Number of rounds for Consensus

Non-Uniform Consensus

- (Non-Uniform) Agreement: No two **correct** processes decide on different values
- Validity: If all processes start with the same value $v \in V$, then v is the only possible decision value
- Termination: All correct processes eventually decide

(For simplicity and w.l.o.g., V={0,1})

The concept of valency

- Let C be a reachable state of a Consensus algorithm:
  - C is 0-valent (1-valent) if starting from C the only possible decision value of correct processes is 0 (1)
  - C is univalent if it is either 0-valent or 1-valent
  - Otherwise, C is bivalent

Intuition

- Valency is an external observer notion
- It captures the fact that an algorithm is committed to a certain decision value at certain point
- If no failures are possible then all executions are univalent

An example

- Consider the last week algorithm for n=3, t≤1. Let 0 be the default decision value
- Consider an initial state C_0=(0,1,1)
- What's the valency of C_0 if no failures are possible (t=0)?
- What's the valency of C_0 if t=1?

Lemma 1

- Let A be an algorithm that solves NUC and tolerates at most 1 failure. Then, A has a bivalent initial state

Assume that all initial states are univalent

By validity, if all processes start from 0 (1), then the decision value must be 0 (1)

Lemma 1 (cont)

$$\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 = 0 \\ 1 & 0 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 = 1 \end{array}$$

There exist two initial states C_0 and C_0' that differ in the input value of a single process *p* and have different valency

Lemma 1 (cont.)

Assume w.l.o.g. that all processes decide 0 in all executions starting from C_0 and *1* in all executions starting from C'_0

Let α (α') be an execution starting from C_0 (C'_0) where *p* fails before sending any msg

For all processes q≠p α is *indistinguishable* from α' ($\alpha \overset{q}{\approx} \alpha'$) ➔ all correct processes decide the same value in both α and α' ✘

#Rounds for N-U Consensus

- Synchronous system *S* with
  - *n* process
  - At most t≤n-2 stopping failures
  - At most 1 process fails at each round

Theorem 1: There does not exist an algorithm that solves NUC and decides in t rounds in S

By contradiction: Let A be such an algorithm

Lemma 2

- In any execution of A, the state reached after t-1 rounds is univalent

Proof:

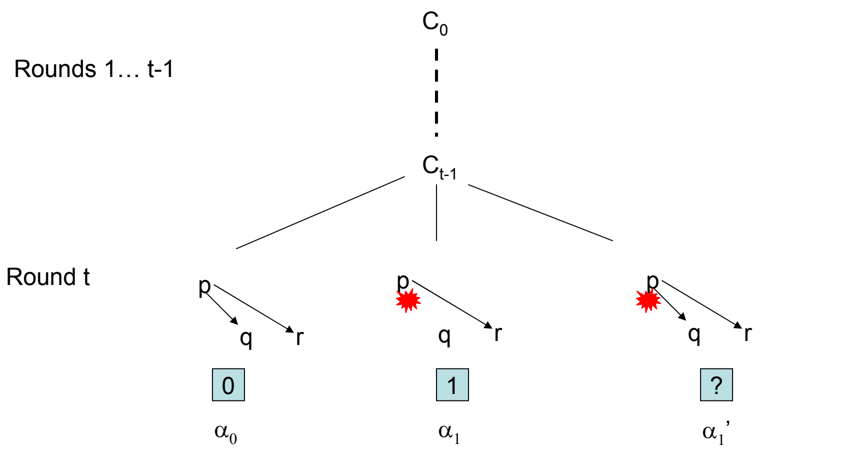
α_{t-1}: a t-1 round execution of A

C_0: the initial state of α_{t-1}

C_{t-1}: the state reached after α_{t-1}

C_{t-1} is bivalent (by contradiction)

Proof of Lemma 2



$$(1)\ \alpha_0 \overset{q}{\approx} \alpha_1' \Rightarrow q \text{ decides } 0 \text{ in } \alpha_1'; \quad (2)\ \alpha_1 \overset{r}{\approx} \alpha_1' \Rightarrow r \text{ decides } 1 \text{ in } \alpha_1'$$

Lemma 3

- There exists an execution α of A such that the state reached after t-1 rounds of α is bivalent

Proof: By induction:

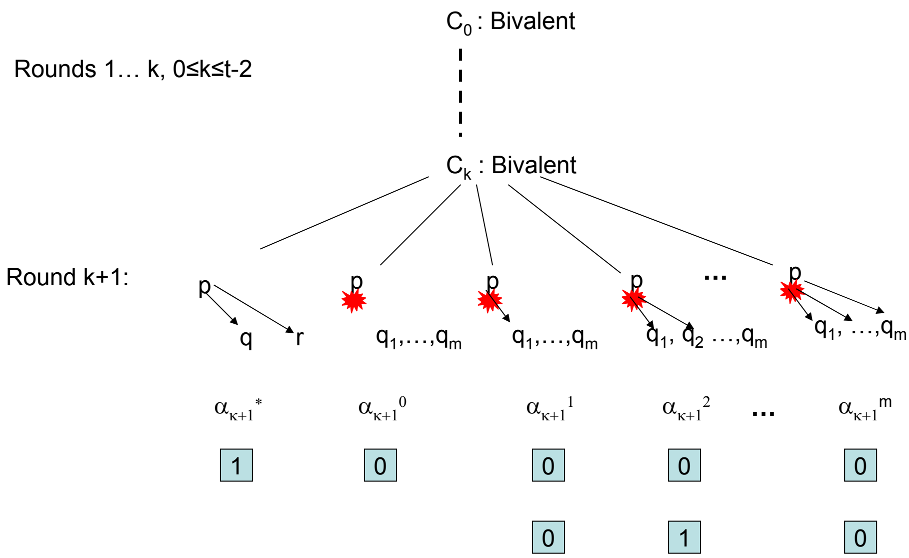
$\alpha_0 = C_0$: C_0 is the initial bivalent state of Lemma 1

α_k: k-round, $0 \le k \le t-2$, execution of A

C_k: the state reached after α_k

If C_k is bivalent, then can extend α_k into α_{k+1} such that C_{k+1} is bivalent

Proof of Lemma 3



Proof of Theorem 1

- By Lemma 2, in any execution of A, the state reached after t-1 rounds is univalent
- By Lemma 3, there exists an execution α of A such that the state reached after t-1 rounds of α is bivalent
- A contradiction

Number of rounds for Uniform Consensus

Uniform Consensus

- (Uniform) Agreement: No two processes decide on different values
- Validity: If all processes start with the same value $v \in V$, then v is the only possible decision value
- Termination: All correct processes eventually decide

(For simplicity and w.l.o.g., V={0,1})

The System Definition

- Synchronous system S with
  - n process
  - At most t, $1<t<n$, stopping failures
  - At most 1 process fails at each round
  - Messages sent by a faulty process are lost by prefix of processes: 1,…,l, where $1 \le l \le n$
- Let A be an algorithm that solves UC in S

#Rounds for Uniform Consensus

Theorem 1: For every f, $0 \le f \le t-2$, there exists an execution of A with f failures in which it takes at least f+2 rounds for all correct processes to decide

Actions and States

- Environment actions: (i,[k])
  - process i fails and messages to 1,…,k are lost
  - (0,[0]) nobody fails
- Each (global) state x of A is a vector of process states $[x_1,\ldots x_n]$ where x_i is the (local) state of process i

Executions (I)

- If x is a reachable state of A, then (i,[k]) is *applicable* to x if i is non-failed in x and t is not exceeded
  - (0,[0]) is always applicable
- The state of A after r rounds from an initial state x_0 is completely determined by $(i_1,[k_1]),\ldots,(i_r,[k_r])$, where $(i_j,[k_j])$ is an e.a. applicable in round j, $1 \leq j \leq r$

Executions (II)

- x is a reachable state of A and (i,[k]) is applicable to x,

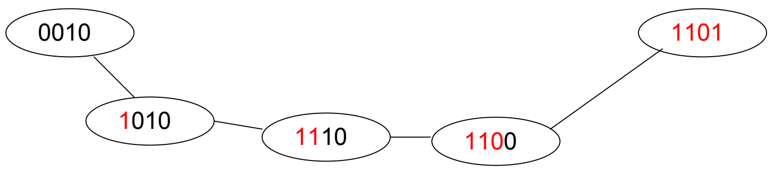
  x·(i,[k]) denotes the state reached after running A for one round from x with (i,[k])
- Execution: x· (i_1,[k_1]) ·…· (i_r,[k_r]) ·…

Similarity

- Let x, y be two states of A
- x and y are *similar*, x~y, if there exists at most one process j such that $x_j \neq y_j$, and at least one process i≠j is non-failed in both x and y
- A set X of states is *similarity connected* if the graph (X, ~) is connected

Lemma 1

- The set of initial states of A is similarity connected



Coloring

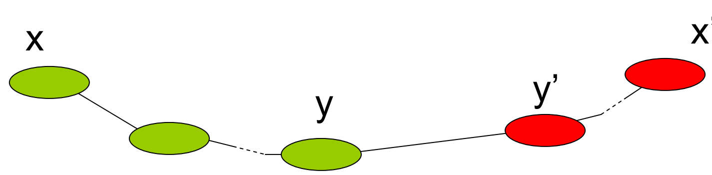
- Each state x is attributed a unique color (value) val(x):
  - If no failures are possible after state x, then x is univalent
  - val(x) is the value decided in a failure free extension of x

Lemma 2 (Uniformity Lemma)

- If
  - X is similarity connected
  - ∃ x,x'∈X such that val(x)=0 and val(x)=1
  - In all states in X exist at least 3 non-failed processes and 2 can still fail (≤t-2 failed)
- Then,
  - ∃ y∈X such that in y·(0,[0]) not all decided

1-round failure-free extension of y

Proof of Lemma 2



- y~y' and val(y)=0 and val(y')=1
- y and y' differ only in state of process j

Claim 2.1: either y or y' satisfy Lemma 2

Proof of Claim 2.1

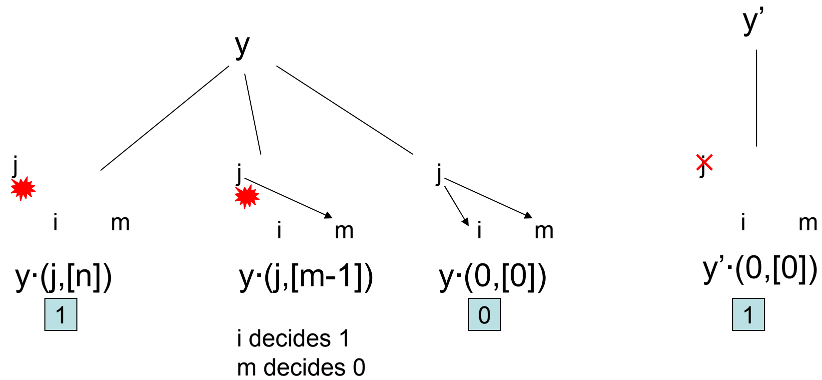
- Assume by contradiction:
  - All processes decide in both y·(0,[0]) and y'·(0,[0])
- Two cases:

(2.1.1) j is failed in either y or y'

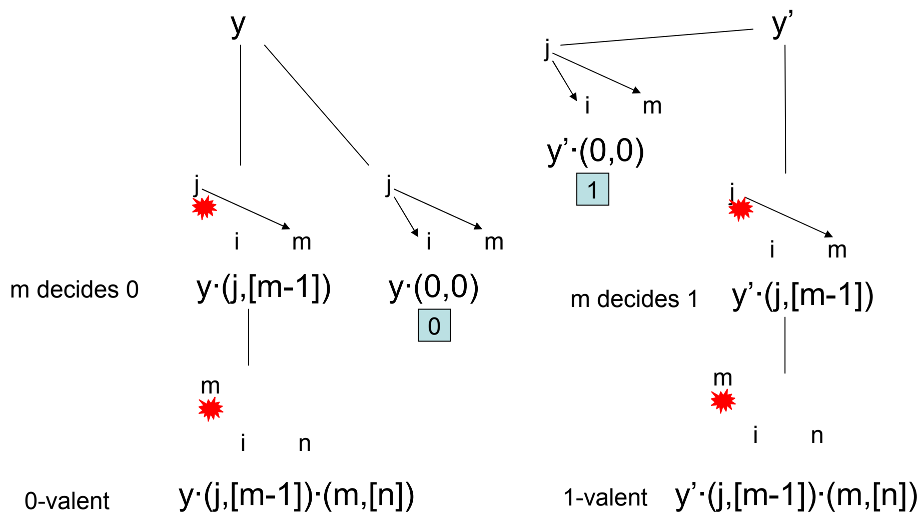
(2.1.2) j is non-failed in both y and y'

Proof of 2.1.1

Assume w.l.o.g. that j is failed in y':



Proof of Claim 2.1.2



no correct process see any difference

Corollary 1

- Theorem 1 holds for f=0

Proof:

(1) The set of initial state is similarity connected (Lemma 1)

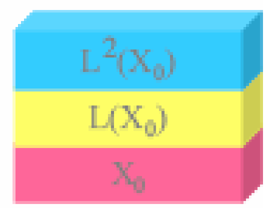
(2) val(0,…,0)=0 and val(1,…,1)=1 (Validity)

(3) n>t>1 ➔ n≥3➔initially 3 correct, 2 could still fail

By Uniformity Lemma, there exists an initial state y_0 such that some process has not yet decided in the 1-round failure-free extension of y_0

Layering

- $L(x)=\{x \cdot (i,[k]) : (i,[k])$ is applicable to $x\}$
- $L(X)=\cup_{x \in X} L(x)$
- $L^0(X)=X$; $L^k(X)=L(L^{k-1}(X))$, $k>0$
- Define system using layers
  - X_0 is the set of initial states
  - All executions are obtained from L(.)



Lemma 3 (Connectivity Lemma)

- If
  - X is a similarity connected set
  - No process is failed in X
- Then, for all k, $0 \le k \le t$:
  - $L^k(X)$ is a similarity connected set
  - no more than k processes are failed in $L^k(X)$

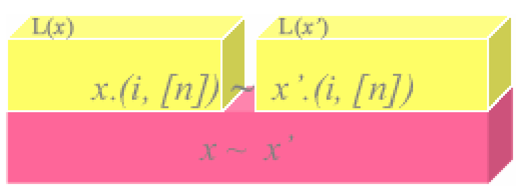
Proof of Lemma 3

- By induction on k
- k=0 is immediate ($L^0(X)=X$)
- Assumption: $L^{k-1}(X)$ is similarity connected and no more than $k-1<t$ processes are failed in $L^{k-1}(X)$
- Prove:

  (3.1) For all $x \in L^{k-1}(X)$, $L(x)$ is sim. con.

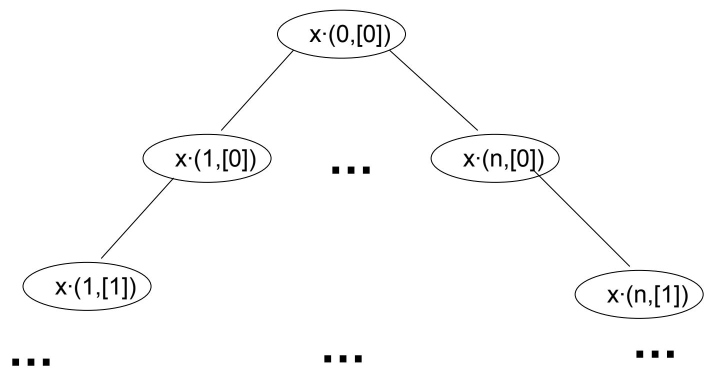
  (3.2) $x \sim x' \Rightarrow \exists y \in L(x), y' \in L(x'): y \sim y'$

Proof of Claim 3.2



- x and x' differ in the state of at most one process i
  - i non failed in both $\Rightarrow$ $x \cdot (i,[n]) \sim x' \cdot (i,[n])$
  - i failed in x (w.l.o.g.) $\Rightarrow$ $x \cdot (0,[0]) \sim x' \cdot (i,[n])$

Proof of Claim 3.1



Proof of Theorem 1

- Fix f, $0 \le f \le t-2$
- X_0 is sim. connected (Lemma 1) ➔ $L^f(X_0)$ is sim. connected (Lemma 3)
- $\exists x,x' \in X_0$ $val(x) \neq val(x')$ (Validity)
- $y = x \cdot (0,[0])_1 \cdot \ldots \cdot (0,[0])_k$
- $y' = x' \cdot (0,[0])_1 \cdot \ldots \cdot (0,[0])_k$
- $val(y) \neq val(y')$ and $y,y' \in L^f(X_0)$
- By Lemma 2: $\exists z \in L^f(X_0)$ s.t. in the failure free extension of z some process decides in at least 2 rounds

Remarks

- The connectivity lemma is a general result for the stopping failure model
- Feature of the model, not of a problem
  - Implies f+2 bound for UC
  - Implies f+1 bound for NUC (HW1)
  - See [Moses, Rajsbaum 98] for more results
- The f+2 bound cannot be obtained using bivalence alone (see paper)

UC Consensus Algorithms

- A simple modification of PS1.1 produces an early-deciding algorithm for UC for $1 \le t < n$ and $0 \le f \le t$ (HW2)
  - Two special cases when it is possible to do better: t=1 and f=t-1 (Charron-Bost, Schiper)
    - f+1 rounds
  - For f=t, we could obviously decide in f+1

Early Stopping

- Early stopping (i.e., halting in O(f) rounds) is harder than early deciding:
  - Requires min(t+1,f+2) rounds for NUC [Dolev, Reischuk and Strong 90]
- HW2: Modify NUC algorithm to satisfy early stopping
- HW2: Modify UC alg. to satisfy early stopping