Computable Functions

John Mitchell

Reading: Chapter 2

Foundations: Partial, Total Functions

• Value of an expression may be undefined
  — Undefined operation, e.g., division by zero
  — 3/0 has no value
  — Implementation may halt with error condition
  — Nontermination
    • \( f(x) = \) if \( x=0 \) then 1 else \( f(x-2) \)
    • this is a partial function: not defined on all arguments
    • cannot be detected at compile-time; this is halting problem
  — These two cases are
    • "Mathematically" equivalent
    • Operationally different

Partial and Total Functions

- Total function: \( f(x) \) has a value for every \( x \)
- Partial function: \( g(x) \) does not have a value for every \( x \)

Functions and Graphs

- Graph of \( f(x) \): \( \{(x,y) \mid y = f(x)\} \)
- Graph of \( g(x) \): \( \{(x,y) \mid y = g(x)\} \)
Mathematics: a function is a set of ordered pairs (graph of function)

Partial and Total Functions

• Total function \( f: A \rightarrow B \) is a subset \( f \subseteq A \times B \) with
  — For every \( x \in A \), there is some \( y \in B \) with \( (x,y) \in f \) (total)
  — If \( (x,y) \in f \) and \( (x,z) \in f \) then \( y = z \) (single-valued)
• Partial function \( f: A \rightarrow B \) is a subset \( f \subseteq A \times B \) with
  — If \( (x,y) \in f \) and \( (x,z) \in f \) then \( y = z \) (single-valued)
• Programs define partial functions for two reasons
  — partial operations (like division)
  — nontermination
    \( f(0) = \) if \( x=0 \) then 1 else \( f(x-2) \)

Halting Problem

Entore Buggati: "I build cars to go, not to stop."

Self-Portrait in the Green Buggati (1925)
Tamara DeLempicka
Computability

• Definition
  Function \( f \) is computable if some program \( P \) computes it:
  For any input \( x \), the computation \( P(x) \) halts with output \( f(x) \)

• Terminology
  Partial recursive functions
  = partial functions (int to int) that are computable

• Church-Turing Hypothesis
  The programming language doesn’t matter – all “reasonable” programming languages define the same class of computable functions

Halting function

• Decide whether program halts on input
  - Given program \( P \) and input \( x \) to \( P \),
  \[ Halt(P, x) = \begin{cases} 
  \text{yes} & \text{if } P(x) \text{ halts} \\
  \text{no} & \text{otherwise}
\end{cases} \]

Clariﬁcations
- Assume program \( P \) requires one string input \( x \)
- Write \( P(x) \) for output of \( P \) when run in input \( x \)
- Program \( P \) is string input to \( Halt \)
- Represent two inputs \( P, x \) as string \( P\!x \) (for example)

Theorem: There is no program for \( Halt \)

Unsolvability of the halting problem

• Suppose \( P \) solves variant of halting problem
  On input \( Q \), assume
  \[ P(Q) = \begin{cases} 
  \text{yes} & \text{if } Q(Q) \text{ halts} \\
  \text{no} & \text{otherwise}
\end{cases} \]

• Build program \( D \)
  \[ D(Q) = \begin{cases} 
  \text{run forever} & \text{if } Q(Q) \text{ halts} \\
  \text{halt} & \text{if } Q(Q) \text{ runs forever}
\end{cases} \]

• Does this make sense? What can \( D(D) \) do?
  - If \( D(D) \) halts, then \( D(D) \) runs forever.
  - If \( D(D) \) runs forever, then \( D(D) \) halts.
  - CONTRADICTION: program \( P \) must not exist.

Examples

• Is there an algorithm to decide whether this program has a run-time type error?
  if \( f(x) \) then \( y=1+"Bob" \) else \( y=2+"Alice" \)

• Is there an algorithm to decide whether this program reads variable \( z \)?
  if \( f(x) \) then \( y=z+"Bob" \) else \( y=z+"Alice" \)

Garbage Collection

• Garbage:
  At a given point in the execution of a program \( P \), a memory location \( m \) is garbage if no continued execution of \( P \) from this point can access location \( m \).

• Garbage Collection:
  - Detect garbage during program execution
  - GC invoked when more memory is needed
  - Decision made by run-time system, not program

Main points about computability

• Some functions are computable, some are not
  - Halting problem
  - Other problems that are equivalent

• Programming language implementation
  - Can report error if program result is undefined due to division by zero, other error condition
  - Cannot warn user if program will not terminate
  - Many useful program properties are not computable