Transaction and Transaction Flow

- What is transaction?
  - (User oriented) unit of processing;
  - In graphs: transaction tokens
    - carried by customer/client.
  - Contrast with task (liveness).
  - Processed/handled by system
    - system services and structure
    - routing through the system
    - depend on sys. arch. & token type

- Examples of transactions:
  - Bank transaction/service
  - Database transaction/operation
  - Processes in operating systems
  - Workload in queuing networks
Transaction and System Descriptions

- System architecture
  - Augmenting control flow graphs
  - Node: specific service provided.
  - Link: routing through system
  - Addition of transaction tokens ⇒ relations among tokens.
  - Token sensitive routing possible

- Transactions description
  - Transactions represented by tokens
  - Present at links (and nodes)
  - Transaction control record
    - types of transactions
    - relation to other transactions
    - relation to system architecture
    - priority info., etc.
  - State: value in the record.
Transaction and Queuing

- Transaction/token handling and queuing
  - Multiple tokens to a single link
  - Service decisions = queuing discipline
  - Information carried by tokens:
    - time-stamp/history/priority/etc.
    - may refer to trans. control record

- Queuing principles at single server:
  - Priority: time vs. other:
    - time: FIFO/FCFS, LIFO/stack, etc.
    - other/explicit: SJF, priority#, etc.
  - Pre-emption allowed?
  - (Buffer) bounded or unbounded?
  - Mixture/combination of queues
  - Batch and synchronization
Testing a Single Queue

- Test case design/selection:
  - Conformance to queuing discipline;
  - Boundary conditions:
    - bounded queue with bound B
    - lower bound: around 0 (always)
    - server busy/idle at lower bound
    - upper bounds: around B (bounded Q)
  - Typical case
  - Mixed queues: test case combinations
    - different priority classes
    - FIFO within priority class
    - similar to segment comb. in CFT

- Test case sensitization:
  - Markings and traces
    - to confirm service order
    - (based on arrival/departure order)
  - Discipline under each case
  - Dynamic nature of arrivals
Queuing Model

- As a special case of transaction flow model:
  - Transactions = customers;
  - Node = server; link = queue;
  - Limited splitting/merging;
  - Model = queuing network.

- Model construction:
  - (Existing queuing models)
  - Specifications and workloads;
  - Performance models;
  - Specify queuing discipline;
  - Boundedness.
Queuing Model Testing

- Sensitization:
  - Transaction/workload generator;
  - (Re)use perf./simulation models;
  - Adjustable queues.

- Confirmation and cross validation:
  - Existing oracles.
  - Analytical models:
    - M/M/1 queues and queuing networks.
  - Performance benchmarks.

- Applications:
  - As part of transaction flow testing.
  - Performance testing.
Tokens and Token Synchronization

- Token synchronization
  - Multiple inlink with tokens
    - one token processed: same as CFT
    - multiple processed: synchronization
  - Relating incoming and outgoing tokens

- Synchronization types
  - *Merger* of two incoming tokens:
    - a new outgoing token generated
    - similar to DFT in “A ← B + C”
    - both B and C present to generate A
  - *Absorption* of one token by another:
    - only one incoming token survives
    - similar to DFT in “S ← S + X”
    - both S (orig) and X present
      ⇒ S (updated)
  - Graph: node with token in it
  - Multi-way merger/absorption similar
Synchronization Testing

- Correct tokens
  - Correct input tokens
  - Merger or absorption?
  - Corresponding correct output tokens
  - What we did already in DFT

- Synchronization of arrivals:
  - Combinations of input token in different arriving order
  - Example with two way synchronization:
    - nothing arrives ⇒ no output
    - one arrives ⇒ no output
    - two arrive (3 cases: A-B, B-A, AB)
      ⇒ correct token generated
  - Multi-way synchronization:
    - similar steps, but more cases
    - staged vs. single synchronization
    - trade-offs: cost vs. accuracy
  - Combination with correct tokens