SICP Midterm Examination Points

Jiacai Cui

Functions & Control

Expression & Value

Name & Assignment

Function Definition

Function Call (Eval & Apply)

Pure / Non-Pure Functions (Print vs Return, None)

Conditional Control (If-Statement)

Boolean Expressions & Short Circuiting

Iterative Control (While-Statement)

Error Messages

Environment Diagrams

Assignment Statements

Def Statements

Call Expressions

Variable Lookup

Lambda Expressions

Recursion

Lists

    Box and Poinger Notation

    Lists in Lists in Lists

Default Arguments

Higher-Order Functions

Lambda Expressions vs. Def Statements

Function Design & Generalization

Functions As Arguments

Functions That Return Functions

Function Currying

Self Reference

Recursion

Functional Abstraction

Definition of Recursion

Structure of a Recursive Function (Divide - Conquer - Combine)

Tree Recursion & Recursion Tree

Mutual Recursion

Iteration vs. Recursion

Recursion Examples:

    Sum-Digits

    Cascade

    Fibonacci

    Count-Partitions

Sequences

Sequence Abstraction

Sequence Operation (Expression)

    Get Item

    Slicing

        Slicing Creates New Values

    Check Membership

Concatenate

Sequence Processing (Statement)

   Iterating

   Unpacking

Range

List

   List Comprehensions

String

   String Literals (Three Forms)

Sequence Aggregation (Sum, Max / Min, All / Any)

Dictionary

Data Abstractions & Trees

Abstraction Barrier (between Representation & Use)

Constructor & Selectors

Abstract Data Type

Closure

Tree Abstraction (Tree = Label + [Branches])

Tree Implementation

   Constructor: `tree`

   Selectors: `label`, `branches`

   Convinence Functions: `is_tree`, `is_leaf`

Tree Processing (using Recursion)

Mutablility

Objects

Mutable Objects

   Lists (Pop, Append, Remove, Extend, Slice Assignment)

Dictionaries

Mutate with Functions

Immutable Objects (Tuples)

Identity & Equality

Default Arguments (Mutable vs. Immutable)

Mutable Functions (Nonlocal)

    UnboundLocalError

    Mutate Mutable Values wihout Nonlocal

Iterable

Iterators (iter & next)

Views of a Dictionary (keys & values)

Built-in Functions for Iteration

    Create Iterators: `map`, `filter`, `zip`, `reversed`

    View Iterators: `list`, `tuple`, `sorted`

Generators and Generator Functions

    yield value

    yield from iterator / iterable

    Generators Representing Infinite Sequences