442
516-61
319677
D-15

N91-17575

A HOL Theory for Voting

Paul S. Miner James L. Caldwell

Outline

- Introduction
- Proofs Comparing Majority and Plurality
- Proofs of Simple Reconfiguration Strategies
- Future directions

Introduction

- Central to fault-tolerant computing is redundancy mangement.
- Common to proofs of fault-tolerance is a maximum fault assumption.

  If there are m or fewer faults in the system, then ...

- Typically a maximum fault assumption is rather restrictive. Usually, this is necessary to avoid assumptions about the behavior of faulty channels.
  - For Interactive consistency, in order to tolerate m faults, $3m + 1$ nodes are required.
  - For a majority vote, $2m + 1$ channels are required.

  ⋮

- A maximum fault assumption is useful because it allows us to reason about fault tolerance in the presence of arbitrarily malicious fault behavior. However, analysis of the architecture may establish certain scenarios in which the assumption may be weakened.

- Should fault-tolerant systems incorporate features which attempt to recover from failure combinations which exceed the maximum fault assumption?
- If so, what is the proof obligation?

- At the very least, it is necessary to show that existing proofs which depend upon the maximum fault assumption still hold.

Hypothetical Scenario

Imagine that plurality voting circuit has been developed for use in a a four channel fault-tolerant computing system. Suppose that a designer is considering using this circuit in a system which depends upon a majority vote in order to maintain correct system state.

Can this voting circuit be used in this system?

First we define existence predicates for majority and plurality as follows:

$$\forall B.\mathit{majority\_exists}\ B \equiv \text{FINITE}\ B \wedge \exists x.|B| < 2|B|_x$$

$$\forall B.\mathit{plurality\_exists}\ B \equiv \exists x.\forall x'.(x \neq x') \supset |B|_{x'} < |B|_x$$

Where B is a bag[1], $|B|$ represents its cardinality, and $|B|_x$ represents the count of x in B.

[1] Essentially a bag is a set without absorption. $[a, a, b] = [b, a, a]$, but $[a, b] \neq [a, a, b]$

From these we define the following functions:

$$\forall B.majority\ B = \ \boldsymbol{\epsilon}\ x.|B| < 2|B|_x$$

$$\forall B.plurality\ B = \boldsymbol{\epsilon}\ x.\forall x'.(x \neq x') \supset |B|_{x'} < |B|_x$$

The property we need to prove is

$$\forall B.\mathit{majority\_exists}\ B \supset (\mathit{majority}\ B = \mathit{plurality}\ B).$$

The first step was to show that

$$\forall B.\mathit{majority\_exists}\ B \supset \mathit{plurality\_exists}\ B$$

For this, we needed to prove the following lemma:

$$\forall B.\text{FINITE}\ B \supset (\forall x\ y.(x \neq y) \supset |B|_y \leq (|B| - |B|_x))$$

From this lemma, coupled with rewriting the right conjunct of *majority_exists* to

$$\exists x.(|B| - |B|_x) < |B|_x,$$

and then using transitivity of '$<$' and '$\leq$' we can establish the existence of plurality from the existence of majority.

In order to show the equivalence between *majority* and *plurality* we needed to establish uniqueness from existence (i.e. if it exists then its unique). This allowed us to substitute in one side of the equation and then show that the chosen value satisfied the predicate embedded in the other.[2]

[2]Thanks to Brian Graham of the University of Calgary for submitting his methods of dealing with the HOL choice operator ('ε ' or '@') to the info-hol mailing list.

Once this was done we looked at proving some other simple facts about voting which may be useful in the analysis of fault-tolerant architectures. Specifically, we proved the preservation of majority for a few common reconfiguration schemes.

- **Graceful Degradation**
- **Perfect Spares**
- **Imperfect Spares**

Of course, we neglected one of the more difficult aspects of reconfiguration, namely that of correctly identifying the faulty channel. All that we have done is prove a little bit of common sense.

Graceful Degradation

The simplest reconfiguration strategy is graceful degradation. This consists of removing a faulty channel and continuing processing with one less channel of redundancy. The proof for this case showed that a majority is preserved if a non-majority element is removed from consideration.

First we show existence

$$\begin{aligned}\forall B.\forall x.\ & \mathit{majority\_exists}\ B \supset \\ & (x \in B) \supset \\ & (x \neq \mathit{majority}\ B) \supset \\ & \mathit{majority\_exists}\ (B - x)\end{aligned}$$

This essentially reduces to showing

$$|B| < 2|B|_{x'} \supset (|B| - 1) < 2|B|_{x'}.$$

From existence we get uniqueness so we can then show

$$\begin{aligned}\forall B.\forall x.\ & \mathit{majority\_exists}\ B \supset \\ & (x \in B) \supset \\ & (x \neq \mathit{majority}\ B) \supset \\ & (\mathit{majority}\ B = \mathit{majority}\ (B - x))\end{aligned}$$

Perfect Spares

Sometimes, in addition to removing a faulty channel, a good channel is added to the configuration. To capture this scenario, we showed that the insertion of the majority element to a bag preserved both existence and value of the majority.

$$\forall B.\ \mathit{majority\_exists}\ B \supset \mathit{majority\_exists}\ ((\mathit{majority}\ B) \odot B)$$

$$\forall B.\ \mathit{majority\_exists}\ B \supset (\mathit{majority}\ ((\mathit{majority}\ B) \odot B) = \mathit{majority}\ B)$$

Imperfect Spares

Finally, recognizing that it is possible for spares to fail, it was shown that the removal of a non-majority (e.g.failed) element coupled with the addition of an arbitrary element (of the proper type) also preserves both existence and the value of majority.

$$\forall B.\ \mathit{majority\_exists}\ B \supset$$
$$\quad \forall x\ x'.\ (x \in B) \supset$$
$$\quad\quad (x \neq \mathit{majority}\ B) \supset$$
$$\quad\quad \mathit{majority\_exists}\ (x' \odot (B - x))$$

$$\forall B.\ \mathit{majority\_exists}\ B \supset$$
$$\quad \forall x\ x'.\ (x \in B) \supset$$
$$\quad\quad (x \neq \mathit{majority}\ B) \supset$$
$$\quad\quad ((\mathit{majority}\ (x' \odot (B - x))) = (\mathit{majority}\ B))$$

Future Efforts

- Establish a base for reasoning about error manifestations in order to reason about Fault Detection and Isolation.

  When can you conclude that a redundant channel is faulty?

- Explore the effects that incorporating a plurality voter would have on the OS proofs.

  This would require adding assumptions concerning the behavior of faulty channels.

- Explore possible ways to incorporate reconfiguration strategies into the OS effort.

  How do you differentiate between a permanent and a transient fault?