic* or *rule-based* approach; and the *data-driven* approach. As we consider the most common characteristics of these approaches, it is clear that they are quite different and may require diverse methodologies.

Rule-based approaches include the use of state machines, formal rule systems and logic formalisms, and linguistic rules (such as grammars), and are often hand-crafted by language experts based on (documented) language structures. Such approaches are flexible (as we have control over the level of linguistic detail required in the grammar rules, and can extend them as needed) and have a high level of precision. This means that they are well-suited to "restricted" or specialised domains and applications where controlled language is used and where so-called producer quality results are essential. However, human language changes all the time and certain rules may be missed or become obsolete, or out-of-lexicon words may be encountered. They require highly-skilled developers, linguists and often also knowledge engineers, and are usually language specific and do not readily generalise to other languages.

Data-driven approaches use statistical and probabilistic methods and machine learning to process large amounts of textual or language data and learn language patterns. These approaches are language agnostic and build their own knowledge, rules and classifiers. As such, they allow for the fast development of a language model for any language, provided that a sufficiently large, good quality data set is available. Data driven approaches may be supervised (where text is annotated by humans) or unsupervised (where a massive corpus is necessary). However, these approaches do not allow debugging, and improving or enlarging the data set does not guarantee that a specific error will not occur again. They also offer no formal guarantees.

Rule-based approaches are often used for under-resourced languages for which limited data and resources are available, but for which linguistic descriptions and/or expert linguists are available. On the other hand, well-resourced languages for which increasingly large corpora and data sets exist are well suited to data-driven approaches. In practice, the best approaches are often hybrid ones, combinations of rule-based and data-driven approaches, which increases the obligation of the researcher to be methodologically well informed. This background knowledge of NLP will be used to consider various research methodological choices, in the context of the conceptual model of section 3, when building language models.

Philosophical world view Post-positivist and pragmatic approaches would be the most appropriate.

- Broadly, rule-based approaches assume an existing "view of the world" – the linguistic description of the language being modelled. So, here a post-positivist philosophical world view is the appropriate choice.
- In the case of data-driven, supervised machine learning, suitably annotated training corpora are needed, which assumes the linguistic knowledge required for an annotation scheme. Here, again, a post-positivist world view is proposed.
- In the case of unsupervised learning (where the training data is not annotated, with no explicitly imposed "linguistic view"), a pragmatic philosophical world view seems appropriate.

Research design Empirical as well as non-empirical research designs may be used in building both rule-based and data-driven language models.

- Design science research, because a language model is a computational artefact, the central notion in design science.
- Experimental, since language models have to be tested and evaluated.
- Statistical modelling, because statistics, probability theory and machine learning algorithms form the basis of data-driven language models.
- Algorithms, since a rule system is per se an algorithm.
- Model/theory building, since a rule-based language model is usually not merely the coding of an already existing set of given rules, but it is often necessary for the researcher (together with the linguist) to carefully and systematically refine and formalise informal linguistic "rules", guidelines and practices that have not been formalised to the necessary levels of detail before.

Research methods Quantitative, theoretical and even qualitative research methods are used in building language models.

- Measurement can be used since all language models have to be evaluated. When a gold standard corpus or data set is available measures such as precision, recall and F-score are often used (Jurafsky & Martin, 2019). For the evaluation of data-driven (including neural) language models perplexity is the most common (intrinsic) evaluation metric (Jurafsky & Martin, 2019).
- Argumentation, since rule-based language models as formal representations allow well-defined operations and reasoning over them. Examples of formalisms are automata theory (for example, finite state computational morphology) (Yona & Wintner, 2008), constructive type theory (GF grammars) (Kolachina, 2015), model theory (Kornai & Kracht, 2015), and many others, and for example, in Peng and Roth (2016) discourse representation theory forms the basis for various machine learning semantic language models.
- Prototyping is a software engineering method that is well suited to, and often used in,

the development of language models as computational artefacts.

- Interviews, the research method under which we classify the collaboration with the expert linguist in rule-based approaches.

In Table 2 we show examples of published research methodology choices for language models in NLP.

3.2 Making coherent choices

We argue that all CS researchers should decide on appropriate methodologies for their research (in a fashion analogous to those in Tables 1 and 2) and should consider the philosophical and methodological questions posed. In particular, PhD and MSc candidates should follow the process as doing so will enable them to make a much clearer exposition of the worth and contribution of their research in their thesis or dissertation.

5 CONCLUSION

As stated earlier, we believe that all research should follow a rigorous and clearly articulated process. This process should clarify the researcher's standpoint, describe why the research has value, describe how the research was done and justify why the particular approach was used. We believe that a discussion of the methodology to be applied in any research should be included in any students' research proposal and in any dissertation or thesis. In addition, research articles should also be explicit about the methodology employed. Being explicit about the research approach used and considering the underlying philosophical questions both at the outset of the research project and at the write up stage will allow researchers to better plan, execute and present their research. It will also make it easier for PhD candidates to assess whether their PhD thesis has met the objective of presenting research that makes a significant, novel contribution in their field.

In this article we have argued that articles and theses/dissertations in the more theoretical or applied areas of computing often do not include discussions of methodology and the related philosophical arguments. In order to address this lack we have shown how a conceptual model of research methodology structure can be applied to different research topics in the field of algorithms – from theoretical topics, to more applied topics, and on to teaching topics inthe area; and in the field of NLP, where we focused on building different kinds of language models. The model that we have used and the philosophical questions we have proposed can be introduced to postgraduate students and developing researchers in order to help them to truly appreciate what research entails and to assist them in producing better publications.

An additional contribution of this work is to add to the debate about the nature of CS research methodologies and how they could or should be taught to students and emerging researchers (Demeyer, 2011; Hassani, 2017; Punyabukkana, 2017; Ramesh et al., 2004).

References

ACM. (2012). The 2012 ACM Computing Classification System. https://www.acm.org/public

ations/class-2012
Ade-Ibijola, A. O. (2016). *Automatic novice program comprehension for semantic bug detection* (PhD Thesis) [Accessed on 31 January 2019]. University of the Witwatersrand, Johan- nesburg. http://wiredspace.wits.ac.za/jspui/handle/10539/21019?mode=full
Association for Information Systems. (2019). IS research, methods, and theories [Accessed on19 January 2019]. https://aisnet.org/page/ISResearch
Baskerville, R. (2008). What design science is not. *European Journal of Information Systems*, *17*(5), 441–443. https://doi.org/10.1057/ejis.2008.45
Biggam, J. (2011). *Succeeding with your master's dissertation: A step-by-step handbook* (2nd). Maidenhead: Open University Press.
Bosch, S. & Pretorius, L. (2011). Towards Zulu corpus clean-up, lexicon development and corpus annotation by means of computational morphological analysis. *South African Journal of African Languages*, *31*(1), 138–158. https://doi.org/10.1080/02572117.20 19.12063275
Bosch, S. & Pretorius, L. (2017). A computational approach to Zulu verb morphology within the context of lexical semantics. *Lexikos*, *27*, 152–182. https://doi.org/10.5788/27-1 -1398
Caplinskas, A. & Vasilecas, O. (2004). Information systems research methodologies and models. *Proceedings of the 5th International Conference on Computer Systems and Technologies - CompSysTech'2004*. https://doi.org/10.1145/1050330.1050413
Cheng, J. & Townshend, B. (2009). A rule-based language model for reading recognition. *ISCA International Workshop on Speech and Language Technology in Education (SLaTE 2009)*, 33–36.
Connolly, R. (2020). Why computing belongs within the social sciences. *Communications of the ACM*, *63*(8), 54–59. https://doi.org/10.1145/3383444
Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches* (4th). Thousand Oaks: SAGE.
Demeyer, S. (2011). Research methods in computer science. *Proceedings of the 27th IEEE International Conference on Software Maintenance*, 600. https://doi.org/10.1109/ICSM.201 1.6080841

Derntl, M. (2014). Basics of research paper writing and publishing. *International Journal of Technology Enhanced Learning*, *6*(2), 105–123. https://doi.org/10.1504/IJTEL.2014.066856

Galliers, R. D. & Stein, M.-K. (2018). Information systems: To be, or not to be, a science? is that the question? *Communications of the Association for Information Systems*, *43*(1), 197–204. https://doi.org/10.17705/1CAIS.04312

Garey, M. R. & Johnson, D. S. (1979). *Computers and intractability. a guide to the theory of NP-Completeness*. San Francisco, CA: W.H. Freeman.

Gregor, S. & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *Management Information Systems Quarterly*, *37*(2), 337–355. https://doi.org/10.25300/MISQ/2013/37.2.01

Håkansson, A. (2013). Portal of Research Methods and Methodologies for Research Projects and Degree Projects. *Proceedings of the International Conference on Frontiers in Education:Computer Science and Computer Engineering FECS'13*, 67–73. http://www.world-academy-of-science.org/worldcomp13/ws

Hammond, M. & Wellington, J. (2013). *Research methods: The key concepts*. Abingdon: Routledge.

Hartmanis, J. (1993). Some observations about the nature of computer science. In R. K. Shyamasundar (Ed.), *Foundations of Software Technology and Theoretical Computer Science* (pp. 1–12). Springer, Berlin. https://doi.org/10.1007/3-540-57529-4_39

Hassani, H. (2017). Research methods in computer science: The challenges and issues. *Computing Research Repository*, *arXiv.1703.04080*. http://arxiv.org/abs/1703.04080

Hevner, A. R., March, S. T., Park, J. & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105. https://doi.org/10.2307/25148625

Holz, H. J., Applin, A., Haberman, B., Joyce, D., Purchase, H. & Reed, C. (2006). Research methods in computing: What are they, and how should we teach them? *Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education*, 96–114. https://doi.org/10.1145/1189215.1189180

Józefowicz, R., Vinyals, O., Schuster, M. & Shazeer, Y., N.and Wu. (2016). Exploring the limitsof language modeling. *Computing Research Repository*, *arXiv.1602.02410*. http://arxiv.org/abs/1602.02410

Jurafsky, D. & Martin, J. H. (2019). *Speech and language processing (3nd edition draft) (Prentice Hall Series in Artificial Intelligence)*. Prentice Hall. https://web.stanford.edu/~jurafsky/slp3/

Kerren, A., Müldner, T. & Shakshuki, E. (2006). Novel algorithm explanation techniques for improving algorithm teaching. *Proceedings of the 2006 ACM Symposium on Software Visu-alization*, 175–176. https://doi.org/10.1145/1148493.1148531

Kolachina, A., P. Ranta. (2015). GF wide-coverage English-Finnish MT system for WMT 2015. *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 141–144. https://doi.org/10.18653/v1/W15-3015

Kornai, A. & Kracht, M. (2015). Lexical semantics and model theory: Together at last? *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, 51–61. https://doi.org/10.3115/v1/W15-2305
Motara, Y. M. (2018). *Preimages for SHA-1* (PhD thesis) [Accessed on 1 February 2019]. Rhodes University, Makhanda. http://hdl.handle.net/10962/57885
Mouton, J. (2001). *How to succeed in your master's and doctoral studies: A South African guide and resource book*. Pretoria: Van Schaik.
Myers, M. D. (1997). Qualitative research in information systems. *Management Information Systems Quarterly*, *21*(2), 241–242. https://doi.org/10.2307/249422
Myers, M. D. (2018). Qualitative research in information systems [Accessed on 21 January 2019]. *Association for Information Systems (AISWorld)*. http://www.misq.org/supplements/
Nemeskey, D. M. (2017). EmLam – a Hungarian language modeling baseline. *Proceedings of the 13th Conference on Hungarian Computational Linguistics (MSZNY)*, 91–102.
Nunamaker Jr, J. F., Briggs, R. O., Derrick, D. C. & Schwabe, G. (2015). The last research mile: Achieving both rigor and relevance in information systems research. *Journal of Management Information Systems*, *32*(3), 10–47. https://doi.org/10.1080/07421222.2015.1094961
Oates, B. J. (2006). *Researching information systems and computing*. London: SAGE.
O'Leary, Z. (2017). *The essential guide to doing your research project* (3rd). London: SAGE.
Olivier, M. S. (1999). *Information systems research: A practical guide*. Johannesburg: Published by author.
Peng, H. & Roth, D. (2016). Two discourse driven language models for semantics. *Proceedingsof the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 290–300. https://www.aclweb.org/anthology/P16-1028
Petter, S., Carter, M., Randolph, A. & Lee, A. (2018). Desperately seeking the information in information systems research. *SIGMIS Database*, *49*(3), 10–18. https://doi.org/10.1145/3242734.3242736
Pilkington, C. & Pretorius, L. (2015). A conceptual model of the research methodology domain. *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 2:KEOD*, 96–107.
Prather, C. M., Choate, D. M., Michel, M. J. & Crowl, T. A. (2009). Putting the "Ph" back into "PhD": Framing graduate research in a theoretical context. *Frontiers in Ecology and the Environment*, *7*(7), 389–390. https://doi.org/10.1890/1540-9295-7.7.389
Punyabukkana, P. (2017). Teaching research methods for computer science students using active learning approach. *Proceedings of the 2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 256–260. https://doi.org/10.1109/TALE.2017.8252343
Ramesh, V., Glass, R. L. & Vessey, I. (2004). Research in computer science: An empirical study. *Journal of Systems and Software*, *70*(1-2). https://doi.org/10.1016/S0164-1212(03)0 0015-3

Sanders, I. D. (2002). *The axial line placement problem* (PhD thesis) [Accessed on Viewed 2018/08/21]. University of Pretoria, Pretoria. http://hdl.handle.net/2263/28673
Sanders, I. D. & Alexander, P. M. (2015). A study of Computing doctorates in South Africa from 1978 to 2014. *South African Computer Journal*, (57). https://doi.org/10.18489/sacj.v0i57.294
Saunders, M., Lewis, P. & Thornhill, A. (2009). *Research methods for business students*. Pearson education.
Shackelford, R., Cross II, J. H., Davies, G., Impagliazzo, J., Kamali, R., vLeBlanc, R., Lunt, B., McGettrick, A., Sloan, R. & Topi, H. (2005). *Computing Curricula 2005: The Overview Report* (Overview Report). The Association for Computing Machinery (ACM), The Asso- ciation for Information Systems (AIS), and The Computer Society (IEEE-CS).
Soloway, E. & Ehrlich, K. (1984). Empirical studies of programming knowledge. *IEEE Transactions on software engineering*, (5), 595–609. https://doi.org/10.1109/TSE.1984.5010283
Thorne, S. (2016). PhD without the Ph? *Nursing Inquiry*, *23*(4), 281–282. https://doi.org/10.1111/nin.12169
University of KwaZulu-Natal. (2020). Information for the guidance of examiners of higher degrees (Form EX 1-8).
University of South Africa. (2014). Procedures for Master's and Doctoral Degrees.
University of the Western Cape. (2019). Information and guidelines for examiners of Master's and Doctoral theses.
Verschuren, P. & Doorewaard, H. (2010). *Designing a research project* (2nd). The Hague: Eleven International Publishing.
Wahyuni, D. (2012). The research design maze: Understanding paradigms, cases, methods and methodologies. *Journal of Applied Management Accounting Research*, *10*(1), 69–80.
Walliman, N. (2001). *Your research project: A step-by-step guide for the first-time researcher*. London: SAGE.
Walsham, G. (2018). Research methodologies for information systems in the development con- text: A tutorial. *The Digital Challenge: Information Technology in the Development Context: Information Technology in the Development Context*.
Whitman, M. E. & Woszczynski, A. B. (Eds.). (2004). *The Handbook of Information Systems Research*. Hershey, PA: Idea Group Inc (IGI).
Willcocks, L. P., Sauer, C. & Lacity, M. C. (Eds.). (2015). *Formulating Research Methods in Information Systems: Volume 2*. Basingstoke: Palgrave Macmillan.
Willcocks, L. P., Sauer, C. & Lacity, M. C. (Eds.). (2016). *Enacting Research Methods in Information Systems: Volume 1*. Basingstoke: Palgrave Macmillan.
Yona, S. & Wintner, S. (2008). A finite-state morphological grammar of Hebrew. *Natural Language Engineering*, *14*(2), 173–190. https://doi.org/10.1017/S1351324906004384