

LOGIC PROGRAMMING WITH PROLOG

*Jamal Othman¹, Mahfudzah Othman², Nora Yanti Che Jan³, Syarifah Adilah Mohamed Yusoff⁴
*jamalothman@uitm.edu.my¹, fudzah@uitm.edu.my², noray084@uitm.edu.my³,
syarifah.adilah@uitm.edu.my⁴

^{1,4}Jabatan Sains Komputer & Matematik,
Universiti Teknologi MARA Cawangan Pulau Pinang, Malaysia

^{2,3}Kolej Pengajian Pengkomputeran, Informatik & Media
Universiti Teknologi MARA Cawangan Perlis, Malaysia

*Corresponding author

ABSTRACT

Programming paradigms can be classified such as the imperative, object-oriented, functional, logic and scripting paradigms. The curriculum structures of computer programming subject at any higher level of education institutions focus on the imperative and object-oriented programming languages. The logic programming paradigms will be offered as elective subject if the students interested to learn on the Artificial Intelligence (AI) or Data Mining. Those students who are specialized in information system engineering, the logic programming becomes the core subject. Logic programming is based on the formal logic or called as declarative statements or predicate. This article will discuss and scrutinize the characteristics of logic programming by using the PROLOG programming language. The important concepts of constructing the knowledge based, facts, rules and resolution will be elaborated more by showing the example of real problem statements to solve the problem by using PROLOG. Samples of real application such the ChatGPT open AI, currently become popular apps which help users to find answers, generate details step of solution or recommend possible options for the resolution will be elucidated at the last part of this article.

Keywords: *logic programming, PROLOG, facts, rules, resolution*

Introduction

Logic Programming has been introduced for a very long time ago. Aristotle is the first important figure who placed the logic on sound systematic foundations (Kneale 1962). Frege published the first version of modern symbolic logic is known as predicate calculus (Heijenoort, 1967). The main principle of predicate calculus is to offer precise characterization for proofing methods. Basically, the logic programming is a method of programming which the program represents set of sentences in the language of symbolic logic. The program which is written in logic programming is often to be said as declarative or descriptive (Genesereth & Chaudry, 2020). Logic programming describes details application area and the goals that should be achieved. It focuses on what is true rather than how to achieve the desired goals. Conclusively the logic programming focus more on the specification rather than an implementation.

Logic programming is typically easier to understand, and the programmers are not requiring of having the extensive knowledge or skills. You don't have to choose a specific method of achieving the application goals. Logic programming is composable. The programmers don't have to make arbitrary choices and it can be combined with each other more easily. Furthermore, the logic programming is

Agile as compared to other programming paradigms such as functional and imperative. An application which developed with logic programming, can readily adapt to unexpected changes or modification to its assumptions and goals. Logic programming is also considered versatile because it can be used for multiple purposes or not specific for natural language processing only, but it also can be applied to process complicated calculation, to do the forecasting, Artificial Intelligence (AI), robotics, expert systems, software engineering and database systems.

The structure of this article will discuss example of cases solved with PROLOG in logic programming and example of application related to logic programming such ChatGPT.

Solution with Logic Programming

a) Proving of Statements

With logic programming we can proof the statements by using the PROLOG programming language. Given the following statements and we need to construct the facts and appropriate rules.

“All engineering students like to play game, eat burger and sleep in the class and that is why there are excellent in academic”

Based on the above statement we need to provide the appropriate facts and rules to support the statements. The facts and rules are as follows.

```
play(engineering, game).
eat(engineering, burger).
sleep(engineering, class).
excellent(X,A,B,C):-play(X,A),eat(X,B),sleep(X,C),
                    X=engineering, A=game, B=burger, C=class.
```

If we submit a query to determine either the engineering students are excellent if they are like to play game, eat burger and sleep in the class, the query will be as follows.

```
? - excellent (engineering, game, burger, class).
true
```

Similar query can be used if we try to ask whether the engineering students are excellent if they are like to play game, eat pizza and sleep in the class, the result of query will be false because the facts do not have any information about `eat (engineering, pizza)`.

```
? - excellent (engineering, game, pizza, class).
false
```

Through the same rule, you may proof the statement for arts students, are they excellent if they like to play game, eat burger and sleep in the class. The query resulted false as shown below.

```
? - excellent (arts, game, burger, class).
false
```

b) Proving through the Database

Given the following table shows the details result of students' assessments.

Table 1: List of students' results for three assessments

StudID	Program	Assessment1	Assessment 2	Assessment 3
2020123456	CS110	75	62	73
2020123457	CS110	81	62	94
2020123458	CS110	68	85	57
2020123459	CS110	49	43	40
2020123451	CS110	68	75	76
2020123452	CS110	66	64	68
2020123453	CS110	80	71	80
2020123454	CS110	72	64	74
2020123455	CS110	74	77	65
2020123460	CS110	59	62	51

Assume the above table consists of hundreds of students' assessments result and these records are stored in a database. In PROLOG we must represents these records as list of facts as shown below.

```
student(2020123456, 'CS110', 75, 62, 73).
student(2020123457, 'CS110', 81, 62, 94).
student(2020123458, 'CS110', 68, 85, 57).
student(2020123459, 'CS110', 49, 43, 40).
student(2020123451, 'CS110', 68, 75, 76).
student(2020123452, 'CS110', 66, 64, 68).
student(2020123453, 'CS110', 80, 71, 80).
student(2020123454, 'CS110', 72, 64, 74).
student(2020123455, 'CS110', 74, 77, 65).
student(2020123460, 'CS110', 59, 62, 51).
```

Based on the facts, we may create varieties of rules such to calculate the coursework marks and to determine the highest or lowest marks. The following is the rule to calculate the coursework mark if the weightage for assessment 1, 2 and 3 are 20%, 25, and 15% respectively.

```
coursework(StudentNumber, CW) :- student(StudentNumber, _, A, B, C),
    CW is A*0.2 + B*0.25 + C*0.15.
```

If you apply the query related to the above rule for a specific student for example the student ID '2020123456' the result will be as follows.

```
?- coursework(2020123456, Marks).
Mark = 41.45.
```

Through the same rule, you can expand to determine list of students who scored less than 30 marks. The rule is as follows.

```
cwless30(StudentNumber):-coursework(StudentNumber,CW),CW=< 30.
```

The result of the above rule is as follows. Only one student detected which the coursework mark is less than 30.

```
?- cwless30(StudentNumber).  
StudentNumber = 2020123459
```

Besides that, you may apply the existing rule to determine the highest or lowest mark of the coursework. The rules are as shown below.

```
highestMark(StudentNumber, CW):-  
    coursework(StudentNumber,CW),\+ (coursework(_,CW1),CW1 > CW) .  
  
lowestMark(StudentNumber, CW):-  
    coursework(StudentNumber,CW),\+ (coursework(_,CW1),CW1 < CW) .
```

The result of the query for the two rules as shown below.

```
?- highestMark(StudentNumber,CW) .  
StudentNumber = 2020123457,  
CW = 45.8 ;  
  
?- lowestMark(StudentNumber,CW) .  
StudentNumber = 2020123459,  
CW = 26.55 ;
```

Based on the facts, we may create several rules which related to the data as provided in the database. If the facts are not adequate, then the results of queries will be not very outstanding, or the result is not significant.

The facts as created from the database we can enhance it by providing the components of condition. For example, we want to know the grade obtained by the student based the calculation of the coursework marks. Through the specified range of marks and it's grade we need to create multiple selection in PROLOG programming language.

Given the following table shows the grade based on the range of coursework mark.

Table 2: Table of range of marks and grade

Range of Marks	Grade
0 – 15	D
16 – 30	C
31 – 45	B
46 – 60	A

The followings are the set of related rules to determine the grade.

```
gradeObtained(StudentNumber,Grade):-coursework(StudentNumber, CW),
    CW >= 0, CW =< 15, Grade = 'D'.
```

```
gradeObtained(StudentNumber,Grade):-coursework(StudentNumber, CW),
    CW >= 16, CW =< 30, Grade = 'C'.
```

```
gradeObtained(StudentNumber,Grade):-coursework(StudentNumber, CW),
    CW >= 31, CW =< 45, Grade = 'B'.
```

```
gradeObtained(StudentNumber,Grade):-coursework(StudentNumber, CW),
    CW >= 46, CW =< 60, Grade = 'A'.
```

The result of the queries are as follows.

```
?- coursework(2020123456,CW).
```

```
CW = 41.45.
```

```
?- gradeObtained(2020123456,Grade).
```

```
Grade = 'B'
```

Logic programming does not only concentrate on the calculation, but it helps human to make decision. Provided the rules as embedded in the application are comprehensive for any kinds of situation and condition to achieve the desired goals. This statement is aligned with Martin (1996) which logic programming is study of proof search.

ChatGPT – Open AI Application

ChatGPT is a complex Artificial Intelligence (AI) model developed by OpenAI. The GPT-3.5 architecture is used to develop this application. GPT stands for "Generative Pre-trained Transformer 3.5." The model has been tested and trained with massive amount of data through various sources from the internet. These allow the users to ask wide range of queries.

ChatGPT can respond the conversational interactions and simulate the conversation with a knowledgeable assistant. Generally, it can answer the questions, provides explanations, offering ideas, and suggest several options for a problem.

Nevertheless, the ChatGPT has some limitations. It may generate inaccurate or sensational answers. Since the ChatGPT struggles to give precise and up to date answers, as academicians are always needed to verify the information generated are based on the valid sources (ChatGPT, 2023).

Conclusion

Logic Programming is a powerful programming paradigm which provides a systematic and declarative approach to problem-solving. By representing the knowledge-based and mathematical computation in logical formulas, it enables the development of useful and flexible programs. Logic Programming languages, such as PROLOG, allow programmers to specify the relationships and constraints within a problem domain and let the system handle the search for best solutions. Overall, Logic Programming offers a unique and valuable approach to programming, particularly in domains that involve complex reasoning and knowledge representation. By leveraging the power of logical inference and constraint solving, logic programming provides a versatile tool for solving a wide range of problems.

References:

- ChatGPT, *Introduction of ChatGPT*, Chat conversation, Open AI, 19 May 2023, <https://openai.com>, accessed on 19 May 2023.
- Genesereth, M., & Chaudhri, V. K. *Introduction to Logic Programming*. Morgan & Claypool, 2020.
- Heijenoort, V., J. Ed. From Frege to Godel; *A Source Book in Mathematical Logic*, 1879-1931. Harvard University Press, 1967.
- Kneale, W.C. and Kneale, M. *The Development of Logic*. Oxford, 1962.
- Martin, P.L., *On the meanings of the logical constants and the justifications of the logical laws*. Nordic Journal of Philosophical Logic, 1(1):11–60, 1996.