20CYS312 - Principles of Programming Languages
Exploring Programming Paradigms

Assignment-01

Presented by Aravind S
CB.EN.U4CYS21009
TIFAC-CORE in Cyber Security
Amrita Vishwa Vidyapeetham, Coimbatore Campus

Feb 2024

AMRITA
VISHWA VIDYAPEETHAM

Outline

Functional Programming Paradigm

Functional programming is a programming paradigm that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data.

Key concepts and features commonly associated with functional programming:

- Immutable Data
- First-Class and Higher-Order Functions
- Pure Functions
- Recursion
- Lambda Calculus
- Pattern Matching

Languages that follow functional programming paradigm:
JavaScript, Haskell, Scala, F, etc

Real-World Application: WhatsApp (Erlang), VScode (F#)

Implementation in JavaScript

```javascript
// Immutability using a spread operator
const originalArray = [1, 2, 3];
const newArray = [...originalArray, 4]; // Creates a new array with 1, 2, 3, 4

// Pure function example
const square = (x) => x * x; // Always returns the square of the input

// Function composition
const addThenSquare = (x) => square(x + 1); // Composes square with addition

// Higher-order function example
const map = (fn, array) => array.map(fn); // Applies a function to each element
const squaredValues = map(square, [1, 2, 3]); // Returns [1, 4, 9]
```

Functional - F#

F# is open-source, cross-platform programming language which allows you to write uncluttered, self-documenting code. F# is a JavaScript and .NET language used for web, cloud, data-science, apps and more.

Some Features of F#:

- Lightweight Syntax
- Immutable by Default
- Pattern Matching
- Async Programming

Paradigms Supported: Functional, Object Oriented, Imperative

File Extension: .fs

Current Version: 8.0.0 (Nov 14, 2023)

Implementation in F#

```fsharp
// Immutable data
let x = 10
let y = 20

// Pure function
let add x y = x + y

// Recursion
let rec factorial n =
    if n = 0 then 1
    else n * factorial (n - 1)

// Pattern matching
type Shape =
    | Circle of radius: float
    | Rectangle of width: float * height: float

let area shape =
    match shape with
    | Circle r -> 3.14 * r * r
    | Rectangle (w, h) -> w * h

// Partial application
let addTen = add 10
let result = addTen 5

printfn "Immutable data: x = %d, y = %d" x y
printfn "Pure function result: %d" (add x y)
printfn "Factorial of 5: %d" (factorial 5)
printfn "Circle area: %.2f" (area (Circle 3.0))
printfn "Rectangle area: %.2f" (area (Rectangle (4.0, 5.0)))
printfn "Partial application result: %d" result
```

```
Immutable data: x = 10, y = 20
Pure function result: 30
Factorial of 5: 120
Circle area: 28.26
Rectangle area: 20.00
Partial application result: 15
```

Logic Programming Paradigm

Logic programming is a programming paradigm based on formal logic. A program in a logic programming language is a set of sentences in logical form, expressing facts and rules about the problem domain.

Key concepts and features of logic programming:

- Facts and Rules
- Logic variables
- Backtracking
- Unification
- Horn Clauses
- Querying

Languages that follow logic programming paradigm:
Prolog, Datalog, Mercury, etc

Real-World Application: AI Research

Implementation in Prolog

```
% Facts: Parent-Child Relationships
parent(john, jim).
parent(john, ann).
parent(jim, lisa).
parent(ann, bob).

% Rule: Grandparent Relationship
grandparent(X, Z) :- parent(X, Y), parent(Y, Z).

% Query: Who are the grandparents of Lisa?
?- grandparent(X, lisa).
```

Logic - Mercury

Mercury is a logic programming language which combines the clarity and expressiveness of declarative programming with advanced static analysis and error detection features. It is primarily used in academic and research settings.

Some Features of Mercury:

- Lightweight Syntax
- Immutable by Default
- Pattern Matching
- Async Programming

Paradigms Supported: Logic, Functional

File Extension: .m

Current Version: 22.01.8 (Sep 17, 2023)

Implementation in Mercury

```
% Define a relation for sibling
:- type person ---> alice ; bob ; carol ; david.
:- pred sibling(person::in, person::in) is semidet.
sibling(alice, bob).
sibling(bob, alice).
sibling(bob, carol).
sibling(carol, bob).
sibling(carol, david).
sibling(david, carol).

% Define a relation for cousin
:- pred cousin(person::in, person::in) is semidet.
cousin(X, Y) :- sibling(ParentX, X), sibling(ParentY, Y), cousin(ParentX, ParentY).
cousin(X, Y) :- sibling(ParentX, Y), sibling(ParentY, X), cousin(ParentX, ParentY).
cousin(X, Y) :- sibling(ParentX, X), sibling(ParentY, Y), X \= Y.

% Define a relation for grandparent
:- pred grandparent(person::in, person::in) is semidet.
grandparent(Grandparent, Grandchild) :- sibling(Parent, Grandparent), parent(Parent, Grandchild).

% Define a relation for parent
:- pred parent(person::in, person::in) is semidet.
parent(alice, carol).
parent(bob, carol).
parent(carol, david).

% Query some relationships
:- io.write_string("Cousins: "), (cousin(alice, david) -> io.write_string("Yes\n") ; io.write_string("No\n")),
   io.write_string("Grandparent: "), (grandparent(alice, david) -> io.write_string("Yes\n") ; io.write_string("No\n")),
   io.nl.
```

Comparison and Discussion

- Functional programming focuses on the evaluation of mathematical functions while logic programming focuses on defining logic-based rules and relationships.
- Functional programming is Well-suited for parallel and concurrent programming, while logic programming is useful in areas such as artificial intelligence, expert systems, and knowledge representation.
- Both, functional and logic programming are order independent.
- Logic programming may have side-effects while functional programming never have those.
- Pure Functional Programming Language: Haskell (Mostly)
- Pure Logic Programming Language: Prolog
- Languages which support both programming paradigm: Mercury, Scala, Erlang, Racket etc.

Bibliography

https://www.geeksforgeeks.org/functional-programming-paradigm/

https://www.geeksforgeeks.org/difference-between-functional-and-logical-programming/

https://en.wikipedia.org/wiki/Logic_programming

https://learn.microsoft.com/en-us/dotnet/fsharp/what-is-fsharp

https://www.javatpoint.com/what-is-f-sharp

https://en.wikipedia.org/wiki/Mercury_(programming_language)

https://mercurylang.org/about.html