Model Checking Lab

Meeting III: Phase Queen Consensus Protocol

Haidi Yue

Software Modeling and Verification Group

Nov. 11, 2010

Distributed Consensus Problem

Problem Discription

Given:

- n processes
- t of them are unreliable (they transmit unreliable messages)

A protocol solves the DC problem, if

- at termination, reliable process (there are $n - t$) have the same local value
- if all reliable process has the same initial value then their final value is the same as the common initial value.

The Phase Queen Consensus Protocol

The Algorithm

```
for k := 1 to t + 1 do
begin
    send(V);                    /* universal exchange */
    for j := 0 to 1 do
        C[j] := the number of received j's;
    V := (C[1] > n/2);          /* local decision */
    if k == p then
        send(V);                /* Queen's Broadcast */
    if C[V] < n − t then
        V := the received message
end;
```

Modeling the Phase Queen Algorithm

Challenges

Naive models are unusable due to state space explosion:

- PIDs of the unreliable processes?

Modeling the Phase Queen Algorithm

Challenges

Naive models are unusable due to state space explosion:

- PIDs of the unreliable processes?

$$\frac{n!}{(n-t)!} \text{ different possibilities}$$

Modeling the Phase Queen Algorithm

Challenges

Naive models are unusable due to state space explosion:

- PIDs of the unreliable processes?

$$\frac{n!}{(n-t)!} \text{ different possibilities}$$

- Initial values of the processes?

Modeling the Phase Queen Algorithm

Challenges

Naive models are unusable due to state space explosion:

- PIDs of the unreliable processes?

  $\frac{n!}{(n-t)!}$ different possibilities

- Initial values of the processes?

  $2^{(n-t)}$ different possibilities

Modeling the Phase Queen Algorithm

Challenges

Naive models are unusable due to state space explosion:

- PIDs of the unreliable processes?

  $\frac{n!}{(n-t)!}$ different possibilities

- Initial values of the processes?

  $2^{(n-t)}$ different possibilities

- Interleaving of broadcast transmission and reception.

Modeling the Phase Queen Algorithm

Challenges

Naive models are unusable due to state space explosion:

- PIDs of the unreliable processes?

  $\frac{n!}{(n-t)!}$ different possibilities

- Initial values of the processes?

  $2^{(n-t)}$ different possibilities

- Interleaving of broadcast transmission and reception.
- **Non-deterministic** behaviour of unreliable processes.

Modeling the Phase Queen Algorithm

Challenges

Naive models are unusable due to state space explosion:

- PIDs of the unreliable processes?

  $\frac{n!}{(n-t)!}$ different possibilities

- Initial values of the processes?

  $2^{(n-t)}$ different possibilities

- Interleaving of broadcast transmission and reception.
- **Non-deterministic** behaviour of unreliable processes.

Modeling Reduction

Modeling ideas

- Modeling the broadcasting as a matrix.

Modeling Reduction

Modeling ideas

- Modeling the broadcasting as a matrix.

```
typedef Achan {
            chan ch[n] = [1] of {bit};
}
Achan Mchan[n];
```

Modeling Reduction

Modeling ideas

- Modeling the broadcasting as a matrix.

```
typedef Achan {
        chan ch[n] = [1] of {bit};
}
Achan Mchan[n];
```

- Given process Ids $\{0, ..., n-1\}$, let $\{0, ..., t-1\}$ be the Ids of unreliable processes.

Modeling Reduction

Modeling ideas

- Modeling the broadcasting as a matrix.

```
typedef Achan {
        chan ch[n] = [1] of {bit};
}
Achan Mchan[n];
```

- Given process Ids $\{0, ..., n-1\}$, let $\{0, ..., t-1\}$ be the Ids of unreliable processes.

  Strong abtraction, why reasonable?

Modeling Reduction

Modeling ideas

- Modeling the broadcasting as a matrix.

```
typedef Achan {
        chan ch[n] = [1] of {bit};
}
Achan Mchan[n];
```

- Given process Ids $\{0, ..., n-1\}$, let $\{0, ..., t-1\}$ be the Ids of unreliable processes.
  Strong abtraction, why reasonable?
- Count only the received 1-values.

Presentation

Futher Reductions

More possibilities to diminish the size of state space

- avoid the transmission of a process to itself, this would save n channels.
- put some ordering on top of the transmission of values by processes and reuse channels, then a single channel would be sufficient.
- rather than broadcasting the majority value, just use a global variable that each process can read
- ...

A GSM-Based E-Commerce Protocol

Literature

Tim Kempster and Colin Stirling.Modeling and model checking mobile phone payment systems. *In Formal Techniques for Networked and Distributed Systems - FORTE 2003, volume 2767 of LNCS, page 95f. SpringerVerlag, 2003.*

The Model

- **one** customer, **one** merchant and **one** message center
- goods delivery via **multiple charging messages**
- **timeouts** in SMS transmission to mobile phone
- **unordered message reception**

Consider the following properties

- **goods atomicity**: "A merchant should receive payment iff the customer received the goods"
- **money atomicity**: "Money should be neither created nor destroyed in electronic commerce protocols"