11.3.2 Byzantinische Verständigung (Lamport-Shostak-Pease) mit beglaubigten Nachrichten.

S11.1 Satz: Lamport-Shostak-Pease, 1982

Der nachfolgende Algorithmus löst in jedem Fall das Verständigungsproblem auf der Basis beglaubigter Nachrichten.

Es sei V eine Menge von Werten. Weiter sei choice(V) eine Funktion mit folgenden Eigenschaften:

1. **Wenn** $V = \{v\}$ **ist, dann ist** $\mathrm{choice}(V) = v$.
2. $\mathrm{choice}(\emptyset) = 0$.

Im weiteren bezeichne $x{:}i$ **den Wert** x **beglaubigt von Prozessor i.**
Es bezeichnet also $v{:}j{:}i$ **den Wert** v **beglaubigt von Prozessor j und dann** $v{:}j$ **beglaubigt von Prozessor i.**

Dabei wird unterstellt, daß jeder Prozessor eine verlangte Nachricht entweder sendet oder dem Empfänger bekannt ist, daß sie nicht gesendet wird.

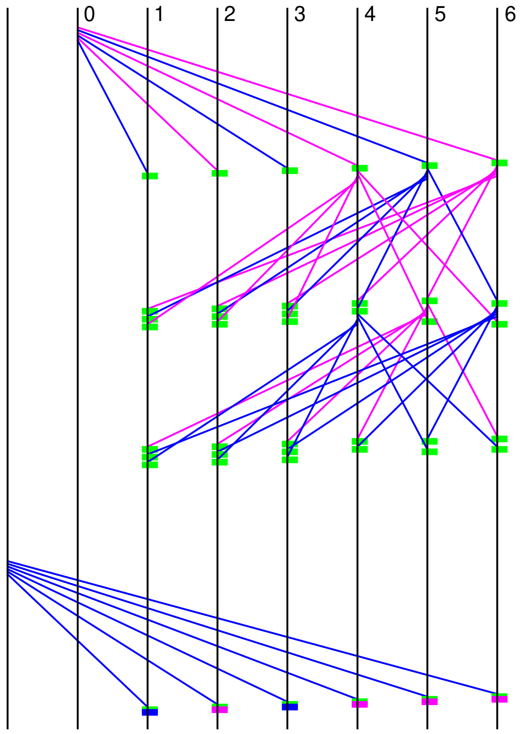
Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Algorithmus SM(m), $m \geq 0$

1. **Der Anführer sendet seinen von ihm selbst beglaubigten Anfangswert an alle anderen.**
2. **Für jedes i:**
   - a **Erhält ein Teilnehmer eine Nachricht der Form** $v{:}0$ **vom Anführer und hat bislang noch keine Nachricht erhalten, so führt er folgende Tätigkeiten durch:**
     - i **Er setzt** $V_i = \{v\}$.
     - ii **Er sendet die Nachricht** $v{:}0{:}i$ **an alle anderen Teilnehmer.**
   - b **Erhält ein Teilnehmer eine Nachricht der Form** $v{:}0{:}j_1{:}\ldots{:}j_k$ **und es ist** $v \notin V_i$**, dann führt er folgende Tätigkeiten durch:**
     - i **Er nimmt** v **in** V_i **auf.**
     - ii **Wenn** $k < m$ **ist, sendet er die Nachricht** $v{:}0{:}j_1{:}\ldots{:}j_k{:}i$ **an alle anderen Teilnehmer mit Ausnahme der Prozessoren** j_1 **bis** j_k**.**
3. **Für jedes i:**
   Wenn Prozessor i keine weiteren Nachrichten erhält, entscheidet er sich für den Wert $\mathrm{choice}(V_i)$.

Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann
Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

Beispiel 5: Lamport SM(5), Teilnehmer 0 bis 3 fehlerhaft (senden keine Nachrichten)



Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

Beweis

Zu erfüllende Bedingungen

IC1: Alle fehlerfreien Prozessoren ermitteln den gleichen Wert.

IC2: Wenn der Anführer fehlerfrei ist, ermitteln alle fehlerfreien Prozessoren seinen Wert.

Es wird zuerst IC2 betrachtet.

Wenn der Anführer fehlerfrei ist, erhält jeder Teilnehmer in Schritt 2 den Wert v des Anführers. Da kein Teilnehmer die Unterschrift des Anführers fälschen kann, nimmt kein fehlerfreier Prozessor jemals einen anderen Wert in V auf, so daß für alle fehlerfreien Prozessoren i gilt: $V_i = \{v\}$. Sie entscheiden sich also im Schritt 3 für v.

Da aus IC2 bei fehlerfreiem Anführer IC1 folgt, ist nur noch der Fall eines fehlerhaften Anführers zu betrachten.

Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann
Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

Es genügt zu zeigen, daß $V_i = V_j$ ist, wenn die Prozessoren i und j fehlerfrei sind.

Prozessor i nehme in Schritt 2 v in V auf.

Macht er dies in Schritt 2A, dann sendet er ihn auch an j.

Macht er dies in Schritt 2B, dann muß er eine Nachricht $v{:}0{:}j_1{:}\ldots{:}j_k$ erhalten haben.

Es sind zwei Unterfälle zu betrachten:

1. $k < m$**: In diesem Fall sendet er die Nachricht $v{:}0{:}j_1{:}\ldots{:}j_k{:}i$ an j, so daß auch Prozessor j diesen Wert erhält.**
2. $k = m$**: Anführer fehlerhaft ---> höchstens $m - 1$ der Prozessoren $j_1, \ldots, j_m$ fehlerhaft. Also muß wenigstens einer fehlerfrei sein und den Wert auch an Prozessor j gesendet haben.**

11.3.3 Byzantinische Verständigung (Dolev)

Reduktion der Nachrichtenzahl durch Erhöhung der maximalen Rundenzahl

Es sei $n = 3m + 1$:

Definitionen:

- **Runde**

  Algorithmus arbeitet in Runden

  $q_r = (q_r^1, \ldots, q_r^n)$ **Zustand am Anfang der r-ten Runde**

  q_r^i **enthält u. a. alle bis zu Beginn der r-ten Runde empfangenen Nachrichten**

- **Abarbeitung einer Runde in 2 Schritten:**

  1. **Aussenden von Nachrichten in Abhängigkeit vom Zustand am Anfang der Runde**
  2. **Empfang aller in der Runde gesandten Nachrichten und Ermittlung des Anfangszustandes für die nächste Runde**

Einigungsprotokolle: Algorithmus von Dolev

- **Nachrichtenarten**

  ***-Nachricht, die als Wert des Anführers 1 zusichert.**

  k-Nachricht, die mitteilt, daß k überzeugt ist, eine zu recht bestehende *-Nachricht erhalten zu haben.

- **Lokale Zustände** $q^j \in Q$ **mit**

  $Q = \wp((\{*\} \cup \{1, \ldots, n\}) \times \{1, \ldots, n\}) \times \mathbb{N}$

  Paare (Nachrichtenkennung, Sender), Runde

- $LOW = m + 1$ **und** $HIGH = 2m + 1$

  Wenn eine bestimmte Aussage von wenigstens LOW Prozessen erhalten wurde, so muß sie von wenigstens einem korrekten gemacht worden sein.

  Wenn eine bestimmte Aussage von wenigstens HIGH Prozessen erhalten wurde, so muß sie von wenigstens LOW korrekten Prozessen gemacht worden sein.

Einigungsprotokolle: Algorithmus von Dolev

- $W_x^j(q) = \{i \mid i \in \{1, \ldots, n\} \wedge (x, i) \in \pi_1(q^j)\}$

  Menge aller Prozessoren, die an j eine x-Nachricht gesandt haben $(x \in \{*, 1, \ldots, n\})$.

- ***j unterstützt i direkt*, wenn j von i eine *-Nachricht erhielt (d. h.** $i \in W_*^j(q)$**).**

- ***j unterstützt i indirekt*, wenn** $|W_i^j(q)| \geq LOW$ **ist, d. h. j weiß infolge der Mitteilung wenigstens eines korrekten Prozessors, daß i eine *-Nachricht erhalten hat.**

- **j *ist überzeugt*, daß i eine *-Nachricht ausgesandt hat, wenn** $|W_i^j| \geq HIGH$ **ist.**

- $C^j(q) = \{k \mid k \in \{1, \ldots, n\} \wedge k \neq s \wedge |W_k^j(q)| \geq HIGH\}$ **ist die Menge der *von j bestätigten Prozessoren* (ohne den Anführer s)**

BP 1 **Einigungsprotokolle: Algorithmus von Dolev**

- **Ein Prozessor j ist im Zustand q_r *aktiviert*, d. h. er sendet in der r-ten Runde an alle einschließlich sich selbst eine *-Nachricht,**

  wenn er schon in einer früheren Runde aktiviert war,
  d. h. $j \in W_*^j(q_r)$ **ist,** **(I1)**

  oder wenn $|C^j(q_r)| \geq LOW + \max(0, \lceil r/2 \rceil - 2)$ **ist** **(I2)**

  oder wenn $r = 2$ **ist und er in der ersten Runde vom Anführer eine *-Nachricht erhielt, d. h.** $s \in W_*^j(q_2)$ **ist.** **(I3)**

Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann

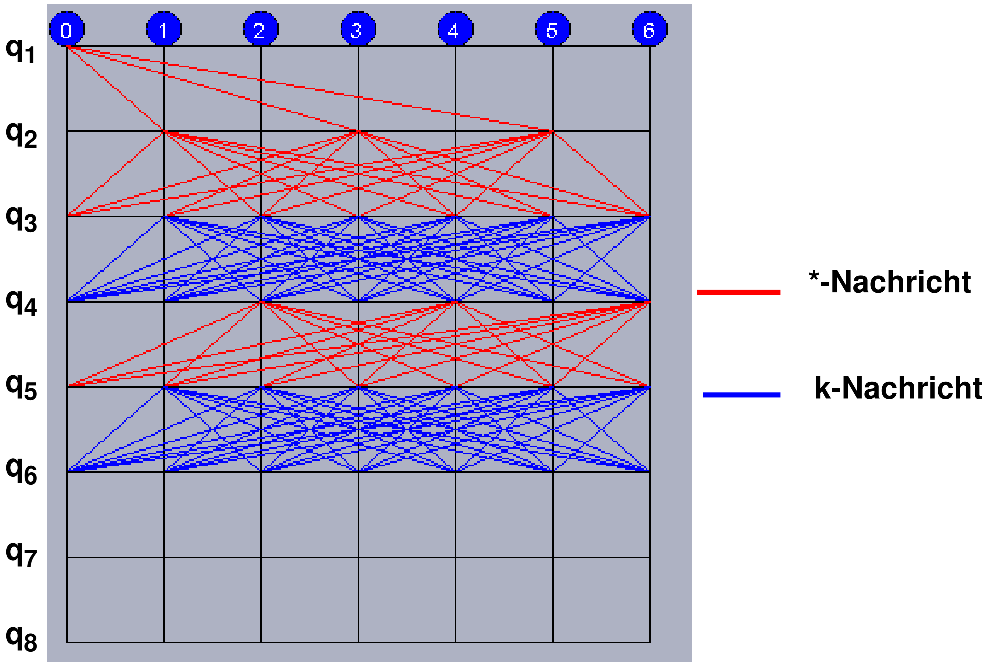
BP 1 **Einigungsprotokolle: Algorithmus von Dolev**

❑ **Algorithmus**

1. **In der ersten Runde sendet ein korrekter Anführer eine *-Nachricht an alle einschließlich sich selbst, falls sein Wert 1 ist.**
2. **In der k-ten Runde (k > 1) senden die Prozessoren an alle anderen (einschließlich sich selbst)**
   - **eine *-Nachricht, wenn sie aktiviert sind,** **(M1)**
   - **eine j-Nachricht, wenn sie j direkt unterstützen,** **(M2)**
     (d. h. $j \in W_*^i(q)$**)**
   - **eine j-Nachricht, wenn sie j indirekt unterstützen.** **(M3)**
     (d. h. $|W_j^i(q)| \geq LOW$**)**
3. **Prozessoren, die von wenigstens HIGH Prozessoren überzeugt sind, d. h.** $|\{i \mid |W_i^j| \geq HIGH\}| \geq HIGH$**, entscheiden sich für den Wert 1.**
4. **Wenn ein Prozessor sich nach der** $(2m+3)$ **-ten Runde für 1 entschieden hat, so nimmt er den Wert 1 als Ergebnis, sonst den Wert 0.**

Universität Erlangen-Nürnberg, IMMD IV, F. Hofmann
Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg ist ohne Genehmigung des Autors unzulässig

Beispiel 6: Fehlerhafter Anführer ist P_0,
Teilnehmer P_2, P_4 und P_6 erhalten anfänglich keine Nachricht



- Veranschaulichung

$i \rightarrow j \dashrightarrow k$: P_i hat an P_j eine *-Nachricht gesandt und P_j hat dies an P_k weitergemeldet

$k \in W_j^i$: $j \rightarrow k \dashrightarrow i$

$k \in W_*^i$: $k \rightarrow i$

$k \in C^i$: Es gibt mindestens m+1 korrekte vom Anführer verschiedene Prozessoren j_v mit $k \rightarrow j_v \dashrightarrow i$.
Prozessor k wird von i bestätigt.

i zeugt für j: $j \rightarrow i$

i zeugt indirekt für j: Für mindestens einen korrekten Prozessor k gilt $j \rightarrow k \dashrightarrow i$