20CYS312 - Principles of Programming Languages
Exploring Programming Paradigms

Assignment-01

Presented by Niran R
CB.EN.U4CYS21048
TIFAC-CORE in Cyber Security
Amrita Vishwa Vidyapeetham, Coimbatore Campus

Feb 2024

Outline

1. LOGIC
2. Logic - Prolog
3. Declarative
4. Declarative - SQL
5. Comparison and Discussions
6. Bibliography

Paradigm: Logic

Logical programming is a paradigm of programming that is based on mathematical logic.
Well-known language associated with logical programming is Prolog.
Common examples of logical programming languages include Prolog and Datalog. These languages provide a framework for expressing knowledge and solving problems in domains that require logical reasoning and rule-based systems. Logical programming is particularly well-suited for tasks where relationships, conditions, and deductions play a central role in problem-solving.

Used in:
Artificial Intelligence (AI)
Natural Language Processing (NLP)
Expert Systems
Knowledge Representation
Symbolic Reasoning
Databases
Semantic Web and Ontologies
Planning and Scheduling
Medical Diagnosis
Game Development

Prolog and its logical nature:

–>Prolog, a logic programming language, distinguishes itself with its declarative paradigm, providing a unique approach to problem-solving. In

–>Prolog, programmers specify relationships, rules, and logical conditions, letting the system deduce solutions based on these declarations. This declarative nature aligns with the fundamental principle of expressing "what" needs to be accomplished rather than "how" to achieve it.

–>Logic programming in Prolog revolves around the formulation of facts and rules. Facts represent statements assumed to be true, while rules define logical relationships and conditions. Prolog programs, often composed of Horn clauses, encapsulate this declarative knowledge.

–>The heart of Prolog's declarative power lies in its ability to perform logical inference. Through a process of resolution, Prolog navigates through the specified rules and facts to determine the validity or falsity of queries. This characteristic makes Prolog particularly well-suited for applications in artificial intelligence, expert systems, and symbolic reasoning.

Declarative Programming:

–>Declarative programming is a programming paradigm that focuses on specifying what should be achieved, leaving the details of how to achieve it to the underlying system. This is in contrast to imperative programming, where the emphasis is on specifying ho to perform a computation.

–>declarative programming languages include SQL for database queries, Prolog for logic programming, and Haskell for functional programming. Each of these languages allows developers to express computations in a declarative manner, providing a higher level of abstraction and often leading to more concise and elegant code.

Used in:
Database Systems
Web Development
Artificial Intelligence and Expert Systems
Functional Programming
Markup Languages
Natural Language Processing (NLP)

SQL and its declarative nature:

–>Structured Query Language (SQL) stands as a cornerstone in the realm of relational databases, offering a declarative approach to managing and querying data.
–>Unlike procedural languages, SQL allows users to express what information they desire without explicitly outlining how to retrieve or manipulate it.
–>This declarative nature simplifies the interaction with databases, making it accessible to both novice and experienced users.
–>SQL's primary function is data retrieval, achieved through the SELECT statement.

Users specify the desired columns, tables, and any filtering conditions, leaving the database management system to determine the most efficient execution plan.
–>This focus on "what" rather than "how" characterizes SQL as a declarative query language.
–>Beyond data retrieval, SQL provides statements for data modification, enabling users to insert, update, or delete records from tables.
–>These statements follow the same declarative principle – users express the desired changes, and the system handles the execution intricacies. **Prolog vs SQL:**

Paradigm:
Prolog: Logic programming paradigm, emphasizing rules, logical relationships, and symbolic reasoning.
SQL: Declarative query language designed for managing and querying relational databases.
Primary Use:
Prolog: Artificial intelligence, expert systems, symbolic reasoning.
SQL: Database management, data retrieval, modification, and schema definition.
Programming Model:
Prolog: Logic programming with facts, rules, and logical inference.
SQL: Declarative language for expressing data queries and modifications.
Data Manipulation:
Prolog: Expresses data manipulation through logical relationships and rules.

SQL: Specifically designed for efficient data manipulation in relational databases.

Variables:

Prolog: Utilizes variables for abstraction and unification.

SQL: Uses placeholders for parameters in queries.

Recursion:

Prolog: Supports recursion as a fundamental part of its programming model.

SQL: While not supporting explicit recursion, some databases offer recursive query capabilities.

Prolog and SQL:

–>Prolog and SQL, though designed for distinct purposes, share notable similarities in their declarative nature and reliance on logical constructs. Both languages adhere to the declarative programming paradigm, wherein users specify what they want to achieve rather than outlining the procedural steps to achieve it.

–>In the realm of querying, SQL is explicitly a query language for relational databases, where users declare the desired result set without detailing the intricacies of data retrieval. Prolog, while not inherently a query language, often serves a similar purpose, allowing users to express queries when searching for solutions based on logical rules and relationships.

–>A common thread between Prolog and SQL lies in their use of logical constructs. Prolog, being a logic programming language, employs logical rules, facts, and an inference mechanism for problem-solving. SQL leverages logical conditions, joins, and set

operations to express queries and manipulate data based on relational logic.
–>Both languages make use of variables, albeit with different purposes. Prolog uses variables for abstraction and unification in logical rules, facilitating the representation of general relationships. SQL employs variables as placeholders for parameters in queries, enabling dynamic conditions and enhancing the flexibility of data retrieval.
–>Set operations are another shared aspect, with SQL providing explicit support for operations like UNION, INTERSECT, and EXCEPT to manipulate sets of data. While Prolog lacks direct set operations, its logical constructs and backtracking mechanisms can achieve similar outcomes when searching for solutions.

Bibliography:

www.geeksforgeeks.org
www.imperial.ac.uk
www.linode.com
www.computerhope.com

References