



Hochschule für Angewandte Wissenschaften Hamburg

*Hamburg University of Applied Sciences*

# Modellierung und Verifikation des Token Ring Algorithmus

Birger Kamp und Maria Lüdemann  
Formale Simulation und Verifikation verteilter  
Algorithmen  
Sommersemester 2016

Birger Kamp und Maria Lüdemann  
Formale Simulation und Verifikation verteilter  
Algorithmen  
Sommersemester 2016

Modellierung und Verifikation des Token Ring Algorithmus eingereicht  
im Rahmen des Projekts New Storytelling  
im Studiengang Master Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuer: Sascha Kluth

Abgegeben am 05. Februar 2015

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Aufteilung . . . . .	4
<b>2</b>	<b>Hauptteil</b>	<b>4</b>
2.1	Der Token Ring Algorithmus . . . . .	4
2.2	Spezifikation . . . . .	5
2.3	Modellierung . . . . .	6
2.3.1	Das Netz . . . . .	6
2.4	Korrektheit . . . . .	6
<b>3</b>	<b>Zusammenfassung und Ausblick</b>	<b>7</b>

# 1 Einleitung

Dieses Dokument beschreibt die Ergebnisse des SVA Praktikums. Dabei sollte ein verteilter Algorithmus gewählt werden um ihn dann in mehreren Schritten in einem Petri Netz zu modellieren, zu spezifizieren und seine Korrektheit zu zeigen.

## 1.1 Aufteilung

Die Arbeit an diesem Dokument teilt sich wie folgt auf:

Teil 1	Name
Teil 2	Name

Tabelle 1: Arbeitsteilung

# 2 Hauptteil

## 2.1 Der Token Ring Algorithmus

Der Token Ring Algorithmus ist ein Wahlalgorithmus der von Chang und Roberts 1979 entworfen wurde. Er kann verteilt auf mehreren Clienten verwendet werden die in einer Ring-Topologie miteinander verbunden sind. Das Ziel des Algorithmus ist, bei Ausfall des Master-Clients im Netz einen neuen zu wählen.

### Voraussetzungen

Damit der Algorithmus auf eine Ring-Topologie angewandt werden kann, müssen folgende Voraussetzungen im Netz gegeben sein:

- Jeder Client kennt seinen Nachfolger
- Jeder Client ist mit seinem Nachfolger verbunden, sodass er mit ihm kommunizieren kann
- Jeder Client hat eine eindeutige ID
- Jeder Client kennt die gesamte Ring-Topologie

## Ablauf

Der Algorithmus startet wenn der Master-Client ausfällt. Der Vorgänger des ausgefallenen Master-Clients baut eine Verbindung zum Nachfolger des ausgefallenen Master-Clients auf, sodass die Ring-Topologie wieder vollständig ist.

Der Client, der den Ausfall bemerkt, startet die Wahl in dem er seinem Nachfolger eine Nachricht mit seiner ID und der Info dass es sich um eine Wahl handelt schickt. Dieser nimmt die Nachricht und überprüft ob seine eigene ID darin vor kommt. Falls nicht, hängt er seine eigene ID hinten an und schickt die vervollständigte Nachricht an seinen Nachfolger.

Wenn ein Client feststellt, dass seine eigene ID bereits in der Nachricht vorhanden ist, nimmt er die höchste ID aus der Liste der gesammelten IDs in der Nachricht. Anschließend sendet er eine "GewähltNachricht mit der höchsten ID an seinen Nachfolger. Der Empfänger der "GewähltNachricht merkt sich, dass der gewählte Client nun der neue Master ist und sendet seinem Nachfolger die gleiche "GewähltNachricht. Jeder wird somit benachrichtigt was die höchste ID ist. Kommt die "Gewählt"Nachricht wieder am Initiator der "GewähltNachricht an, wird die Wahl erfolgreich beendet und der Algorithmus ist terminiert.

## Eigenschaften

Laufzeit Welche Grundlegenden Eigenschaften hat der Algorithmus was tut er und warum, wofür?

## 2.2 Spezifikation

Damit ein Modell erstellt werden kann, das den Algorithmus abbildet, müssen zunächst die charakteristischen Eigenschaften des Algorithmus bestimmt werden.

Der Algorithmus (s. Abschnitt 2.1) lässt sich in folgende drei Phasen einteilen:

**Phase 1:** Ein Client bemerkt den Ausfall des bisherigen Masters

**Phase 2:** Sammeln aller beteiligten Client-IDs

**Phase 3:** Bekanntgeben des neuen Masters

Der jeweilige Ablauf der Phasen lässt sich mit folgenden Punkten spezifizieren:

Phase	Eigenschaft
Phase 1: Master-Ausfall bemerkt	Sendet Nachricht zum Wählen und hängt seine eigene ID daran
Phase 2: Wahl	Client, der Wahl-Nachricht erhält, hängt seine eigene ID an die Nachricht
	Client sendet die erweiterte Nachricht an seinen Nachfolger
	Sobald ein Client eine Wahl-Nachricht erhält, in der seine eigene ID bereits enthalten ist, geht der Algorithmus in Phase 3 über
Phase 3: Neuen Master mitteilen	Der Client, der feststellt, dass Phase 2 vorbei ist, sendet eine Nachricht mit dem neuen Master an seinen Nachfolger
	Ein Client, der die Nachricht über einen Master erhält, merkt sich den neuen Master
	Ein Client, der die Nachricht über einen Master erhält, teilt seinem Nachfolger diese Nachricht mit
	Sobald der Client, der Phase 3 eingeleitet hat, die Nachricht über den neuen Master erhalten hat, terminiert der Algorithmus

## 2.3 Modellierung

Wie haben wir ihn modelliert -Gefärbtes netz - Ids - Guards Erklären an wo die spezifizierten Punkte im Netz zu finden sind.

### 2.3.1 Das Netz

Hier Netzbild einbinden

## 2.4 Korrektheit

Warum ist der Algorithmus korrekt - Determinismus - es kann immer nur eine Transition schalten es gibt keine Parallelität - Erreichbarkeitsgraph zeigen und erklären

## 3 Zusammenfassung und Ausblick

Netz flachklopfen um über IDs ein beliebig großes Netz generieren zu können um zu zeigen, dass auch bei steigender Clienten Zahl der Erreichbarkeitsbaum gradlinig deterministisch bleibt