Nota, on used below is described in Dijkstra’s paper on self-stabilization. When machine $i$ is performing the algorithm below, $L$ denotes the state of the machine $(i-1)$ modulo $(N+1)$, and $S$ is the state of machine $i$ itself. The machines are numbered $0$ through $N$ (thus, there are $N+1$ machines).

**Solution with K-state Machines ($K > N$)**

Here each machine state is represented by an integer value $S$, satisfying $0 \leq S < K$. For each machine, one privilege is defined, viz.

for the bottom machine:

\[
\text{if } L = S \text{ then } S := (S + 1) \mod K \text{ fi}
\]

for the other machines:

\[
\text{if } L \neq S \text{ then } S := L \text{ fi}
\]

---

**Paxos**

Phase 2. (a) If the proposer receives a response to its `prepare` requests (numbered $n$) from a majority of acceptors, then it sends an `accept` request to each of those acceptors for a proposal numbered $n$ with a value $v$, where $v$ is the value of the highest-numbered proposal among the responses, or is any value if the responses reported no proposals.

(b) If an acceptor receives an `accept` request for a proposal numbered $n$, it accepts the proposal unless it has already responded to a `prepare` request having a number greater than $n$.

---

**Cryptography**

- **Encoding (encryption)** of a message that can only be read (decryption) by a key.
- In **shared key cryptography** (symmetric cryptography) the sender and the recipient know the key, but no one else does.
  - E.g., DES (Data Encryption Standard) – 56 b key operates on 64 b blocks of data. Notation: $K_{des}(M)$.
  - How do Alice and Bob get the shared key $K_{AB}$ to begin with?
- In **public/private key pairs** messages are encrypted with a published public key, and can only be decrypted by a secret private decryption key.
  - E.g., RSA / PGP keys – at least 512 b long. Code for E & D is "open-source" (hence known to attacker).