Q1. Describe 5 hashing strategies/functions, what are different types of hash resolution techniques.

Answer

Introduction to Hashing

Hashing is a fundamental technique in Computer science that transforms data into fixed-size hash values, enabling efficient data retrieval & storage. It is widely used in applications such as data indexing, Password Storage, & ensuring data integrity. Hashing functions assign unique identifiers to data, facilitating quick access & retrieval in various algorithms & data Structures.

Hashing strategies/functions :

1 Division Hashing

This simple hashing strategy involves dividing the key by a Prime number & using the remainder as hash value.

Example If key is 54 & divisor is 11, hash value would be 10.

2 Multiplicative Hashing

Multiply key by constant A in range (0, 1) & extract the fractional Part of of the Product. Then, multiply this fraction part by the size of hash table to get hash value.

Example With key 27, Constant A = 0.618, & hash table size 100, hash value is floor(27 × 0.618 % 1 * 100) = 6

3 Universal Hashing

Use family of hash functions; randomly selecting one at runtime. This reduces the likelihood of collisions, improves overall performance.

Example If we have a family of hash function $H = \{h_1, h_2, h_3 \cdots h_n\}$, we randomly choose one function from family to hash the key.

4 SHA-256 (secure Hash Algo 256-bit)

Part of SHA-256 family, this cryptographic hash function transforms input data into a fixed-size (256-bit) hash value, ensuring a secure & irreversible Process.

Example For input "hello", SHA-256 generates the hash Value "2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824"

5 Hash/City Hash

Developed by Google, cityhash is designed for fast hashing of Short Character Strings. It operates in multiple rounds with a Variety of shifts & bitwise operations.

Given the input "example-string", City Hash Produces the hash value "102067849232353369
64."

Hash Collision Resolution Techniques:

1 Chaining

166

In chaining, each slot in the hash table maintains a linkedlist of elements that hash to same location. Collision are resolved by appending elements of list.

Example If two keys hash to same location, they are simply added ~~ad~~ to linked list at that position.

2 Open Addressing - Linear Probing

In linear Probing, if a collision occurs, the algorithm searches for next available slot linearly until an empty slot found

Example If the initial hash index is occupied, the algorithm Checks next index & Continues untill an empty slot is found.

3 Double Hashing

Double Hashing involves using a secondary hash function to determine the interval between Probe attempts, Providing a moue & varied exploration of hash table.

Example If Collision occurs at index H, double hashing might attempt to insert the key at indices $H+f(k), H+2*f(k), H+3*f(k)$ & So on.

These hash Collision resolution techniques Shows the diverse approaches employed in Computer Science to efficiently manage & ~~retrie~~ retrieval of data. Each method have It's strengths & weakness, making the Suitable for specific use Cases & applications.

Q2 Write note Of Chomsky hierachy & mention What kinda machine will be used (make diagram of Turing machine used with each type of Gramm in Chomsky hierarchy.

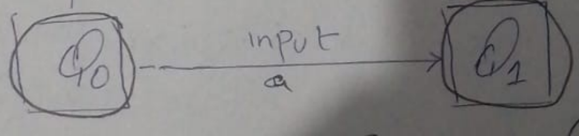
Answer

The Chomsky Hierarchy, Proposed by lnguist & Cognitive Scientist NoamChomsky, classifies formal grammar into four types based on their generative power. Each type corresponds to a different class of language, & various machiy models, includes Turing machines, are assouated with these grammatical types.

1 Type-3 - Regular Grammar (Regular language)

Machine Model: Finite State Machine (FSM)

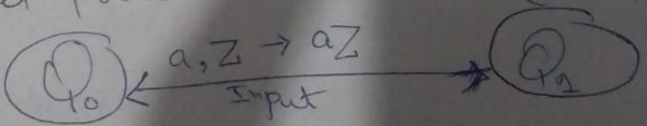
Diagram:



2 Type 2 - Context-Free Grammar (Context-Free language)

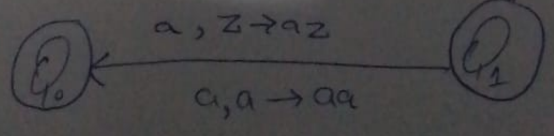
Machine Model: Pushdown Automation (PDA)

Diagram:



3 Type-1 Context Sensitive Grammar

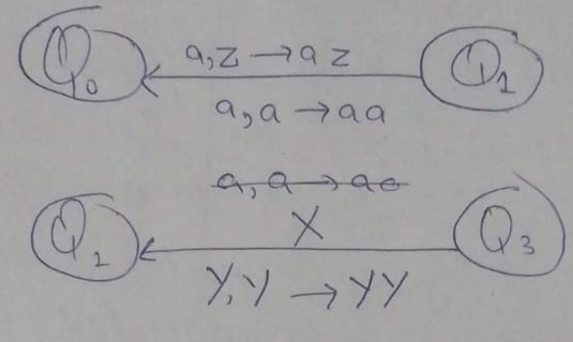
Machine Model: Linear Bounded Automation (LBA)



4 Type-0 Unrestricted Grammar (Recursively Enumerable languages)

Machne Model: Turing Machine (TM)

Diagram:



In Chomsky as we Progress from Type-3 to type-0, the generative power of the grammar increases, & the associated Machine models become more Complex. Regular languages are recognized by finite State machines, Context free languages by Pushdown automata, Context Sensitive Languages by linear-bounded automata; & recursively enumerable languages by Turing machines

These models represents the theoretical foundation for understanding the Computation Capacity of different types of formal languages