

Proseminar im Softwareprojekt 2002/2003
Carl-Von-Ossietzky-Universität, Oldenburg

Zeit und Raum in Rechnernetzen

Thema Nr. 56

Tim Dederling, Klaus Krogmann

1.0 Inhaltsverzeichnis

Kapitel	Thema	Seite
1.0	Inhaltsverzeichnis	2
2.0	Einführung	3
2.1	Synchrone Systeme.....	3
2.2	Problemstellungen an anschaulichen Beispielen.....	3
2.2.1	Firing-Squad-Problem.....	4
2.2.2	Das Problem der untreuen Ehemänner	5
2.2.3	Analyse: Theorem.....	6
2.2.4	Analyse des globalen Wissens	9
2.2.5	Verzögerte Kommunikation im Fall der untreuen Ehemänner	9
2.2.6	Das Problem der untreuen Ehemänner unter Henrietta III	11
2.2.7	Henrietta IV	12
2.3	Beschränkt asynchrone Systeme	13
2.3.1	Das Problem der Byzantinischen Generäle (Byzantinisches Agreement).....	13
2.3.2	Mögliche Fehlerarten beschränkt asynchroner Systeme.....	15
2.4	Vollkommen asynchrone Systeme	16
2.5	Fazit	16
3.0	Index	18
4.0	Abbildungsverzeichnis.....	19
5.0	Quellenverzeichnis	20

2.0 Einführung

Nicht zuletzt die allgegenwärtige Präsenz des Internets erfordert die Überwindung von Zeit und Raum in Rechnernetzen. Grundsätzlich sind alle verteilten Rechnersysteme zu einer großen Klassen zusammenzufassen, die eine Reihe von Vorteilen bietet:

- lokale Verfügbarkeit (also die Möglichkeit der Datenhaltung vor Ort und dem Zugriff auf CPU-Zeit)
- dezentrales Management (über ein Rechnernetz zusammengefasste Rechner lassen sich von einer Stelle aus administrieren)
- hohe Verfügbarkeit durch mögliche Redundanz (der Ausfall einzelner Komponenten kann durch andere Komponenten im gleichen Netz mit ähnlichen Fähigkeiten kompensiert werden)

Auf der anderen Seite wirft ein solches Rechnernetz neuartige Probleme auf, die sich am einfachsten mit anschaulichen Beispielen erklären und verstehen lassen.

Daher soll das nächste Kapitel (2.1) zunächst einmal ausgewählte Problemstellungen veranschaulichen.

2.1 Synchroner Systeme

Synchrone Systeme bieten einige Vorteile wie z. B. die Informationsübertragung ohne einen direkten Datenaustausch. Da alle Komponenten solcher Systeme synchronisiert sind, kann Information auch über Timeouts „transportiert“ werden:

- über ein bestimmtes Ereignis E ist bekannt, dass es in einer bestimmten Zeitspanne T eintreten wird (alle Teilnehmer/Clients des synchronisierten System erfahren davon, wenn es eintritt)
- wird das Ereignis E nicht innerhalb dieser Zeit beobachtet können alle Komponenten (unter Voraussetzung von globalem Wissen) folgern, dass es E nicht eingetreten ist.

Siehe hierzu Kapitel 2.2.2.

2.2 Problemstellungen an anschaulichen Beispielen

Die hier verwendeten Beispiele orientieren sich zu einem großen Teil an Problemstellungen aus dem militärischen Bereich, denn hier gibt es strenge Hierarchien und Regeln, die sich auf den Computerbereich übertragen lassen.

2.2.1 Firing-Squad-Problem

Ein General führt eine Armee von Soldaten. Erhält dieser einen Angriffsbefehl, so soll er seine Soldaten derart instruieren, dass alle gleichzeitig feuern.¹

Dieses Problem lässt sich in mehrere mögliche Problemstufen unterteilen. Im einfachsten Fall (siehe **Abbildung 2.2.1-1**, Seite 4) ist eine direkte Kommunikation zwischen General und seinen Soldaten möglich (etwa über eine Signalthorn). Sind alle Soldaten instruiert auf dieses Signal unverzüglich mit dem Angriff zu beginnen, treten keine Probleme auf.

Anders gelagert ist der Fall, wenn die Überbringung der Nachricht über eine größere Entfernung geschehen muss, die Soldaten also untereinander räumlich getrennt sind. Es ist also vonnöten, dass die Informationen über einen Angriff lokal weitergeleitet wird. Zunächst sei zur Vereinfachung angenommen, dass die Übertragung zuverlässig funktioniert, es also keine Veränderungen oder bewussten Manipulationen gebe.

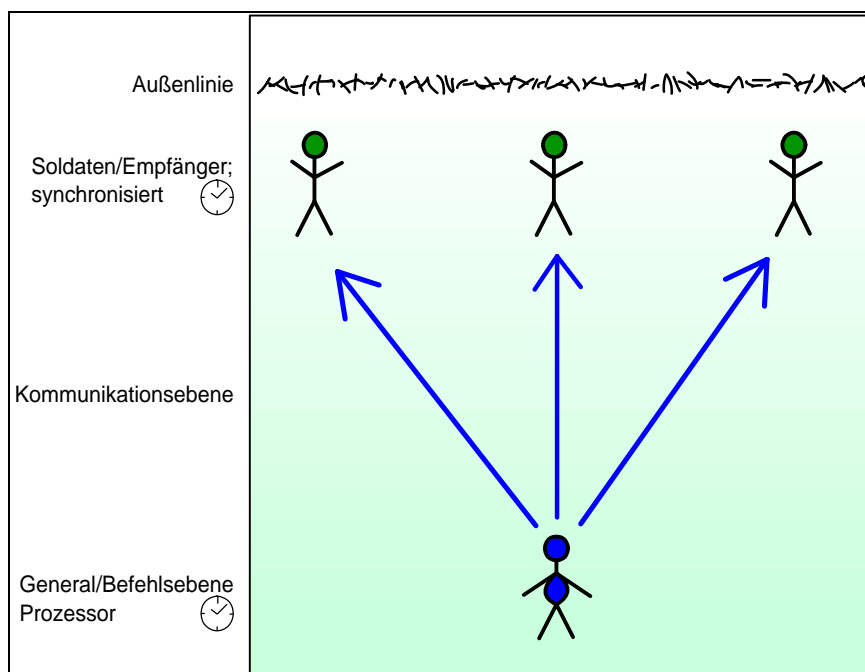


Abbildung 2.2.1-1 Firing-Squad-Problem, alle Soldaten sind direkt erreichbar

Weiterhin wird angenommen, dass alle Soldaten und der General über exakt synchronisierte Uhren verfügen (siehe **Abbildung 2.2.1-2**, Seite 5). Aber selbst in diesem Fall muss der General noch wissen, wann sein Befehl die Soldaten erreicht. Sollte keine maximale Übertragungszeit (für die Nachricht) feststehen, könnte ein Fall eintreten, bei dem ein Befehl

¹ siehe [1], Seite 157

um 1 Uhr ausgegeben wird, dass um 2 Uhr angegriffen werden soll. Aber es ist nicht sichergestellt, dass nicht vielleicht ein Teil der Truppen den Befehl erst nach 2 Uhr erreicht. Neben diesem Synchronisationsproblem taucht ein weiterer Sachverhalt auf. Bisher wurde davon ausgegangen, dass alle Soldaten gleich reagieren und genau wissen, was sie zu tun haben. Globales Wissen war also Voraussetzung. In Praxis muss es jedoch nicht zwingend klar sein, dass erst auf einen bestimmten Befehl hin gefeuert werden muss und keinesfalls vorher.

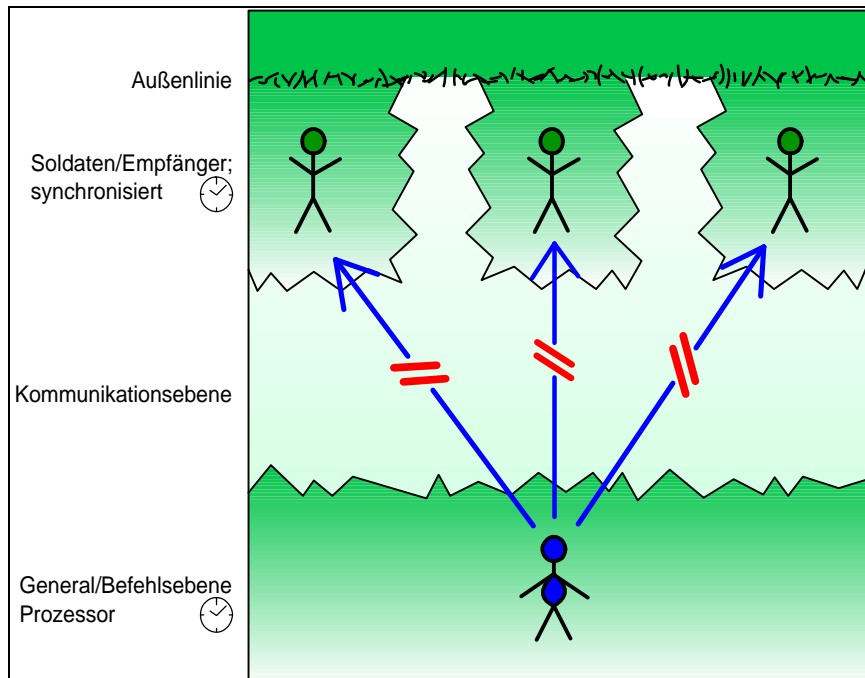


Abbildung 2.2.1-2 Firing-Squad-Problem mit zeitlicher Synchronisation, räumliche Trennung

Im Computer-Bereich wurde das Firing-Squad-Problem bisher hauptsächlich für reguläre, synchron getaktete ein- und mehrdimensionale Prozessorfelder untersucht. Es wird über ein einfaches Automatenmodell gelöst, in dem alle Komponenten gleichförmig getaktet sind. Durch die Taktung erfolgt eine rundenbasierte Nachrichtenverteilung, in der die Länge einer Runde so bemessen ist, dass diese Nachricht in jedem Fall alle Empfänger vor Ende der Runde erreicht. Somit entsteht eine Zerlegung von Berechnungsoperationen in Runden, die Synchronisation und globales Wissen sind gelöst.

2.2.2 Das Problem der untreuen Ehemänner

Der Inselstaat Mamajorca besitzt eine matriacharische Gesellschaftsform. Königin Henrietta I regiert mit großer Weisheit. Alle Untertaninnen gehorchen ihr aufs Wort und beherrschen das logische Schließen perfekt. Darüber hinaus ist noch wichtig zu wissen, dass die Insel so

klein ist, dass ein lautes Ereignis, das sich dort ereignet, überall wahrgenommen werden kann.

Eines Tages hält die Königin eine Versammlung aller Frauen auf dem Marktplatz ab und gibt Folgendes bekannt:

- *Mindestens einer Eurer Ehemänner ist untreu gewesen.*
- *Keine von Euch weiß, ob ihr eigener Gatte treu ist oder nicht, aber jede weiß dies von allen anderen Männern.*
- *Ich gestatte keine öffentliche Diskussion über dieses Thema.*
- *Sobald eine Frau weiß, dass ihr Gatte untreu ist, so hat sie diesen um Mitternacht desselben Tages zu erschießen.*

*Daraufhin vergehen neun Tage ohne Besonderheiten. In der zehnten Nacht konnte man eine Reihe von Schüssen hören.² (siehe **Abbildung 2.2.2-1**)*

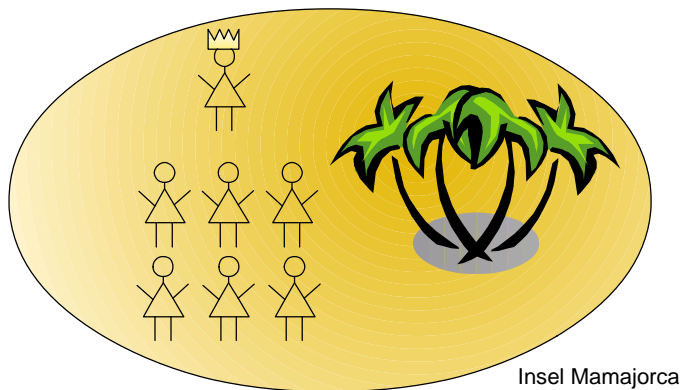


Abbildung 2.2.2-1 Henrietta I verkündet das Problem der untreuen Ehemänner

Betrachtet man diese Bedingungen, so erkennt man, dass das Verbot der öffentlichen Diskussion die Möglichkeit der Informationsübertragung ohne direkten Datenausch ermöglicht (siehe Kapitel 2.2.1). Die Versammlung auf dem Marktplatz entspricht einer Synchronisation, bei dem ein globaler Wissenstand bei allen Frauen erreicht wird.

2.2.3 Analyse: Theorem

Es sei t die exakte Zahl der untreuen Ehemänner, dann werden genau diese in der t -ten Nacht nach der Versammlung erschossen.³

Für den Fall $t = 1$ (siehe auch **Abbildung 2.2.3-1**, Seite 8) ergibt sich der folgende Ablauf: Alle Frauen, bis auf eine wissen von genau einem Mann, dass er untreu ist. Die Frau des

² siehe [1] Seite 159

³ siehe [1] Seite 159

untreuen Ehemannes hingegen kennt nur treue Ehemänner. Jetzt kann diese Ehefrau aus der Aussage der Königin, dass es mindestens einen untreuen Ehemann gibt, schließen, dass ihr eigener Mann der einzige untreue ist. Folglich wird sie ihren Mann (dem Befehl der Königin folgend) in der ersten Nacht erschießen.

Für diesen Fall tritt nun für alle anderen Frauen Gewissheit ein, dass ihr eigener Ehemann treu gewesen sein muss. Andernfalls hätte die Frau mit dem untreuen Ehemann mindestens einen untreuen Ehemann gekannt und daher in der ersten Nacht nicht sicher wissen können, dass ihr eigener Mann untreu war.

Es liegt nahe einen induktiven Beweis zu führen, um das Verhalten für beliebig viele untreue Ehemänner zeigen zu können (siehe **Abbildung 2.2.3-1**, Seite 8 und **Abbildung 2.2.3-2**, Seite 8):

- $t > 1$: es sei bis zur $(t-1)$ -ten Nacht ruhig geblieben.
- *Tag t* : alle Frauen mit untreuen Männern kennen $t-1$ andere untreue Männer, da aber keiner in der vergangenen Nacht erschossen wurde, gab es bisher noch keine Frau, die sicher wusste, dass ihr eigener Mann untreu war.
- Daher kann man implizieren, dass jede Frau mindestens $t-1$ untreue Männer kennen muss (Frauen von treuen Ehemännern kennen t untreue Ehemänner, Frauen von untreuen Ehemännern $t-1$ untreue Ehemänner).
- eine Frau die $t-1$ untreue Männer kennt, weiß nun also, dass (es ist ja der Tag t angebrochen) ihr eigener Mann untreu war, wird ihn also in der selben Nacht bestrafen.

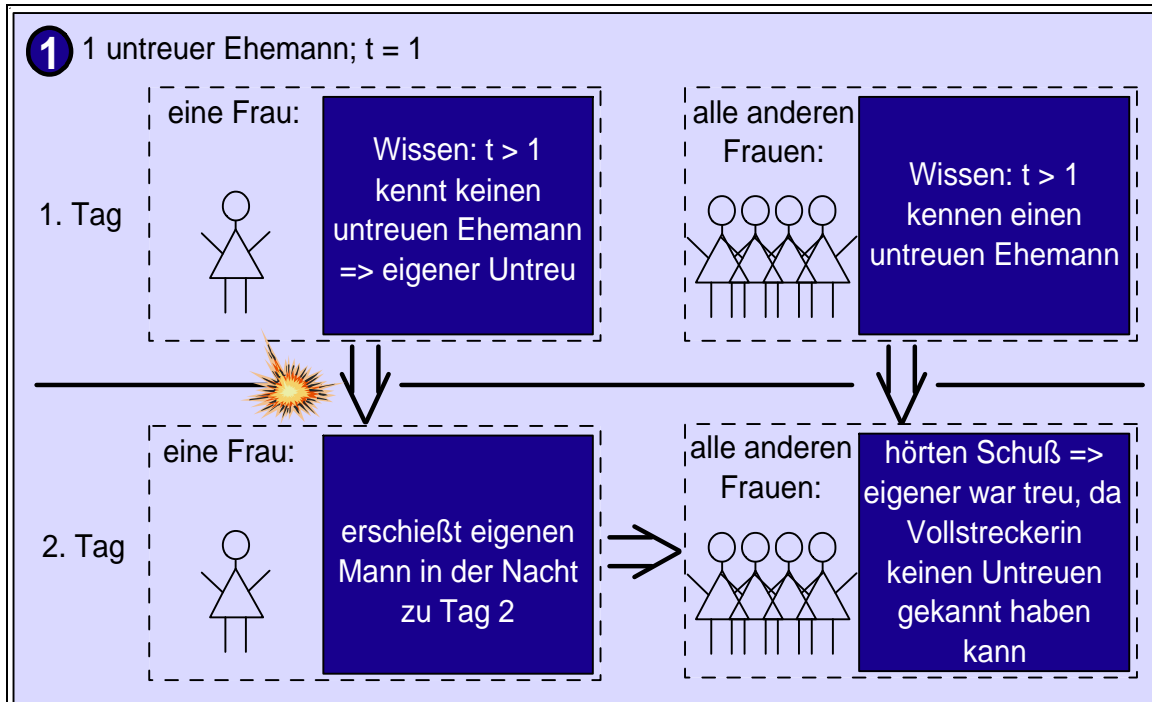


Abbildung 2.2.3-1 Problem der untreuen Ehemänner, Induktionsanker; Fall: ein untreuer Ehemann

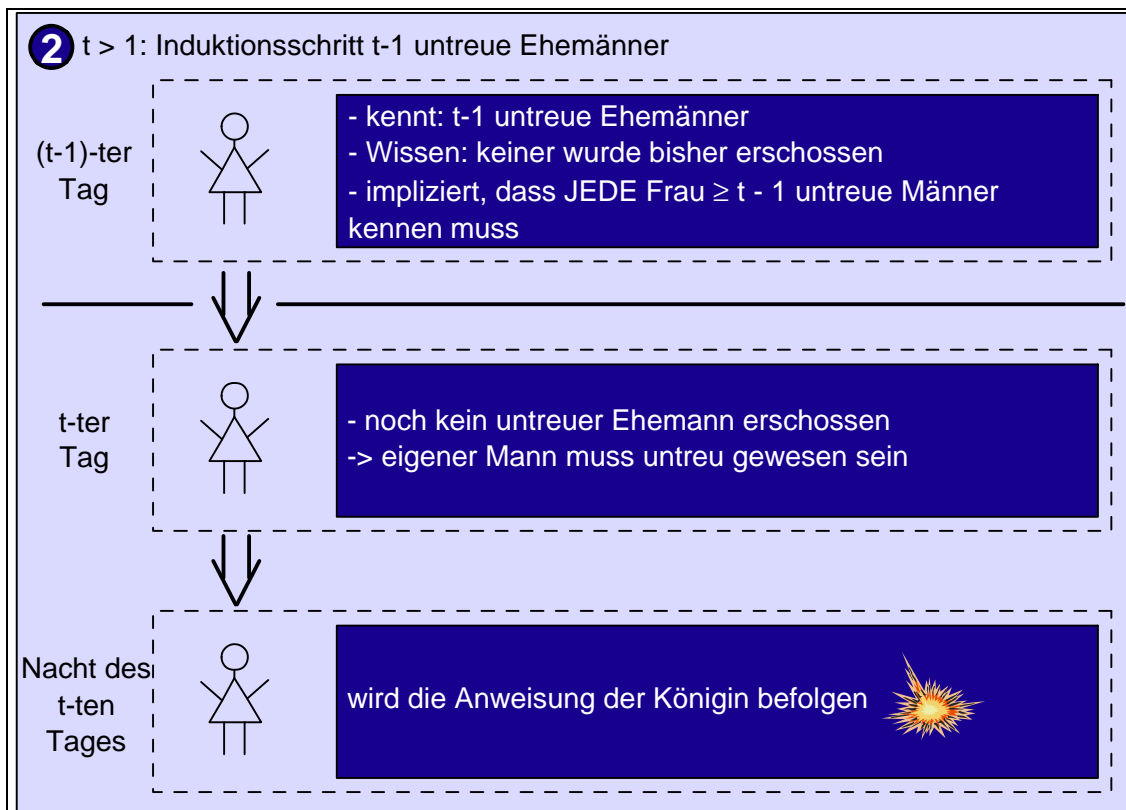


Abbildung 2.2.3-2 Problem der untreuen Ehemänner, Induktionsschritt; $t-1$ untreue Ehemänner

2.2.4 Analyse des globalen Wissens

Zu Beginn, nach der Verkündung der Königin, ist den Frauen nur bekannt, dass $t \geq 1$ ist, sowie die Tatsache, dass dies alle Frauen zur gleichen Zeit mitgeteilt bekommen haben. Für jede einzelne Frau ist die genaue Zahl t der untreuen Ehemänner aber nicht bekannt.

Jede Ehefrau, die t untreue Ehemänner kennt, kann daraus ableiten, dass die korrekte Zahl der untreuen Ehemänner t oder $t + 1$ betragen muss (je nachdem, ob ihr eigener Mann sie betrügt).

Weiterhin ist globales Wissen das Wissen über die anderen Frauen: Jede Frau weiß das es mindestens $t-1$ untreue Ehemänner gibt und auf der anderen Seite wissen diese Frauen, dass alle weiteren Frauen mindestens $t-2$ untreue Männer kennen.

Nach Ablauf der t -ten Nacht schließlich wird die korrekte Zahl t ebenfalls korrektes Wissen.

Hier hat wiederum Informationsübertragung ohne physikalische Datenübermittlung stattgefunden (vgl. Kapitel 2.2.2, Seite Fehler! Textmarke nicht definiert.). Siehe dazu

Abbildung 2.2.4-1 Seite 9.

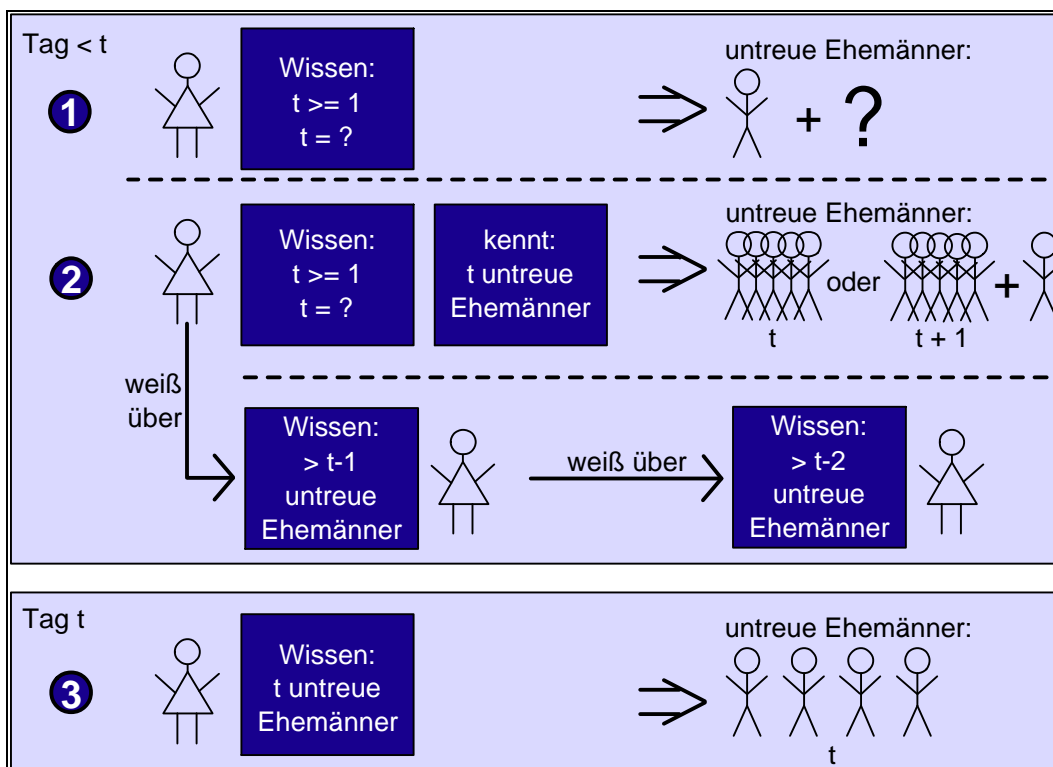


Abbildung 2.2.4-1 Das Problem der untreuen Ehemänner in seinen Phasen des globalen Wissens

2.2.5 Verzögerte Kommunikation im Fall der untreuen Ehemänner

Im nächsten Jahrhundert setzt Henrietta II die Treue-Kampagne ihrer Mutter fort, allerdings mit neuer Technologie.

Sie installiert ein Post-System auf der Insel, das jeden Brief in endlicher Zeit dem Empfänger zustellt.

Anstelle einer Versammlung auf dem Marktplatz geschieht die Unterrichtung aller Untertaninnen zum Thema „untreue Ehemänner“ nun dergestalt, dass die Königin zunächst Briefe an alle versendet mit Informationen über das neue Mail-System. Anschließend erhält jede Frau einen Brief mit dem gleichen Wortlaut wie das, was Henrietta I verkündet hatte.⁴

Leider ist das System von Henrietta II nicht durchdacht: im Gegensatz zu Henrietta I wird auf eine Synchronisation verzichtet (die Botschaft der Königin erreicht nicht gleichzeitig alle Frauen), stattdessen verlässt sie sich auf das die paarweise Kommunikation über dem Brief – allerdings mit unbekannter (unbeschränkter) Verzögerung.

Dadurch passiert auf der Insel gar nichts: Keine Frau auf der Insel kann sich sicher sein, dass alle anderen Frauen den Befehl der Königin erhalten haben. Daher müssen auch nicht zwingendermaßen alle Frauen bereits gehandelt haben.

Es zeigt sich also, dass in diesem jetzt asynchronen System keine Informationen mehr aus dem Nicht-Eintreten von Ereignissen gewonnen werden können.

(siehe **Abbildung 2.2.5-1** und **Abbildung 2.2.5-2**)

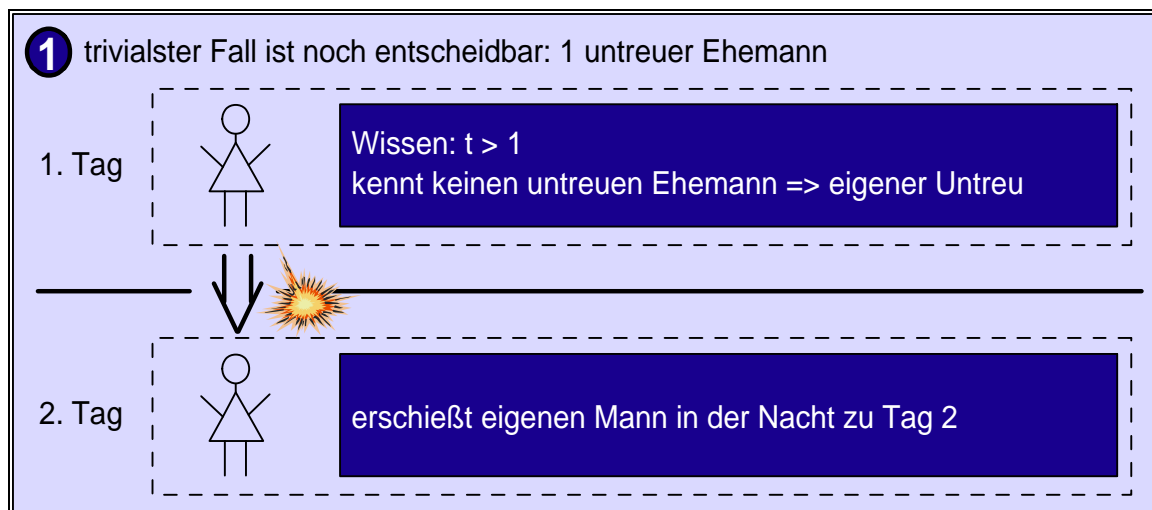
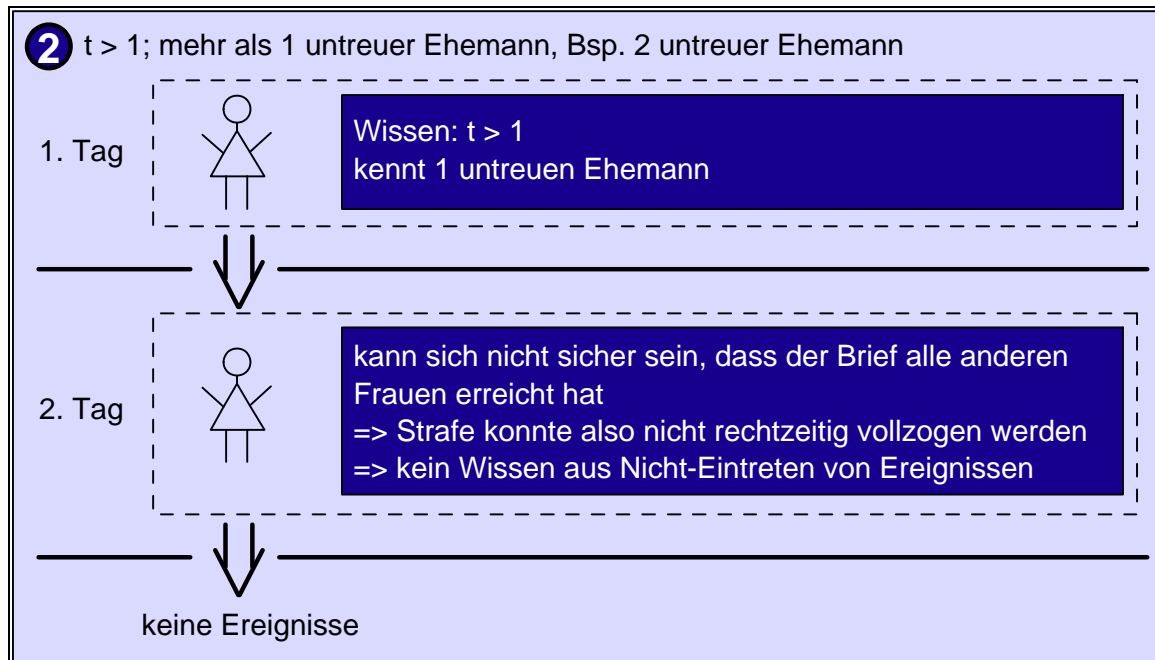


Abbildung 2.2.5-1 Henrietta II; Fall: 1 untreuer Ehemann, lösbar

⁴ siehe [1], Seite 160

Abbildung 2.2.5-2 Henrietta II, Fall $t > 1$ untreue Ehemänner

2.2.6 Das Problem der untreuen Ehemänner unter Henrietta III

*Henrietta III hat nun aus den Fehlern ihrer Mutter gelernt und verbessert den Post-Service. Jeder Brief wird nun garantiert noch am gleichen oder spätestens am nächsten Tag zugestellt.*⁵

Jetzt ist eine festgelegte Zeitspanne der Unsicherheit entstanden, die in diesem konkreten Fall $u = 2$ beträgt.

Die Überbringung der Nachricht (die gleiche, wie die der Großmutter) geschieht ebenfalls wieder über das neue Mail-System, dennoch entsteht eine große Ungerechtigkeit, derart, dass einige, aber nicht alle untreuen Ehemänner bestraft werden.

In der Analyse ergibt sich folgendes:

Jede Frau, die den Brief am ersten Tag erhält, muss berücksichtigen, dass eine andere Frau der Brief eventuell erst einen Tag später erreicht. Das heißt, erst mit einer Verzögerung von einem Tag kann aus dem Nicht-Eintreten von Ereignissen die Information gewonnen werden, dass keine andere Frau Gewissheit erlangt hat.

Bei allen Frauen, die den Brief später als andere Frauen erhalten, ergibt sich ein große Ungerechtigkeit: sie werden ihren Mann nicht erschießen. Eine Frau, die nämlich den Brief einen Tag später erhält und in ihrer 1. Nacht nach dem Erhalt des Briefs ($(2t-2)$ -te Nacht)

⁵ siehe [1] Seite 161

einen Schuss hört, muss annehmen, dass es nur $t - 1$ untreue Ehemänner gab, selbst wenn ihr eigener Ehemann untreu war. (siehe **Abbildung 2.2.6-1**, Seite 12)

Das Problem lässt sich durch das Einführen einer Wartezeit von $d \geq u - 1$ lösen. Jede Frau muss, nachdem sie Gewissheit über die Untreue ihres Ehemannes erlangt hat, noch die Wartezeit d einhalten, bevor sie ihren Mann erschießt. Zwar wird die Verbreitung von Informationen über das konkrete t in diesem Fall mit $u + d$ verzögert, aber alle Frauen können jetzt mit Sicherheit handeln.

Weiterhin ist es möglich durch die Einführung eines globalen Kalenders und dem Abstempeln von Briefen die Laufzeit eines Briefes zu erkennen. Dadurch kann jede Frau gehörte Schüsse richtig zeitlich einordnen und Konsequenzen für sich selber ziehen.

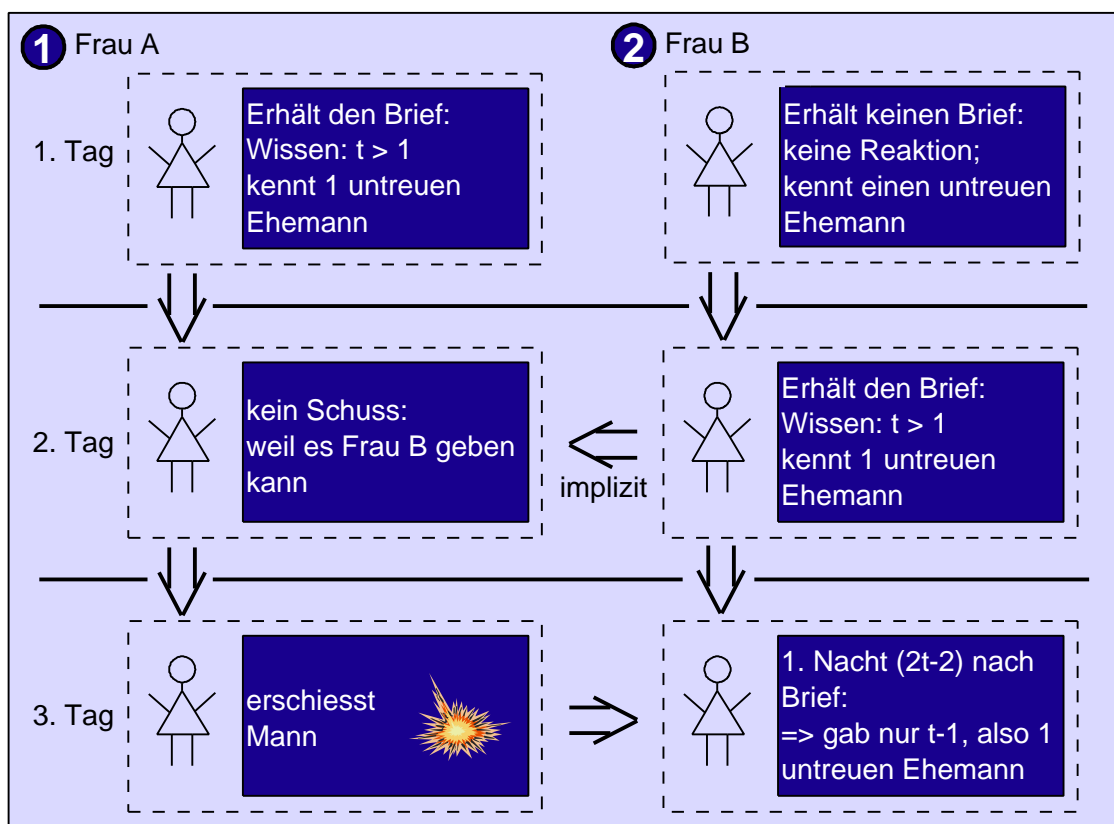


Abbildung 2.2.6-1 Wissen unter Henrietta III

2.2.7 Henrietta IV

Die Geschichte endet damit, dass Henrietta IV alle Untertaninnen noch einmal auf dem Marktplatz versammelt und mit diesem Tag den Mamajorcischen Kalender einführt. Ihre weiteren Bemühungen, die Kampagne der Vorfahren fortzuführen, scheitern jedoch daran, dass ihre Untertaninnen inzwischen das Vertrauen in die Weisheit ihrer Herrinnen verloren

hatten. Man konnte nicht mehr bei allen Frauen sicher sein, dass das neue Dekret peinlich genau befolgt werde würde.⁶

Das fehlerhafte Arbeiten einzelner Komponenten (hier Frauen) führt selbst in synchronen Systemen zu einer deutlichen Verschärfung der Probleme: Selbst wenn einzelne Frauen loyal arbeiten, können Sie es nicht mehr als erwiesen ansehen, dass ihr eigener Ehemann untreu war, wenn bisher keine Schüsse fielen. Die globale Konsistenz ist hier also unmöglich zu erreichen.

2.3 Beschränkt asynchrone Systeme

Beschränkt asynchrone Systeme unterscheiden sich von synchronen Systemen vor allem dadurch, dass durch unterschiedliche Arbeitsgeschwindigkeiten der einzelnen Prozessoren die exakte Antwortzeit nicht vorausgesehen werden kann. Ein *Immediate Agreement*, bei dem sich alle korrekten Prozesse gleichzeitig entscheiden (vgl. Firing-Squad-Problem, Kapitel 2.2.1, Seite 4), ist deshalb bei beschränkt asynchronen Systemen nicht vorauszusetzen.

Vielmehr begnügt man sich hier mit einem *Eventual Agreement*, bei dem alle korrekten Prozesse zwar den gleichen Wert, allerdings zu verschiedenen Zeitpunkten wählen. Um dies zu gewährleisten müssen hier maximale und minimale Antwortzeiten festgelegt werden.

2.3.1 Das Problem der Byzantinischen Generäle (Byzantinisches Agreement)

Dieses Problem beschreibt die einvernehmliche Entscheidungsfindung (Agreement) verteilter Systeme durch Mehrheitsentscheid.

Man kann sich dieses System als einen Graphen vorstellen, bei dem jeder Knoten einen eigenständigen Prozess, veranschaulicht einen General, darstellt, der beginnend mit einem Ausgangswert aus mehreren Eingabewerten von benachbarten Knoten einen Entscheidungswert produziert und diesen an diese Knoten sprich Generäle zurückliefert.

⁶ siehe [1], Seite 162

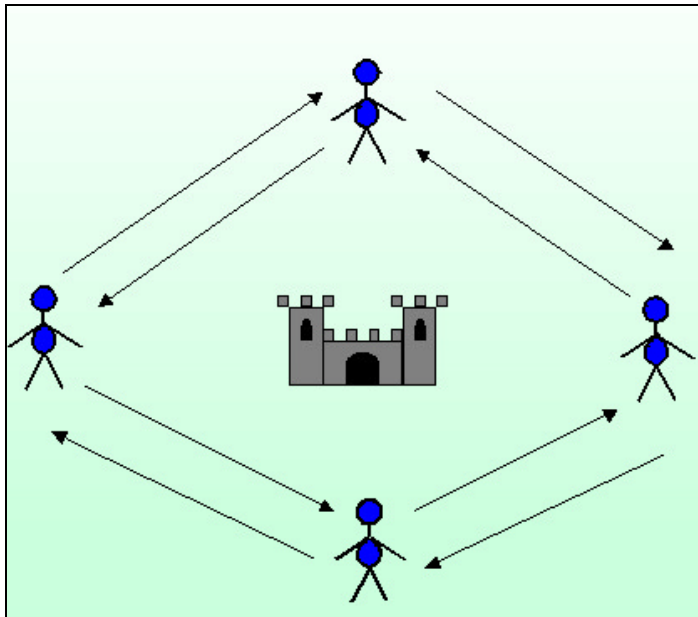


Abbildung 2.3.1-1 Das Problem der Byzantinischen Generäle

Es ist hierbei nicht zwingend jeder Knoten direkt mit jedem anderen verbunden, es muss aber von jedem Knoten, zumindest mit dem Umweg über andere Knoten, ein Weg zu jedem anderen existieren.

Idealerweise sollte jeder Prozess eine ungerade Anzahl von mindestens 3 Eingabewerten erhalten, um Pattsituationen bei der Entscheidungsfindung zu vermeiden. Sollte diese Pattsituation trotzdem eintreten, so ist ein vordefinierter Entscheidungswert zu produzieren.

Die maximale Anzahl der fehlerhaften Prozesse ist allerdings beschränkt und abhängig von der Gesamtzahl der Prozesse.

Die Zeitspanne vom Versenden des Ausgangswertes über den Empfang der Eingabewerte von den benachbarten Knoten bis zur neuen Entscheidungsfindung wird als eine Runde bezeichnet.

Der Ausgabewert muss in Abhängigkeit vom Eingabewert folgenden Bedingungen genügen:

- *Terminierung (Wait Freeness)*: Jeder korrekte Prozess muss nach endlich vielen Runden eine Entscheidung treffen. Da diese Entscheidung von den Informationen anderer, möglicherweise nicht korrekter Prozesse abhängt, kann verschärfend festgelegt werden, dass diese Entscheidung innerhalb einer vorgegebenen Zeitschranke gefallen sein muss.
Diese Bedingung ist essenziell, um beim Ausfall einer Komponente nicht ewig auf dessen ausbleibendes Ergebnis zu warten.
- *Übereinstimmung (Agreement)*: Alle korrekten Prozesse entscheiden sich letztendlich für den gleichen Wert.

Ansonsten wäre keine Gesamtentscheidung (Agreement) des Verteilten Systems möglich.

- *Sinnhaftigkeit (Validity)*: Als Entscheidungswerte sind nur die aktuellen Eingabewerte zulässig. Insbesondere muss ein korrekter Prozess bei identischen Eingabewerten diesen auch als Entscheidungswert wählen. Das sorgt dafür, dass zumindest bei eindeutiger Ausgangslage die Entscheidung korrekter Prozesse eben diese widerspiegelt.
- *Nichttrivialität*: Die Menge der Entscheidungswerte umfasst selbstverständlich mehr als ein Element. Ansonsten wäre ein Entscheidungsverfahren aufgrund des bereits vorher feststehenden Ergebnisses sinnlos.

Ein Entscheidungsverfahren mit diesen Eigenschaften wird als Byzantinisches Agreement oder in dieser Version als nichttrivialer Consensus bezeichnet. Die Einzelentscheidungen nicht korrekter Prozesse werden durch einen einfachen Mehrheitsentscheid herausgefiltert.

So kann das Problem einzelner nicht korrekter Prozesse, also nicht loyaler Generäle, die falsche Meldungen verbreiten, kompensiert werden.

2.3.2 Mögliche Fehlerarten beschränkt asynchroner Systeme

Zur Vereinfachung wird angenommen, dass alle Leitungen fehlerfrei sind, eine ausbleibende Nachricht also auf einen defekten Prozessor zurückzuführen ist.

- *Fail-Stop (Crash)*: kompletter Ausfall eines Prozessors.
- *Omission (partielle Unterbrechung)*: Einige Nachrichten gehen verloren, alle ankommenden Nachrichten sind allerdings korrekt.
- *Timing*: Aktionen geschehen zu früh oder zu spät.
- *Byzantinische Fehler*: Prozessoren arbeiten zwar, liefern allerdings falsche Ergebnisse.

Unterkategorien Byzantinischer Fehler:

- *symmetrische Fehler*: Ein inkorrekt arbeitender Prozessor liefert immer den gleichen Fehler.

Solche Fehler können durch das Byzantinische Agreement kompensiert werden.

- *erkennbare Fehler*: Eine falsche Nachricht kann vom Empfänger als solche erkannt werden.

Durch entsprechende Codierung lassen sich Übertragungsfehler kompensieren, falsche Informationen lassen sich evtl. durch lokales Wissen enttarnen.

- *beweisbare Fehler*: Ein korrekter Prozessor kann anderen Prozessoren einen Beweis für die Unkorrektheit eines dritten erbringen.
Dies ist durch Authentisierungsverfahren in Form von digitalen Signaturen möglich.

Ohne die Möglichkeit Fehler anderer beweisen zu können reicht auch eine Redundanz von 3, also 3 autonomen Systeme nicht in jedem Fall aus, um einen Fehler zu kompensieren.

Nehmen wir als Beispiel ein System von 3 Prozessoren P_1 , P_2 und P_3 an, die jeweils mit den beiden anderen verbunden sind.

Sendet nun P_1 fehlerhaft einen Wert von 0 an P_2 und einen Wert von 1 an P_3 , so können sich P_2 und P_3 nicht auf einen gemeinsamen Wert einigen, da keiner von beiden beweisen kann, dass er den fehlerhaften Wert von P_1 erhalten und nicht selbst produziert hat.

Würde P_1 seine Nachrichten allerdings mit einer digitalen Signatur versehen müssen, so könnten sich P_2 und P_3 gegenseitig die unterschiedlichen Werte von P_1 zusammen mit dessen Signatur präsentieren und somit P_1 in nur 2 Runden als fehlerhaft identifizieren.

2.4 Vollkommen asynchrone Systeme

In vollkommen asynchronen Systemen ist nichttrivialer Consensus nicht möglich. Das liegt nicht an den individuellen Geschwindigkeitsunterschieden der einzelnen Prozessoren, sondern vielmehr an der möglichen Nachrichtenverzögerung, sodass keinerlei maximale Übertragungszeit und somit keine Terminierung eines Consensus garantiert werden kann.

2.5 Fazit

Abschließend möchten wir die unterschiedlichen Vor- bzw. Nachteile der verschiedenen Synchronisierungsmodelle gegenüberstellen.

Synchrone Systeme (vgl. Kapitel 2.1, Seite 3) eignen sich aufgrund ihrer nötigen Gleichtaktung vor allem für Anwendungen mit geringer räumlicher Ausdehnung wie zum Beispiel bei Prozessorfeldern und anderen integrierten Schaltungen. Auch das LAN⁷ selber läuft in getakteten Runden ab, wobei die eigentliche Kommunikation über das LAN als

⁷ LAN = Local Area Network

beschränkt asynchron zu bezeichnen ist. Gerade in sicherheitskritischen Bereichen kommt man aufgrund ihrer hohen Fehlerkompensation nicht an synchronen Systemen vorbei.

Bei beschränkt asynchronen Systemen (vgl. Kapitel 2.3, Seite 13) ist eine höhere räumliche Ausdehnung möglich, es müssen allerdings aufgrund der daraus resultierenden eventuellen Nachrichtenverzögerung und der möglichen Geschwindigkeitsunterschiede der einzelnen Prozessoren zusätzliche zeitliche Vereinbarungen getroffen werden. Die geforderten minimalen und maximalen Antwortzeiten sind ein wesentlicher Bestandteil beschränkt asynchroner Systeme. Das bekannteste Beispiel dürfte das Internet sein, in dem keine Aussage über die einzelnen Prozessorgeschwindigkeiten getroffen werden kann, zu lange Verzögerungen werden allerdings durch *Timeouts* abgefangen.

Vollkommen asynchrone Systeme (vgl. Kapitel 2.4 Seite 16) sind sehr schwer zu beherrschen, da keine Aussage über die zeitliche Verzögerung von Nachrichten getroffen werden kann.

Eine Terminierung zeitlich abhängiger Prozesse kann in vollkommen asynchronen Systemen nicht garantiert werden. Deswegen kann hier ein nichttrivialer Consensus nicht erreicht werden.

Hinweis:

Bei allen Grafiken ist „ $t > 1$ “ durch „ $t \geq 1$ “ zu ersetzen

3.0 Index

- Agreement 13, 14
Beschränkt asynchrone Systeme 13
Byzantinische Fehler 15
Byzantinischer Fehler
 beweisbare Fehler 16
 erkennbare Fehler 15
 symmetrische Fehler 15
Byzantinisches Agreement 13
Consensus 17
Crash 15
Entscheidungsfindung 14
Eventual Agreement 13
Fail-Stop 15
Fazit 16
Fehlerarten 15
fehlerhafte Komponenten 13
Firing-Squad-Problem 4
General 13
globaler Kalender 12
globales Wissen 5, 9
Henrietta I 5
Henrietta II 9
Henrietta III 11
Henrietta IV 12
Immediate Agreement 13
Informationsübertragung 9
LAN 16
Laufzeit 12
Mamajorca 5
Manipulationen 4
maximale Übertragungszeit 4
Nichttrivialität 15
Omission 15
partielle Unterbrechung 15
Pattsituation 14
Post-Service 11
Post-System 9
Problem der Byzantinischen Generäle 13
Problem der untreuen Ehemänner 5
Prozess 13
Sinnhaftigkeit 15
Synchrone Systeme 3
Synchronisationsproblem 5
synchronisierte Uhren 4
Terminierung 14, 17
Theorem
 untreuer Ehemänner 6
Timeout 17
Timeouts 3
Timing 15
Übereinstimmung 14
Validity 15
verzögerte Kommunikation 9
Verzögerung 10
Vollkommen asynchrone Systeme 16
Wait Freeness 14
Wartezeit 12
Zeitschranke 14

4.0 Abbildungsverzeichnis

Abbildung 2.2.1-1 Firing-Squad-Problem, alle Soldaten sind direkt erreichbar	4
Abbildung 2.2.1-2 Firing-Squad-Problem mit zeitlicher Synchronisation, räumliche Trennung	5
Abbildung 2.2.2-1 Henrietta I verkündet das Problem der untreuen Ehemänner	6
Abbildung 2.2.3-1 Problem der untreuen Ehemänner, Induktionsanker; Fall: ein untreuer Ehemann	8
Abbildung 2.2.3-2 Problem der untreuen Ehemänner, Induktionsschritt; $t-1$ untreue Ehemänner	8
Abbildung 2.2.4-1 Das Problem der untreuen Ehemänner in seinen Phasen des globalen Wissens	9
Abbildung 2.2.5-1 Henrietta II; Fall: 1 untreuer Ehemann, lösbar	10
Abbildung 2.2.5-2 Henrietta II, Fall $t > 1$ untreue Ehemänner	11
Abbildung 2.2.6-1 Wissen unter Henrietta III.....	12
Abbildung 2.3.1-1 Das Problem der Byzantinischen Generäle	14

5.0 Quellenverzeichnis

- [1] Rüdiger Reischuk: Zeit und Raum in Rechnernetzen, in I. Wegener (Ed.), Highlights aus der Informatik, Springer Verlag Heidelberg 1996, S. 155-176.

- [2] Prof. Tobias Nipkow: Das Byzantinische Übereinkommen, TU München, Januar 2000, <http://www4.informatik.tu-muenchen.de/lehre/seminare/ps/WS9900/ByzGen.ps.gz>

- [3] Michael Weber: Verteilte Systeme, Uni Ulm, WS 2000/2001, <http://medien.informatik.uni-ulm.de/lehre/courses/ws0001/verteilte2/Kap12.pdf>

- [4] Prof. Dr. Angelika Steger: Randomized Algorithms, TU München, WS 2002/2003, <http://home.in.tum.de/~puehler/ra/lectures/lec20021212.pdf>