International Journal of Computer Trends and Technology
ISSN: 2231 – 2803 / https://doi.org/10.14445/22312803/IJCTT-V70I4P102

Volume 70 Issue 4, 13-18, April 2022
© 2022 Seventh Sense Research Group®

Original Article

Sanskrit Programming Language

Hriday S. Kharpude[1]

[1]*Smt. Sulochanadevi Singhania School(IBDP), Maharashtra, India.*

Received: 05 March 2022
Revised: 24 April 2022
Accepted: 26 April 2022

Abstract *- Language means sophisticated communication developed for interaction between humans with a certain commonality in their backgrounds. Today we merely learn a language and carry on to communicate with others familiar with the same language for the rest of our life. We forget to appreciate and give thought to creating such a complex uniformity amongst the then 'unintelligent' beings. The oldest language known to have been conversed in by humans on Earth, Sanskrit, is the Mother of all languages. While even theorizing about its creation is strenuous, its origin is fortunately not a mystery. The entire process was documented by the students of Mahariṣi Pāṇini, who is considered the creator of Sanskrit. Primarily, Pāṇini created a treatise on Sanskrit grammar named Aṣṭādhyāyī. This served as an outlay for building the main structure of Sanskrit. Correlating with today's programming world, Pāṇini primarily made an outlay of programming language syntax and then put together the pieces to form a compact, well-defined programming language. This analogy potentially highlights the main aim and objective of this paper: to explore the existence of Sanskrit as more than a language, as a Programming Language.*

Keywords *- Sanskrit, Programming languages, Computer science, The origin of Sanskrit, Natural languages.*

I. INTRODUCTION

Human-derived languages happen to be the unique form of communication ever developed in living beings. It is nearly impossible for any other species, including Neanderthals, to have had a 'language .'That is so because language evolution is also connected with biological evolution, which has led to an argument wherein the recent human evolution is said to be more influenced by language rather than genes.[1]

The first evolution and development of 'language' are unpredictable. However, its ancestry can be successfully related to the Homo Sapiens. This conclusion is derived from the fact that as near as 40,000 years ago, the Neanderthals did not show any signs of 'symbolic thinking.' At the same time, the Homo Sapiens of the time had plentiful art, musical instruments, and also sewing instruments![2]

Unfortunately, it is so that machines fail to understand human languages as they solely operate on '1's and '0's or what is referred to as 'Machine Language' - ML. The first officially introduced programming language was Plankalkül, an abbreviation of Plan Calculus, in 1944-45. However, unofficially, the first algorithm was developed by Ada Lovelace for the first computer invented by Charles Babbage in the 1800s. [3][4] And that was the start of a series of mediums connecting the machine world to the real world: Assembly Language, Shortcode, Autocode, FORTRAN, etc.[3]

But given our customary knowledge, it begs the question: why aren't human languages, or rather, natural languages, plainly used for programming?

A major reason is that most natural languages have very limited means of Abstraction. They have somewhat of a knowledge gap wherein they only provide small, fixed sets of pronouns to use as means of Abstraction, and the rules for binding pronouns to meanings are often unclear.[5] However, programming often becomes very intense and requires the usage of pointers such as simpler tags or names for referring to more complex things. On the other hand, English provides considerable means for Abstraction and is much preferred for Computer Science and Development purposes.

Another notable factor is that most languages have considerable ambiguity in their words[6]. They have alterable meanings that stand to be used for puns and regular sentences in our daily lives. However, that is a serious problem for a machine that only understands black and white and no such thing as gray.

Nonetheless, the English language was easily adopted into the programming system and spread quite widely without any objections to its usability. One of the reasons for this easy, undisputed acceptance of English into Computer Programming is Abstraction.

In programming, Abstraction is defined as the process of reducing the object to its essence so that only the necessary characteristics are exposed to the users[7].

For instance, in English used daily, we have to type:

"My name is xyz,"

but while programming in English, we can merely mention:

“name = ‘xyz’ ”

This convenience of effort is called Abstraction. And as is evident from the above example, means of Abstraction in English are considerably obvious. Comparatively, other natural languages have lesser means of Abstraction and several cases of homophones and synonyms, which are quite the trouble for a processor to identify.

Other than the fact that English is quite useful in terms of Abstraction, it was also the dominant language across the world, hence becoming a default choice for the creation of Coding Interfaces. In other words, English at the time and today can be rightfully referred to as the world's 'lingua franca'[8].

According to the Oxford Dictionary, a language is referred to as 'Lingua Franca' when adopted as a common language between speakers with different native languages and cultural backgrounds. And English is referred to as GLOBAL lingua franca because it is the common language or mode of communication that enables people to understand one another regardless of their cultural and ethnic backgrounds, making communication and understanding one another considerably efficient and easy across countries.[8]

A survey conducted by y studios asked developers about why they think English prevails in coding. One of the answers was:

“Having English keywords keeps the programming abstracted from normal language for us…We do NOT want local language keywords for programming. Anyone who has made their programming language at university and tried local keywords will agree with me on how confusing it is to read the code."[9]

Another response was:

“*English has the advantage that it's spoken by many hundreds of millions of people. A language using non-English key words probably wouldn't have the demand needed to support it. The only exception seems to be Chinese. It's spoken by even more people than English is, and the huge difference between English and Chinese would make learning more of a barrier for Chinese learning to program.*”[9]

Continuing on the second response, the Chinese do indeed have programming languages, like 中文培基(Zhōngwén péi jī) and Wenyan. These languages operate with the Chinese language as its core for the entire interface. This was done to enable Chinese-speaking people who haven't learned English to write code and not be lost in today's technological era.[10]

Similarly, several other programming languages find their base in the natural language of that region. These programming languages attempt to make programming easier and accessible to different people worldwide. Sanskrit happens to be one such natural language with great potential to serve as the basis of an efficient programming language.

Onwards, a careful analysis of the Sanskrit Language brings out the varying similarities it shares with present-day programming languages[11]:

- *Pratyāhāras*[12], the key pillars on which the Sanskrit *sūtras*[13] are built, are abbreviations or acronyms of certain groups of letters that need to be referred to as a rule. The primitive data types of present-day programming languages can be compared to the *pratyāhāras*.

- *Saṃjñās*[14] are equivalent to programming languages' reserved words/keywords. Keywords are predefined, reserved words used in programming that have special meanings to the compiler[15]. Similarly, the *saṃjñās* serve an important role in defining the final Sanskrit Language rules.

- *Pratyayas*[16] are analogous to several standard operators in present-day programming. They are intensely used in the formation of words, and all Sanskrit vocabulary is based upon the various forms of *pratyay*.

- *Vidhis*[17] can correspond to the various built-in functions in programming languages. Each such *vidhi* is an algorithm in itself and, just like the built-in functions mentioned earlier, generates a specific output for a certain input. Such algorithmic processing forms a major part of the *Aṣṭādhyāyī*[18] and is seen at every step of it.

- *Anuvṛttis*[19] can be compared to the built-in libraries of programming. Libraries mainly serve to provide functionalities that we can readily use without designing them while coding. Along similar lines, *anuvṛttis* enables Sanskrit users to define a word just once in one *sutra*, and instead of repeating the definition, all following *sutras*, by default, use that definition.

II. METHODOLOGY

In methodology, we get to know about the precise procedure that has been undergone while creating the Sanskrit Programming Language.

The procedure for creating a programming language involves 2 phases[20]:

- Designing the outlay and syntax of your language
- Building a compiler for this outlined language

However, in this programming language, an alternate method has been used. Rather than building a programming language from the ground up, the purpose of Sanskrit-based interaction between a programmer and the machine can be achieved by simply adding an external layer of Sanskrit interaction to a 'core language,' such as Python. On the downside, this language must be considered efficient in the time taken to compile and execute programs.

This programming language will have Python as its 'core' in layman's terms. However, its interface will be altered to accept Sanskrit commands only. Primarily, the short-term goal is to serve the purpose of performing basic arithmetic. It should identify regular mathematical operations in Sanskrit and perform accordingly on numeric values entered by the user.

Now for the creation, primordially basic addition, subtraction, multiplication, and division functions must be defined. But for each function, we must have triggers/keywords to activate the individual functions. After researching mathematical operators in Sanskrit, the below-mentioned ‘operator name’ list has been inducted for triggering the defined functions in the source code[21].

Table 1 provides a list of the keywords defined in the source code that the user can use for calling respective mathematical operations on any given pair of numbers.

Table 1. Mathematical Operations in Sanskrit and their corresponding keywords defined in the source code

Operation	Trigger Keyword	English Writing
Addition	युतम् । योजनम् । संयोजनम् । संकलः ।	Yutam, Yojanam, Samyojanam, Samayojanam, Sankalah
Subtraction	व्यवकलनम् । व्यावकलनम् । वियोजनम् ।	Vyavakalanam, Vyaavakalanam, Viyojanam
Multiplication	गुणनम् ।	Gunanam
Division	विभाजनम् । भाजनम् ।	Vibhajanam, Vibhaajanam, Bhajanam, Bhaajanam

While using the above keywords in programming, the user must follow a syntax that has been defined for the program. The syntax is as follows:

<operator name> <numeric value> <numeric value>

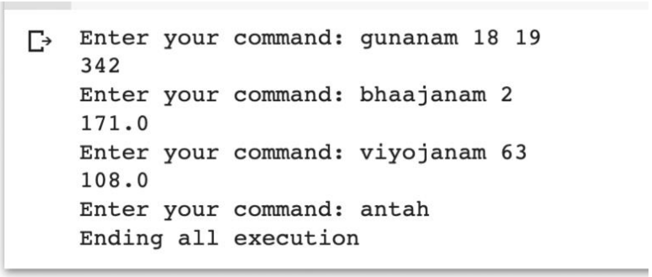
Additionally, the user can operate involving the previously calculated value and another number input by the user. This is done by simply using the keyword and the new number separated by a space. Such syntax is recognized by the system and performs the respective operation on the previously calculated and the new number. The syntax:

<operator name> <numeric value>

Lastly, the functioning and simultaneous execution of the programming language is as follows:-

Firstly, the user runs the program, followed by the formation of an input line. Here the user can input mathematical operations using the Sanskrit language following the defined syntax explained earlier.

Once a command in the earlier mentioned formats is entered, the user can press 'enter,' the line is executed, and the output is displayed below. Another input line is automatically created below the previous output. Here, the user can input another operation and continue the loop or enter a keyword to end all execution. This keyword for that is 'antah' or 'antam'[22], depending on the user's choice. The above keyword triggers a break statement which pulls the console out of the 'while' loop and 'if' conditions and brings the execution of the entire source code to an end.



```
Enter your command: gunanam 18 19
342
Enter your command: bhaajanam 2
171.0
Enter your command: viyojanam 63
108.0
Enter your command: antah
Ending all execution
```

Fig. 1 Figure one demonstrates the code in correspondence with the earlier paragraph

Table 2 provides a list of ALL defined keywords that can be used by the user while operating the interface.

Table 2. List all defined keywords available for use by the user

Keywords	Operation
Yutam, Yojanam, Samyojanam, Samayojanam, Sankalah	Addition
Vyavakalanam, Viyojanam, Vyaavakalanam	Subtraction
Gunanam	Multiplication
Vibhajanam, Vibhaajanam, Bhajanam, Bhaajanam	Division
Antah, Antam	Terminate Execution

III. AIM OF THE STUDY

The purpose behind this entire discussion and the creation of the programming language was to serve as cause for this paper's main idea, i.e., to present the possibility of adopting Sanskrit as a Programming Language.

IV. RESEARCH DESIGN

In this research paper, the research methodology is exploratory as we explore Sanskrit intending to establish and further the presence and advantages of using Sanskrit as a Programming Language.

V. TOOLS USED

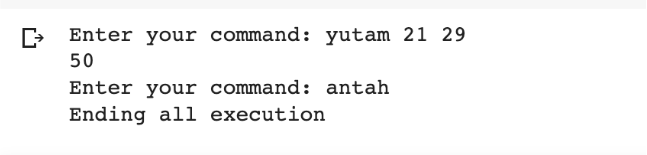
To develop the basic Sanskrit Programming Language, the use of Google Collaboratory[23] has been done. Google Colab is an online platform offered by Google Research that allows users to execute Python code online and is especially well suited for machine learning and data analysis.

VI. RESULTS

This research paper shows why Sanskrit as a programming language to communicate with machines while being a natural language for communication amongst humans is advantageous.

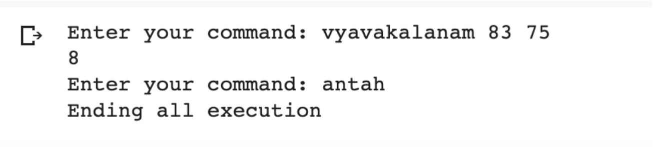
As was explained earlier, this project of creating a new programming language does not include creating a synthetic programming language from the very basics, such as interpreters and compilers, since that would require a considerable amount more knowledge and effort than can be handled by a single person. Hence, in an attempt to skip the above process, direct use or rather 're-use' of the python programming language has been done. The compilers and interpreters of Python are being built upon by adding a 'Sanskrit Language Only' interactive interface. This interface is similar to the code editor window for any IDE(Integrated Development Environment).

Presently, the interface works on the lines of a programming language interface with the functionality of basic mathematical operations in Sanskrit syntax. This syntax has been defined as fit for Sanskrit mathematical operators using keywords such as Sanskrit for the operations of addition, subtraction, multiplication, and division. These command terms were then used to create conditional statements for triggering specific mathematical operations as and when called by the user. Presently this works on a line-to-line basis. It reads a single line entered by the user, executes it, and provides a new line of input for the user.



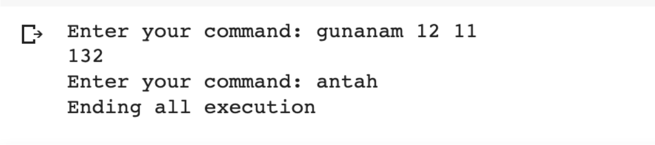
```
Enter your command: yutam 21 29
50
Enter your command: antah
Ending all execution
```

Fig. 2 Demonstrates the execution of the addition command as entered in Sanskrit on two random integers entered by the user



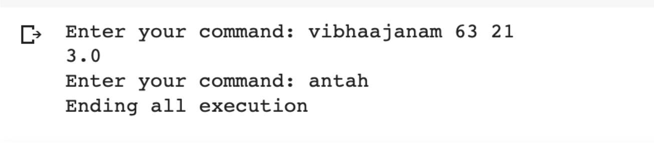
```
Enter your command: vyavakalanam 83 75
8
Enter your command: antah
Ending all execution
```

Fig. 3 Demonstrates the execution of the subtraction command as entered in Sanskrit on two random integers entered by the user



```
Enter your command: gunanam 12 11
132
Enter your command: antah
Ending all execution
```

Fig. 4 Demonstrates the execution of the multiplication command as entered in Sanskrit on two random integers entered by the user



```
Enter your command: vibhaajanam 63 21
3.0
Enter your command: antah
Ending all execution
```

Fig. 5 Demonstrates the execution of the division command as entered in Sanskrit on two random integers entered by the user

VII. DISCUSSION

As was elaborated earlier, this project of creating a basic Sanskrit Programming Language is built with Python at its core. That is, a completely new compiler and interpreter required for a new programming language has not been developed. Rather, an interactive interface has been developed on a python IDE - Google Colab. Presently, this language performs the basic function of calculating mathematical operations. However, the user has to access the operations of this calculator by using commands with Sanskrit syntax. This syntax involves each mathematical operation as it is called in Sanskrit.

Primarily, every input given by the user is split into a number of terms as they are divided by a space. This is done using the '.split()' function of strings. The returned array is later used in 'if' conditions to perform appropriate operations following the user's choice.

To develop the Sanskrit syntax, research had to be conducted to affirm the Sanskrit words corresponding to the arithmetical operations. These keywords were then introduced in the source code for 'if-elif' statements in the Python source code. During execution, the keyword input by the user is stored in the array returned by the '.split()' function and runs through the condition statements. Depending on the mathematical function triggered by the keyword, the corresponding value is output for the given numbers input by the user.

Another feature that has been introduced is the continuation of the previously calculated value. The user can operate involving the most recently calculated value and another value input by the user. This is enabled by simply mentioning the keyword for the respective operation and the new number. The logic behind this feature is checking the array length of the input command. For a normal command which involves a keyword and two numbers, the array size is three, but a value continuation command has an array size of two. Thus, the array size enables the system to recognize the syntax input by the user, and the respective function is executed on the previous answer and the new number input by the user. The final value is then displayed.

VIII. CONCLUSION

Through this paper, we understand the importance of language in programming, a rather not-so-trivial connection. An evident reason why English was, without argument, accepted as a means of communication in computer science is established. In this discussion, we see the advantages of English over other natural languages in programming and the advantages of Sanskrit in programming.

We see the various features of Sanskrit, how without even induction into computer science, they still correlate to the terminologies of today's programming world. In summary, the most important advantages of Sanskrit for programming are:

- Sentence construction is not applicable: Primarily, all of today's natural language processors and systems often have difficulty identifying/forming the right sentences. This problem is completely eradicated in the case of Sanskrit Linguistics because it does not have sentence construction and organization as a part of its grammar[24]. Correct words organized in any random order have the same meaning.
- Abstraction: Sanskrit Language in itself has strong means of Abstraction. [Abstraction is defined as the process of reducing the object to its essence so that only the necessary characteristics are exposed to the users]. This is achieved because of *Sandhis*[25] (Sanskrit for 'joining'), which is the combination of two words that fit certain rules and conditions. Such combining is the very essence of Abstraction and evidence that Sanskrit would be a very good fit as means for programming.

IX. LIMITATIONS

Reflecting on this project and research paper, there are quite a few aspects for further development and improvement:

- Furthering the programming language project by adding more features such as finding remainder, factorial, trigonometric functions, etc. aa
- Alter the programming language to work on a 'verse' basis, and not command prompts as is the true form of Sanskrit.
- Replacing the Python compilers and interpreters with newly developed systems, especially meant for Sanskrit Programming.
- Defining a universal code for Sanskrit Characters similar to ASCII code in English
- Further exploring features of Sanskrit which could be useful for computer programming.

X. ACKNOWLEDGEMENT

This paper and the research behind it would not have been possible without the exceptional support of my supervisor, Mr. Mritunjay Sharma. His enthusiasm, knowledge, and exacting attention to detail have been an inspiration and kept my work on track from my first idea of this concept to the final draft of this paper.

I also extend my gratitude to my counselors, Ms. Sweksha Vaish, Ms. Dikshita Goel, and Mr. Arnav Joshi, for their unwavering support and belief in me. Without their help, completing this paper while maintaining a constant writing standard would have been considerably difficult.

Last but not least, I would also like to thank my dear parents, Mr. Sharad Kharpude and Mrs. Rutuja Kharpude, for their guidance and support, which helped me shape my thoughts a lot.

REFERENCES

[1] Pagel, M. Q&A: What is human language, when did it evolve, and why should we care?. BMC Biol, 15(64) (2017). https://doi.org/10.1186/s12915-017-0405-3

[2] Pagel M. Wired for culture: origins of the human social mind. WW Norton & Company, (2012).

[3] History of programming languages, DevSkiller - August 26, (2021). https://devskiller.com/history-of-programming-languages

[4] William Harris, April 12, (2021).Who Invented the Computer? https://science.howstuffworks.com/innovation/inventions/who-invented-the-computer.htm

[5] Chapter 3: Programming, The University of Virginia https://www.cs.virginia.edu/~evans/cs150/book/ch3-programming.pdf

[6] Steven T. Piantadosi, Harry Tily, Edward Gibson: The communicative function of ambiguity in language, Cognition, 122(3) (2012) 280-291.ISSN 0010-0277. https://doi.org/10.1016/j.cognition.2011.10.004. (https://www.sciencedirect.com/science/article/pii/S0010027711002496)

[7] Java Abstraction, W3 Schools https://www.w3schools.com/java/java_abstract.asp

[8] Why the English Language Became the World's Lingua Franca, English Explorer - September 12, (2021). https://www.englishexplorer.com.sg/why-the-english-language-became-the-worlds-lingua-franca/

[9] The Language of Codes: Why English is the Lingua Franca of Programming, Y Studios, August 16, (2018). https://ystudios.com/insights-passion/codelanguage

[10] Stian Haklev, 21 November , Chinese Python: Translating A Programming Language, Random Stuff That Matters. (2008).http://reganmian.net/blog/2008/11/21/chinese-python-translating-a-programming-language/

[11] The wonder that is the Aṣṭādhyāyī (the ancient Sanskrit grammar of Pāṇini) - Golden Reed, Hiraṇyaya Vetasa https://goldenreed.wordpress.com/2021/09/25/the-wonder-that-is-the-a%e1%b9%a3%e1%b9%adadhyayi-the-ancient-sanskrit-grammar-of-pa%e1%b9%87ini/

[12] Pratyāhāra प्रत्याहार, Sanskrit Dictionary https://sanskritdictionary.com/praty%C4%81h%C4%81ra/2616/40

[13] Sūtra सूत्र, Sanskrit Dictionary https://sanskritdictionary.com/s%C5%ABtra/269853/1

[14] Saṃjñā संज्ञा, Sanskrit Dictionary https://sanskritdictionary.com/sa%E1%B9%83j%C3%B1%C4%81/3956/40

[15] Keywords and Identifiers, www.programiz.com https://www.programiz.com/c-programming/c-keywords-identifier

[16] Pratyaya प्रत्यय, Sanskrit Dictionary https://www.sanskritdictionary.com/pratyaya/2596/40

[17] Vidhi विधि, Sanskrit Dictionary https://www.sanskritdictionary.com/vidhi/212219/1

[18] Ashtadhyayi - work by Panini, The Editors of Encyclopaedia Britannica https://www.britannica.com/topic/Ashtadhyayi

[19] Anuvṛtti अनुवृत्ति, Sanskrit Dictionary https://www.sanskritdictionary.com/anuv%E1%B9%9Btti/8685/1

[20] How would I go about creating a programming language? Federico Tomassetti https://tomassetti.me/how-to-create-programming-language/

[21] गणितन्यासाय शब्दाः (Words for Arithmetic Operations), Himanshu Pota, (2008). https://learnsanskrit.wordpress.com/2008/05/03/गणितन्यासाय-शब्दाः-words-for-arithmetic-operations/

[22] 'End' Meaning in Sanskrit, Bsarkari.com https://www.bsarkari.com/end/meaning-sanskrit

[23] Google Colaboratory, Google Research https://research.google.com/colaboratory/

[24] Krishnamurthi CG, Ph.D. Sanskrit, University of Hyderabad, January 8, Is Sanskrit Independent of the order in which words appear in a sentence?, (2017). https://qr.ae/pGq2Eq

[25] Sandhi, learnsanskrit.org https://www.learnsanskrit.org/guide/core/sandhi/