Lecture 4: Reachability Analysis

Software Quality Assurance (INSE 6260/4-UU)
Winter 2016
INSE 6260/4-UU

Software Quality Assurance

- Software Quality
  - Factors and Models
  - Metrics
- Quality Assurance
  - Inspection
  - Testing Techniques
  - Reachability Analysis
Reachability Analysis

- Reachability analysis deals with determining the set of reachable states in a state-transition-based system model.

- The reachable states information is useful in:
  - Formal verification (proving a property for reachable states).

- Reachability analysis of large systems is a complex task attracting significant research efforts.
  - Local view vs. global view.
The concept of states is a higher level description, compared to structure.

A state characterizes the behavior of a system, given a fixed set of values of the memory elements.

**Initial state**: A state, in which the system begins its functioning.

**Reachable state**: A state that can be reached from the initial one though a finite sequence of transitions under allowed inputs.
Reachability Analysis

- Reachability analysis shows the product space of the two processes and the signal queues of their input ports.
- To perform the reachability analysis, the first step is to transform the process graph into a transition system.
- Each transition consists of either an input or an output.
Reachability Analysis For Communicating Processes

Communicating Finite State Machines Model CFSM

Communication software design errors:

I) Semantic errors which cause the provision of incorrect services to the distributed system users that affect the safety of the system

II) Syntactic errors which ultimately cause the protocol deadlock, and therefore affects the liveness propriety of the system
Safety and Liveness

- Safety properties specify that “something bad never happens”
- Doing nothing easily fulfills a safety property
  - as this will never lead to a “bad” situation
- Safety properties are complemented by liveness properties
  - that require some progress
- Liveness properties assert that:
  - “something good” will happen eventually

[Lamport 1977]
Definitions

A global state is a pair of $<S,C>$, where $S = (s_1, s_2, \ldots, s_n)$ is a vector of states and $C$ is a vector of channels.

$s_1, s_2, \ldots, s_n$ represent the current states of processes $P_1, P_2, \ldots, P_n$.

$C = (C_{ij}, \text{for all } i \neq j, \text{and } i, j \leq n)$ represents the current contents of the channels $C_{ij}$ linking the processes.

The initial (final) global state of a system composition is a pair $<S,C>$ in which each of the component states of $S$ are initial (final) states in their respective processes, and all channels are empty.
The next global state is obtained as follows
\(<S,C> \rightarrow <S',C'>\) if there exists a transition that can be executed or fired. Two cases are possible:

1) There exists a transition for a process where the message \(x\) can be sent;

2) There exists a reception transition of a message \(x\) for a process and a message \(x\) is in the input queue
Unspecified reception or incompleteness

- A reception of message $x$ at the state $S$ of $P_i$ is said to be Unspecified iff there exists a reachable global state $<S,C>$, in which the following is true:
  
  there exists a message in the queue and a reception transition but the message in the queue is different from the message expected by the reception transition.
Syntactic Design Errors for Communicating Processes

Deadlocks

• A global non-final state $<S,C>$ is a deadlock if the channels are empty and reception transitions are unspecified for some messages and some individual processes are not in a final state

Livelocks

• A system is in the livelock (dynamic deadlock) if the processes are exchanging messages that are not useful for providing the service
Reachability Graph (Tree)

- **Reachability Graph (Tree)**
  - *Nodes* represents regular global states
  - *Directed arc*, connecting two nodes or states \(<S1,C1>\) and \(<S2,C2>\), corresponds to a transition in one of the communicating processes
  - *Root* of the Tree corresponds to the initial state \(<S0,C0>\)
  - A *path* in the Tree corresponds to an execution sequence of the interleaved receptions and transmissions, and it represents the reachability of the last state in the path from the initial state of the path
Reachability Graph (Tree)

- The expansion of the tree from a particular node stops if one the following conditions is satisfied:
  - 1. the node already exists in the tree
  - 2. the node corresponds to a deadlock state
  - 3. an unspecified reception error is detected at the node
  - 4. the node corresponds to a final state
Reachability Analysis is a *global state exploration process* that:
- Starts from the initial global state, and
- recursively explores all possible transitions that lead to new global states
- The result is a *reachability graph*, which captures all possible states
Example

P1 and P2 Communicate with FIFO queues
Reachability Tree
Modified P2

Diagram showing relationships between nodes labeled 0, 1, 2, 3, and 4 with edges marked by numbers (+2, -2, +3, -4, +4, -3, -4, +2).
Reachability Tree
Communication Software Design Errors - Deadlock

- **Deadlock**
  - Both machines are at receiving states and channels are empty. The network can not progress further.
  - **Receiving state** is a state where all its outgoing transition are all receiving transition. It cannot move without msg in its incoming channel.

![Diagram of communication software design errors - deadlock](image)
Communication Design Errors - Unspecified Reception

- Unspecified reception
  - There exists a message in the queue and a reception transition, but the message in the queue is different from the message expected by the reception transition. The network cannot progress further.
  - E.g. there is msg B in C2 but sender does not have a receiving transition with msg B.
Non-executable States and Transitions

- E.g. State 3 of Receiver will never be executed or become the current state. It is called **non-executable state**
- Transitions (2,3,+B) and (3,1 ,-C) will never be executed. They are called **non-executable transitions**
Unboundedness or overflow

- If a software state can be reached such that the communication channel contains a number of messages exceeding the predefined capacity for that channel
Software Verification Using Reachability Analysis

- A global state \(<S,C>\) is said to be reachable from the initial global state \(<S_0,C_0>\), denoted \(<S_0,C_0> \Rightarrow^* <S,C>\), IFF:
  - There exists an execution path consisting of the interleaving of message receptions and transmissions that takes the system of communicating processes from the initial global state \(<S_0,C_0>\) to \(<S,C>\)
Reachability Analysis Process

- Starting from initial global state, where channels are empty and machines at their initial states
- Explore all possible reachable states by firing all the possible transitions (and generating global states) from any given reachable state
- All deadlock and unspecified reception errors will be captured and marked as individual global state
- By examining the number of messages in the channels we can design the buffer size for the software
- We can detect non-executable states and transitions by marking those states that are touched and transitions that are fired during the reachability analysis
Reachability Analysis Example 1

CFSMs

Global state/reachable state

Sender's State  Channel C1's content  Channel C2's content  Receiver's State

gsn – Global State ID

Reachability Graph

Unspecified Reception, Receiver do not know how to receive B

E: channel empty
Reachability Analysis Example 2
Reachability Analysis Example 3

a) Perform the reachability analysis on the Network (M, N)
b) What sizes of buffers are needed for the two FIFO channels?
Reachability Analysis Example 3 - Solution

- One unspecified reception (see gs4)
- Both channels need buffer size of 2 (see gs8 and gs9)
- No non-executable states and transitions

Figure 1. Reachability graph.
Reachability Analysis Example 4

- Perform the reachability analysis on the Network (P1, P2)
- Find deadlock, unspecified reception global states in the software
- Are there non-executable transitions and nodes in the CFSMs
- How many buffers are required in each of the two channel?
Reachability Analysis Example 4 - Solution

- No deadlock or unspecified reception
- Non-executable
  - P1: S2, (S1, S2, +5), (S2, S1. -4)
  - P2: S3, (S2, S3, +4), (S3, S1. -5)
- Buffer
  - C21: 2
  - C12: 1
Given the following network of two communicating finite state machines,

a) Perform the reachability analysis on the Network (M, N).

b) What sizes of buffers are needed for the two FIFO channels?

c) Are there non-executable states or transitions?
Pros and Cons of Reachability Analysis

- **Advantages:**
  - Easily automated
  - Many logical errors can be detected by only examining individual global states in the reachability graph

- **Disadvantages:**
  - State space explosion problem
  - Does not work on unbounded protocols
  - Many relationships among the software state variables, expressing the desirable logical correctness properties of the software are not apparent from simply traversing the reachability graph