Ideal Stabilization

Mikhail Nesterenko
Kent State University

Sébastien Tixeuil
UPMC Sorbonne Universités & IUF

AINA 2011, Singapore, 24 March 2011
Motivation
Distributed System
Legitimate State
Transient Faults
Recovery
Recovery
Legitimate State
Self-stabilization
Self-stabilization
Ideal Stabilization
Questions

• Existence ?
• Construction ?
• Composition ?
• Implementation vs. Specification ?
• Proof techniques ?
Model
Distributed System
Model

Input → Output

[Diagram of a complex network of connections]
Model

Input → 1 → 1 → 1
        ↓     ↓     ↓
       F     T     

Output ← 0 ← 0 ← 3 ← 4
Merge Symmetry
Merge Symmetry

A

B

---
Input Completeness
Input Completeness
Ideal Stabilization to Non-ideal Specifications
State Displacement
A Necessary Condition

- **Ideal stabilization** may be *possible only* if the specification contains an *input-complete* subset of sequences such that every disallowed specification state contains at least one process whose projection does not occur in the subset.
A Necessary Condition
Conflict Managers
Conflict Managers

Program

Mappings
Leader Election
Leader Election
Leader Election
Stabilization to Ideal Specifications
Ideal Specifications
Alternating Bit Protocol

next: receive ack(nm) →
   if nm = ns then
     ns := ¬ns
     send data(ns)

timeout: timeout() → send data(ns)

reply: receive data(nm) →
   if nm ≠ nr then
     nr := nm
     send ack(nm)
**Alternating Bit Protocol**

**next:** receive $ack(nm) \rightarrow$

  if $nm = ns$ then

    $ns := \neg ns$

    send $data(ns)$

**timeout:** $timeout() \rightarrow$ send $data(ns)$

**reply:** receive $data(nm) \rightarrow$

  if $nm \neq nr$ then

    $nr := nm$

    send $ack(nm)$
next: receive $\text{ack}(nm)$ $\rightarrow$
    if $nm = ns$ then
        $ns := \neg ns$
        send $\text{data}(ns)$
    timeout: $\text{timeout}() \rightarrow$ send $\text{data}(ns)$
reply: receive $\text{data}(nm)$ $\rightarrow$
    if $nm \neq nr$ then
        $nr := nm$
        send $\text{ack}(nm)$
Conclusion

Weaker Guarantees

- Weak stabilization
- Probabilistic stabilization
- Probable stabilization
- Pseudo stabilization
- \( k \)-stabilization

Self-stabilization

Complexity Guarantees

- Pseudo stabilization
- \( k \)-stabilization
- \( k \)-time-adaptive
- Snap-stabilization
- Time-adaptive

Stronger Guarantees

- Fault-tolerant stabilization
- Strong stabilization
- Strict stabilization
- Predicate-preserving stabilization
- Fault-containment

Ideal Stabilization?

Predicate-preserving stabilization

Strict stabilization

Fault-containment
Conclusion

Weaker Guarantees

Weak stabilization
Probabilistic stabilization
Probable stabilization
Self-stabilization

Pseudo stabilization

Stronger Guarantees

Fault-tolerant stabilization
Strong stabilization
Strict stabilization

Ideal Stabilization

Predicate-preserving stabilization

$k$-stabilization
$k$-time-adaptive
Snap-stabilization
Fault-containment

Complexity Guarantees

Predicate-preserving stabilizatio
Ideal Stabilization

• New way of **reasoning** about distributed fault-tolerance

• Abitrary degree of precision when **specifying** the system behavior after transient faults occur

• **Composition** is easy

• **Assertional** vs. operational proofs
Thank You