Concept paper

Viktor Gsteiger
Matriculation Number: 18-054-700

26.09.2020
Seminar: 58826-01 - Turing Award Winners and Their Contributions

Contents

1 Audience

The main audience for my seminar paper will be the other students as well as the examiners of the Seminar Turing Award Winners and Their Contributions. In the following part I will be analyzing my audience in respect to my topic and make suggestions in what way I may have to structure my report to reach both groups the best way.

1.1 Students

The students are all more or less proficient in at least one high level programming language of our time. They have also seen a lot of pseudo code in several lectures and know the basic flow of a program. Most however, will not know where the standard notation of so many of today's programming languages come from. I would have to pay due diligence to explain the importance of a meta language to describe a programming language. I would also have to reintroduce some to the definition of a context-free grammar and its ramifications. Furthermore I would have to lay out a red ribbon throughout the report to make clear why this contribution of Peter Naur was so important and what its ramifications are for the languages and the field of computer science.

1.2 Examiners

The Examiners should know the importance of a standardized meta language and should know the definition of a context-free grammar. However, they may not know much about Peter Naurs philosophies and his rather less formal approach to the field of computer science and programming. They may be interested in his contributions to the teaching of computer science and his contribution to the design of modern languages. I hope to entertain them with a good structure and a completed message that will stick with them.

2 Peter Naurs career

Peter Naur, born in 1928 to a family of artists and business minded parents, had the early interest involving astronomy. He was allowed to work at the local observatory. Peter Naur published his first scientific paper with 15 and he had attained many technical skills of mathematics early on from professional astronomers which took him on as a young prodigy. After finishing his astronomy degree in Copenhagen he was recommended to conduct research at King's college, Cambridge where he focused of astronomy and the emerging field of computer science. Due to weather constraints Naur had to divert his time from astronomy and had more time to spend programming the Electronic Delay Storage Automatic Calculator (EDSAC). Peter Naur, used to do complicated computational calculations by hand focused his energy mostly on the limitations of the EDSAC such as the limited number range. After leaving Combridge, Naur conducted research at Harvard University and Princeton, where he learned the state of the art in computing. The now more established field of computer science would be the focus of Naur, who returned to Denmark in 1953.

He joined the computer center of Copenhagen and was asked to participate in the development of an algorithmic programming language, later called ALGOL. In the beginning he mostly focused on the highly influental Zurich report of 1958 which lay the ground work for ALGOL. Naur, a very pragmatic computer scientist, was not content with the Zurich report and quickly organized a conference in Copenhagen to discuss the difficulties Naur saw in the Zurich report. After the conference, which did not satisfy Naur entirely, he initialed the ALGOL Bulleting. By editing this discussion journal, Naur became unintentionally the leading European behind the effort to create ALGOL.

Naurs contribution to ALGOL lay in selecting the right forms of description to define the language. This was in line with his later research as he was more interested in the meta aspects of the language rather than specific implementation. One important piece of this description is the Backus-Naur form named by Knuth (1964) (originally called the Backus normal form). The Backus-Naur form is a metalanguage of metalinguistic formulas for chomsky context-free grammars. Naur applied the notation proposed by Backus (1959) and modified it slightly and used it for the important ALGOL 60 report by Backus et al. (1963).

Naur also contributed to the establishment of computer science as an academic field in Denmark and he has continued to advocate for an applied form of computer science not only for his students but also for the field in general hereby oposing Dijkstra and Wirth structured programming agenda.

3 Scientific contribution

Peter Naur received the Turing Award in 2005 for his fundamental contributions to programming language design and his substantial contribution to the definition of Algol 60, to compiler design, and the art and practice of computer programming.

The main scientific contribution I would focus on in this seminar report would be on the fundamental contributions to programming languages, especially Naurs contribution to the Backus-Naur form and it's influence on the later design of programming languages. I would also consider the substantial contribution of Naur (1963) to the field of compiler design in perspective of the language ALGOL 60. The report would give an overview over the development of a formal notation techniques used to describe the syntax of languages. I would consider the contributions of Chomsky (1956) to the definition of the Chomsky hierarchy and it's influence on the field of formal languages. Furthermore, I would consider the important contribution of Backus (1959) to the Backus normal form.

From this background on I would continue to establish the situation in the 1960s where Naurs involvement in the Algol development process and his own philosophy on languages and the art of computer programming influenced his contribution to the Backus normal form to establish it as the metalanguage to talk about Algol. I would also consider secondary articles on the Backus-Naur form by for example McCracken and Reilly (2003) or Rohl (1968) to establish the reception of the Backus-Naur form and it's influence over the field of language and compiler design.

Together with this background and the development of this formal language I would then consider how the Backus-Naur form has influenced the work of Naur (1963) work on compiler design and more importantly on how it influenced his philosophy regarding the integration of programmers into the newly developed language. Naur (1961) was a strong opponent of too strict rules for programmers and I would hope to find a red line going through his work as a scientist, a pioneer, and a teacher to see his broad interest in philosophy and psychology play out in a general way.

I believe that the considerations of meta aspects like the notation and Naurs contribution to the development of the Algol language has helped it become one of the most important programming language, however, not becuase it was so widely used but rather because it focused on the meta aspects and could be so easily extended and adapted to new environments and needs of the coming times. It would also be important to discuss the time setting and how Algol compared to other languages. But maybe that would be too much for the scope of this report.

4 Programming project

I have several ideas on a programming project:

4.1 Backus-Naur form parser

A parser written in python which could evaluate statements written in the Backus-Naur form and do certain actions on this context=free grammar.

4.2 Algol 60 simulator

A simpulator that can read and evaluate Algol files, do some mild static code checking and print out the result of a Algol 60 program.

5 Conclusion

The conclusion of this concept report is that I will have to discuss some things with my examiner. Mostly I would like to know the scope, if I am on a good path, if it involves enough technicalities and if I covered an interesting topic in their opinion. I would also like to discuss the project in more detail and maybe get some feedback on my ideas.

References

Donald E. Knuth. *Backus Normal Form vs. Backus Naur Form*, chapter 7.12. ACM, 1964.

J. Backus. The syntax and semantics of the proposed international algebraic language of the zurich acm-gamm conference. In *IFIP Congress*, 1959.

J. W. Backus, F. L. Bauer, J. Green, C. Katz, J. McCarthy, P. Naur, A. J. Perlis, H. Rutishauser, K. Samelson, B. Vauquois, J. H. Wegstein, A. van Wijngaarden, and

M. Woodger. Revised report on the algorithmic language ALGOL 60. *The Computer Journal*, 5(4):349–367, 01 1963. ISSN 0010-4620. doi: 10.1093/comjnl/5.4.349. URL `https://doi.org/10.1093/comjnl/5.4.349`.

P. Naur. The design of the gier algol compiler. *Annual Review of Automatic Programming*, 4:49–85, 1963.

N. Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956.

Daniel D. McCracken and Edwin D. Reilly. *Backus-Naur Form (BNF)*, page 129–131. John Wiley and Sons Ltd., GBR, 2003. ISBN 0470864125.

Jeffrey S Rohl. A note on backus naur form. *The Computer Journal*, 10(4):336–337, 1968.

Peter Naur. A course of algol 60 programming. *ALGOL Bull.*, 2(Sup 9):1–38, January 1961. ISSN 0084-6198.