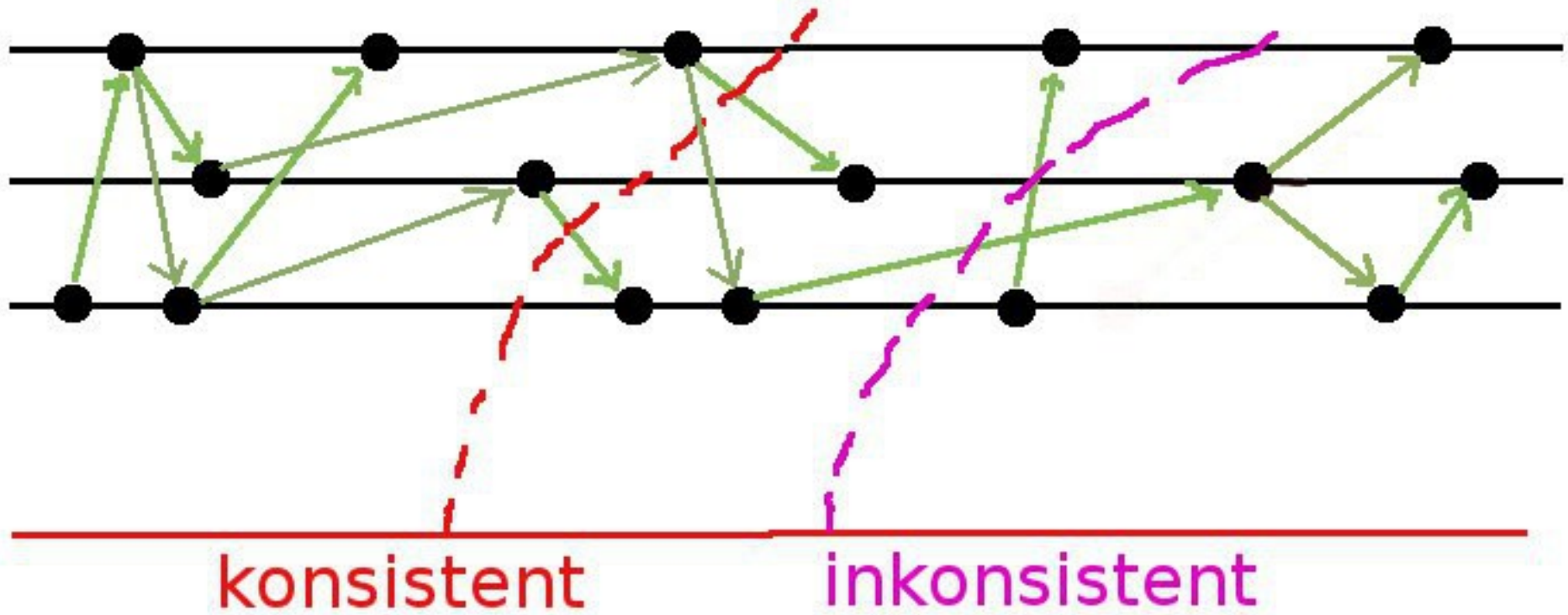


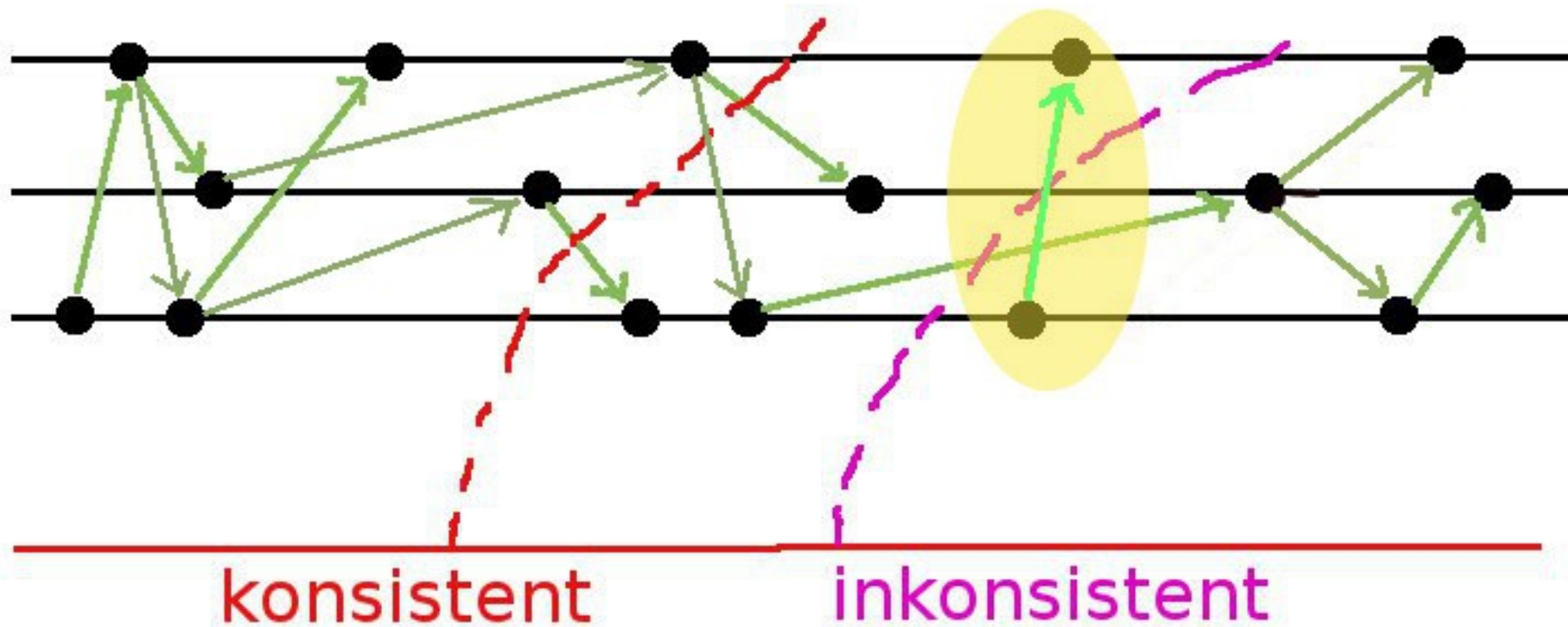
Konsistenter Schnappschuss

- Ziel: verteilten Zustand ausrechnen
- Beispiel:
 - Summe aller Kontostände
 - während Überweisungen getätigt werden
 - ohne das System einzufrieren

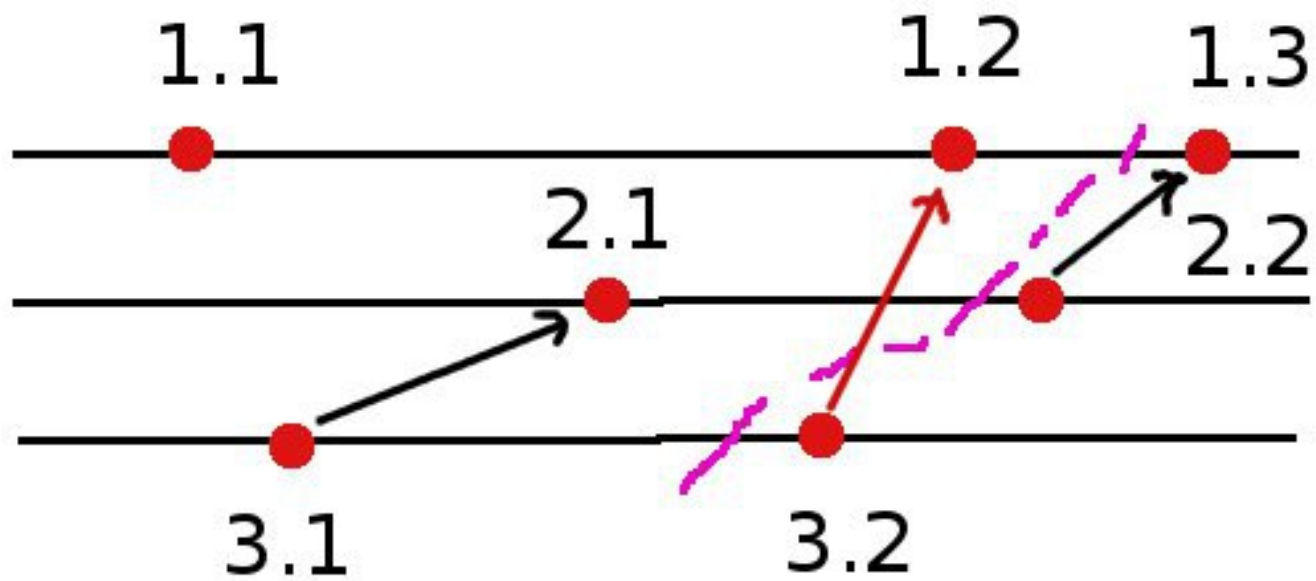
Konsistenz?



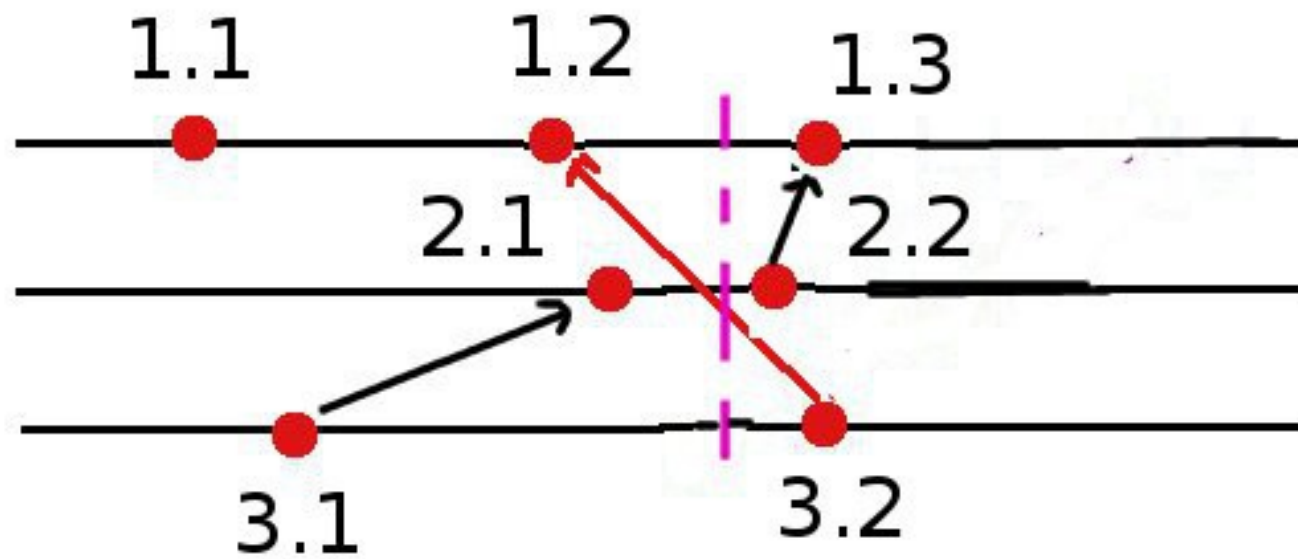
Konsistenz?



Gummiband-Transformationen



...zeigen inkonsistente Schnitte

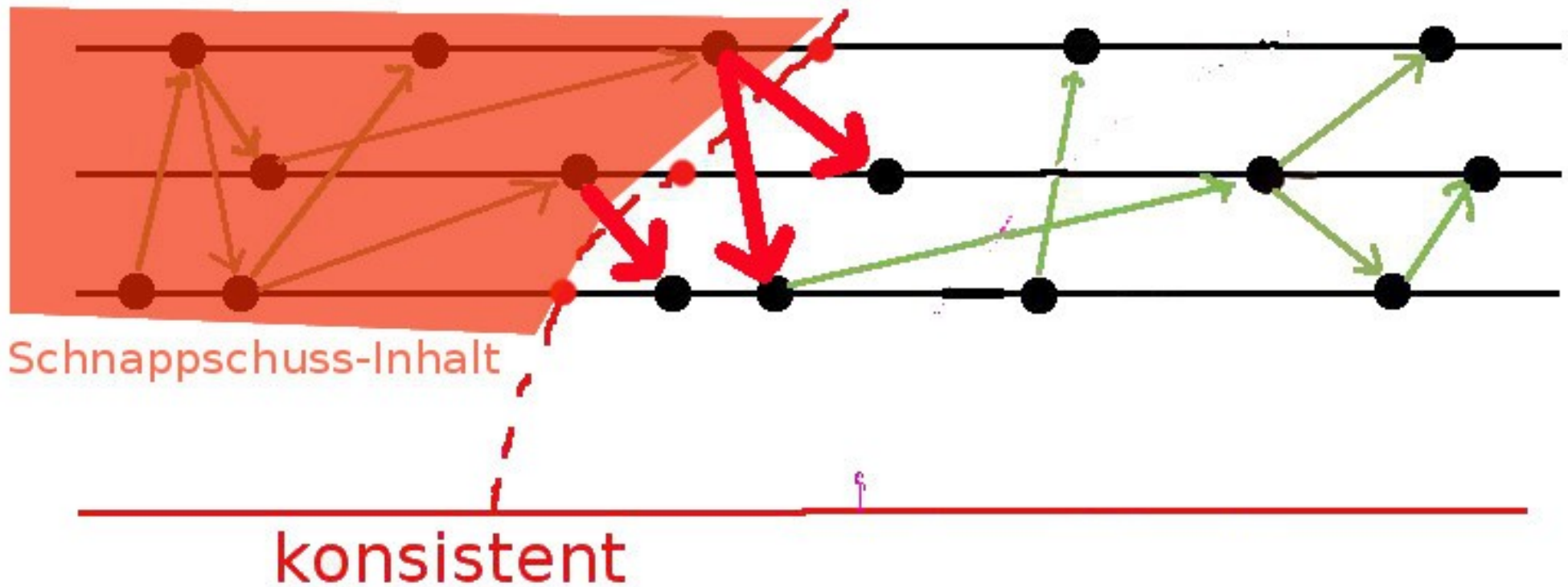


...falls Schnitt senkrecht

Konsistenz und kausale Abhängigkeit

- Für jedes Ereignis e im Schnappschuss sind
 - die kausalen Vorgänger $e' \leq e$ **UND**
 - die von e aus gesendeten Nachrichtenim Schnappschuss enthalten

Konsistenz und kausale Abhängigkeit



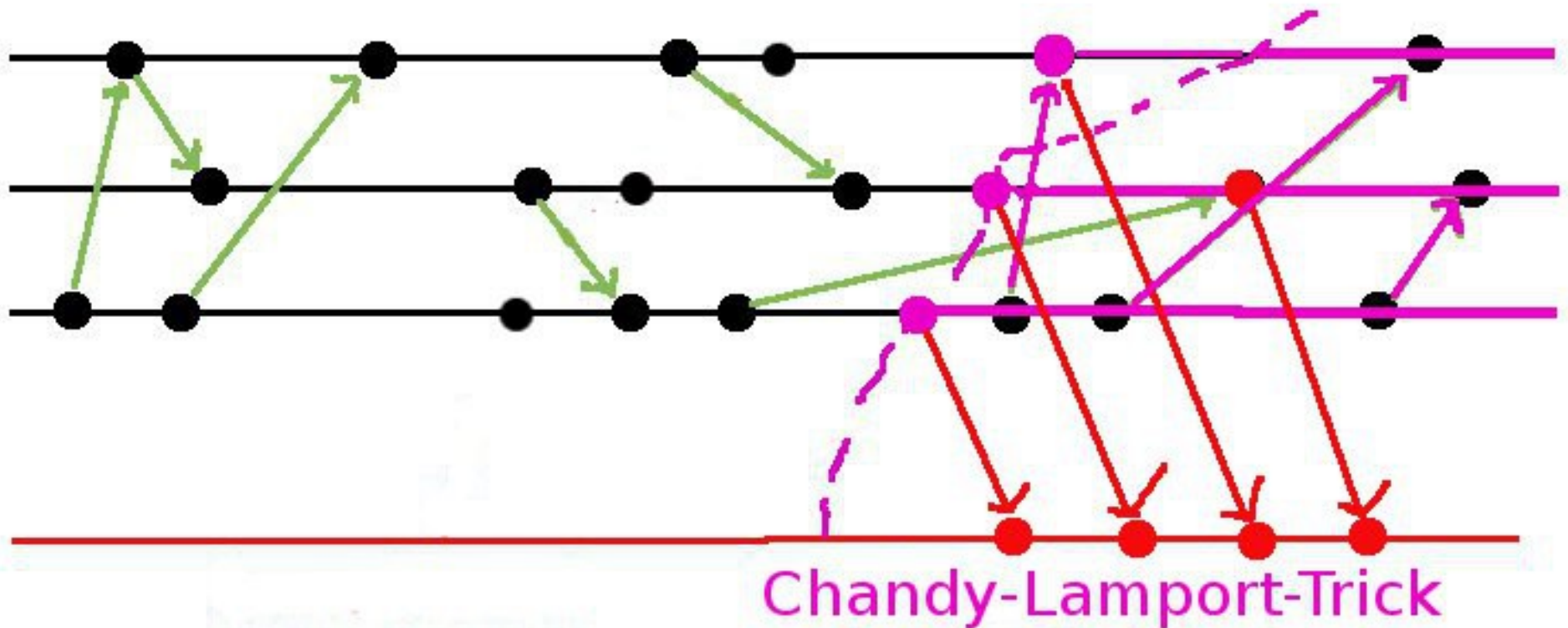
Chandy-Lamport

Markiere Prozess

Markierter Prozess sendet markierte Nachrichten

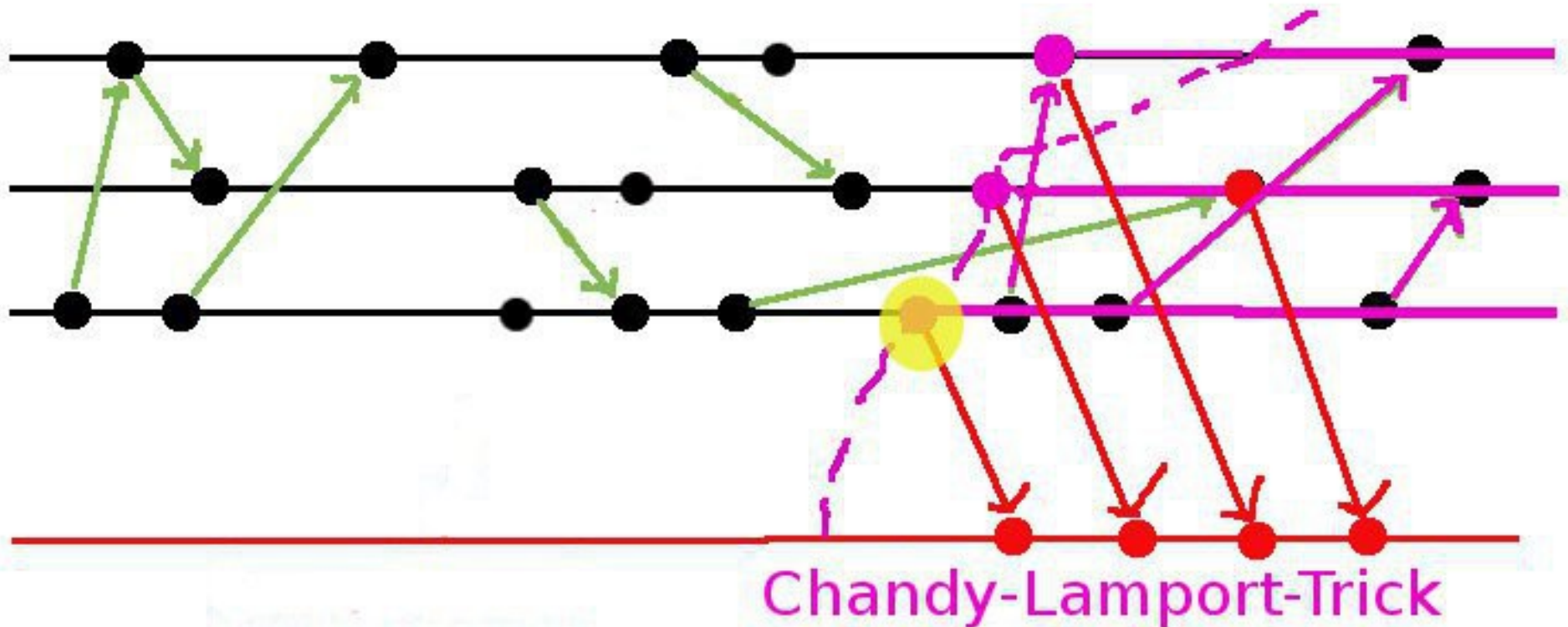
Prozess	Empf. Nachricht	Aktion
unmarkiert	unmarkiert	-
unmarkiert	markiert	Schnappschuss
markiert	unmarkiert	Nachricht -> Schnappschuss
markiert	markiert	-

Chandy-Lamport in Aktion



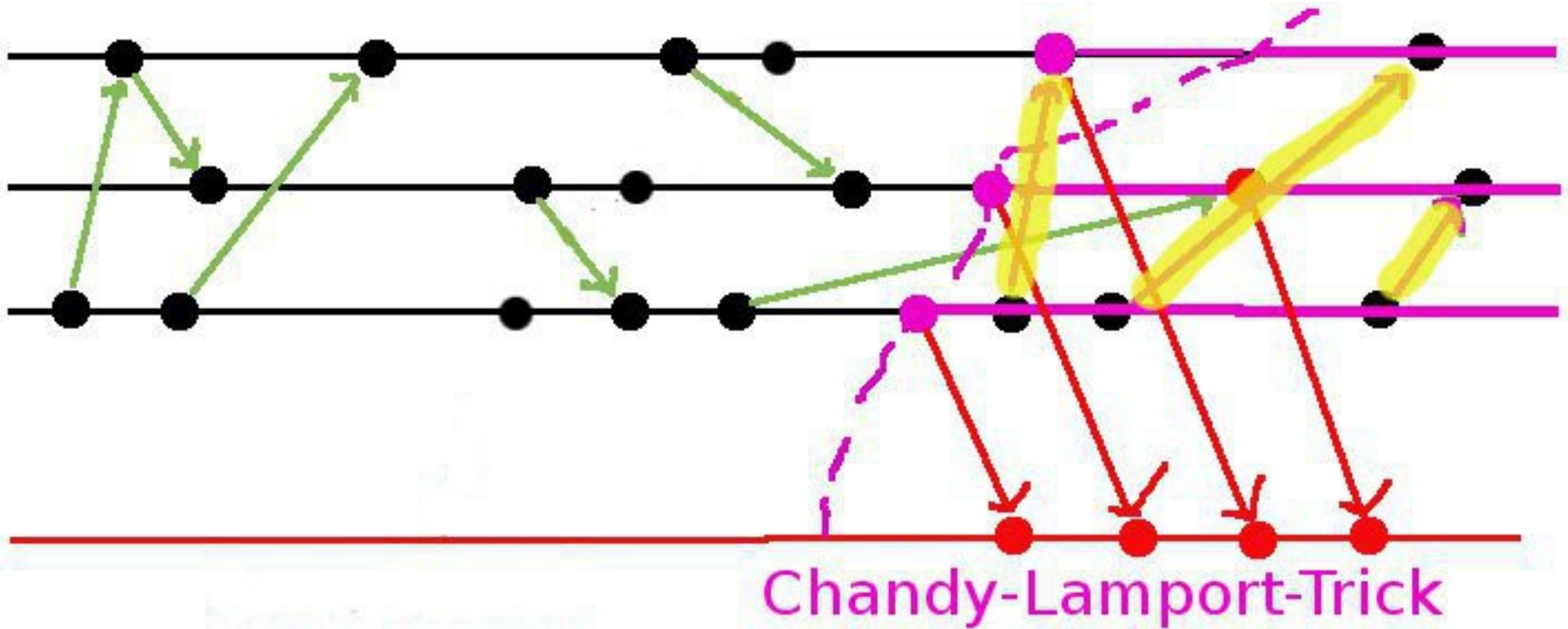
Chandy-Lamport (1)

Zustand an Beobachter



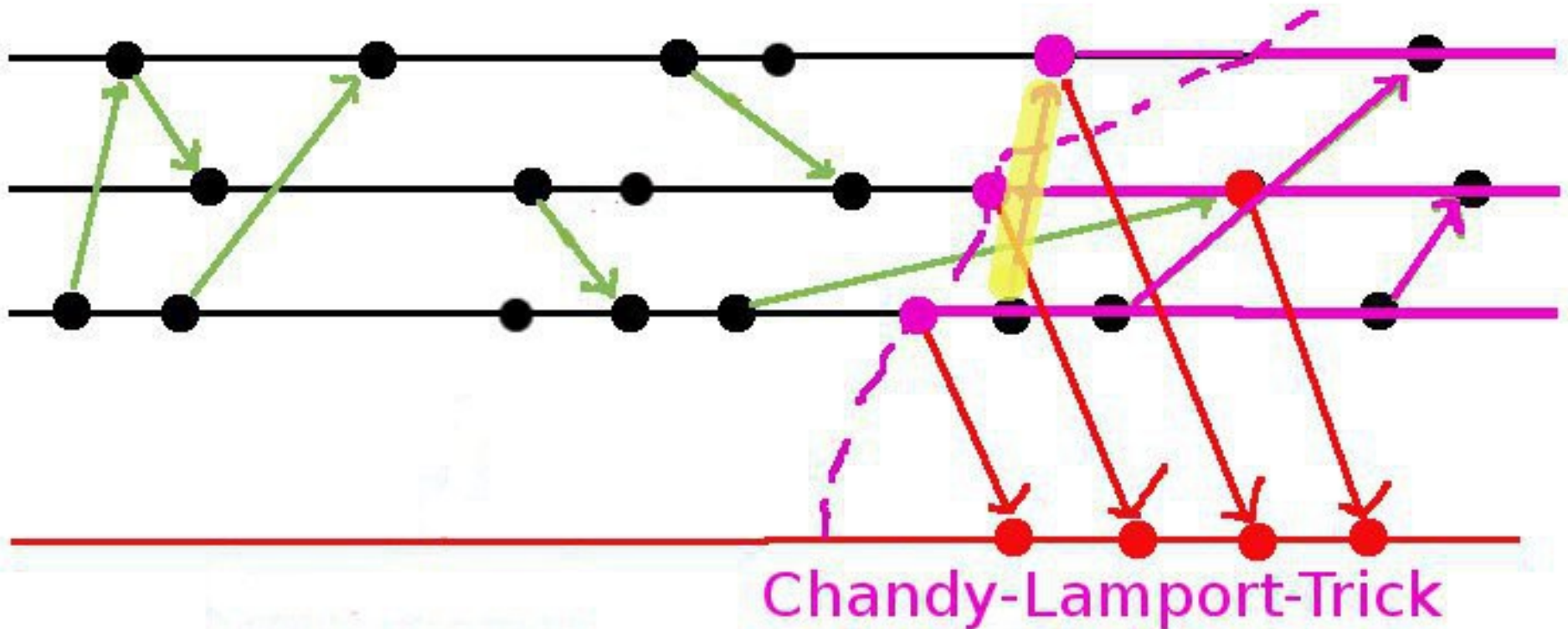
Chandy-Lamport (2)

Schnappschuss: markiere Nachrichten



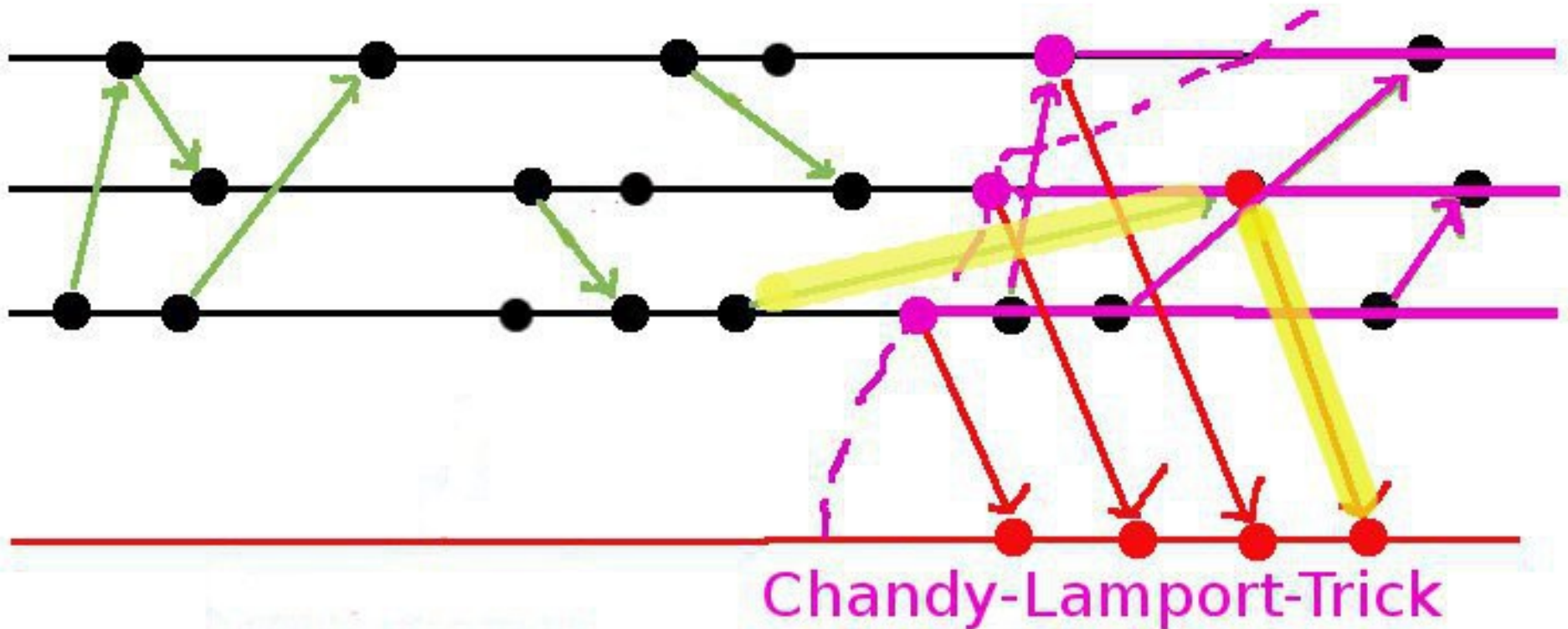
Chandy-Lamport (3)

Unmarkierter Prozess empfängt markierte Nachricht



Chandy-Lamport (4)

Nachrichten aus der Vergangenheit




Chandy-Lamport Algorithmus

1. The observer process (the process taking a snapshot):
 1. Saves its own local state
 2. Sends a snapshot request message bearing a snapshot token to all other processes
2. A process receiving the snapshot token *for the first time on any message*:
 1. Sends the observer process its own saved state
 2. Attaches the snapshot token to all subsequent messages (to help propagate the snapshot token)
3. Should a process that has already received the snapshot token receive a message that does not bear the snapshot token, this process will forward that message to the observer process. This message was obviously sent before the snapshot “cut off” (as it does not bear a snapshot token and thus must have come from before the snapshot token was sent out) and needs to be included in the snapshot.

Spezialfall: Terminierung

Chandy-Lamport Algorithmus

- Zustand enthält (S,R)
 - Anzahl gesendeter Nachrichten S
 - Anzahl empfangener Nachrichten R
- Beobachterprozess : (S-R) für alle summieren
 Anzahl fehlender Nachrichten
- Warte bis alle fehlenden Nachrichten eingetroffen

Mattern: kann man Vektorzeit benutzen?

- ...?
- Idee 1993

Mattern: Vektorzeit-Lösung

- Beobachter wählt (Vektor-)Zeitstempel s in Zukunft
- alle Knoten
 - bestätigen: lokale Zeit kleiner
 - merken: falls Zeit $> s$, dann Schnappschuss
- Beobachter setzt lokale Vektorzeit auf s
- Beobachter: Dummy-Nachricht an alle
- führt zu Schnappschuss bei allen
- Beobachter sammelt Nachrichten mit Zeit $< s$

Problem bei Vektorzeit-Schnappschuss

- Problem: Terminierung
 - Nachrichten mit kleinerer Vektorzeit unterwegs?
 - Nachrichtenzähler nötig