Charakterisierung der Lösbarkeit von Consensus unter Nachrichtengegnern[1]

Kyrill Winkler[2]

Abstract: Motiviert durch die zunehmende Allgegenwärtigkeit von kleinen, mobilen, drahtlos kommunizierenden Endgeräten und den daraus resultierenden dynamischen Netzen, beschäftigt sich diese Arbeit mit den theoretischen Grundlagen von verteilten Algorithmen in solchen Systemen. Unser Hauptaugenmerk liegt hier auf der Charakterisierung der Lösbarkeit des klassischen Consensus-Problems in dynamischen Netzen, in denen beliebige Nachrichten verloren gehen können. Dafür verwenden wir das Modell der Nachrichtengegner, welches es uns erlaubt, einen beinahe beliebigen dynamischen Verlust von Nachrichten in einem synchronen System darzustellen. Wir betrachten verschiedene Aspekte dieser Fragestellung, beginnend mit dem verhältnismäßig simplen Fall der grenzwertmäßig abgeschlossenen Nachrichtengegner. Wir gehen dann über zu stabilisierenden Nachrichtengegnern, die diese Eigenschaft nicht besitzen, bevor wir schließlich, mit der Hilfe von Punktmengentopologie, eine Charakterisierung von allgemeinen Nachrichtengegnern präsentieren.

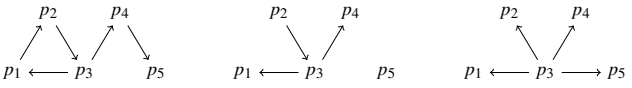
1 Grundlagen

Bei drahtloser Kommunikation kann es, aufgrund eingeschränkter Ressourcen (wie z.B. schwacher Energiespeicher oder Übertragungsleistung), sowie wegen lokalen Interferenzphänomenen (wie z.B. dem Capture Effekt) zum Verlust einzelner Nachrichten kommen. Zumindest konzeptionell wäre das so, als würde ein Angreifer ("Nachrichtengegner") bestimmte Nachrichten unterdrücken um die kommunizierenden Teilnehmer zu stören. Ein mathematisches Modell, das es uns erlaubt solche Systeme abzubilden, ist das Nachrichtengegner-Modell von Afek und Gafni. Eng verwandte Modelle wurden bereits von Santoro und Widmayer [SW89], sowie von Charron-Bost und Schiper [CBS09] für ähnliche Fragestellungen untersucht. Das Nachrichtengegner-Modell ist jedoch allgemeiner und enthält diese Modelle als Spezialfall. Es beschreibt ein Ensemble von n Zustandsautomaten, die *Prozesse* genannt werden, und durch den Austausch von Nachrichten miteinander kommunizieren. Ein Nachrichtengegner sorgt dafür, dass einzelne Nachrichten verloren gehen. Wir nehmen an, dass die Prozesse einzigartige Identifikationsnummern haben und n kennen. Unter einem bestimmten Nachrichtengegner ist das Lösen eines verteilten Problems genau dann möglich, wenn die Prozesse eine Strategie bzw. einen Algorithmus haben um den Nachrichtengegner zu überwinden. Wenn dagegen der Nachrichtengegner eine Strategie hat, welche es den Prozessen unmöglich macht, das Problem zu lösen, dann ist das Problem unter diesem Nachrichtengegner unlösbar.

[1] Englischer Titel der Dissertation [Wi19]: "Characterization of Consensus Solvability under Message Adversaries". Diese Kurzfassung beruht auf dem eingeladenen Übersichtsartikel [WS19].

[2] Technische Universität Wien, kwinkler@ecs.tuwien.ac.at

cc by sa



G_1 für Runde 1 G_2 für Runde 2 G_3 für Runde 3

Abb. 1: Kommunikationsgraphen G_1, G_2, G_3 für 3 Runden.

Formal agiert ein Nachrichtengegner in einem synchronen System: Zu Beginn einer Runde r werden Nachrichten anhand eines sogenannten *Kommunikationsgraphen* G_r ausgetauscht. Dieser Graph ist gerichtet und die Anwesenheit einer Kante (p,q) in G_r bedeutet dass die entsprechende Nachricht von p nach q in Runde r erfolgreich zugestellt werden kann, während ihre Abwesenheit bedeutet dass der Nachrichtengegner diese Nachricht unterdrückt. Anschließend berechnet jeder Prozess, abhängig von den empfangenen Nachrichten, einen Folgezustand, woraufhin die nächste Runde beginnt (siehe Abb. 1). In einem deterministischen Algorithmus ist die Abfolge der Zustände der Prozesse durch deren Initialzustand sowie der Sequenz der Kommunikationsgraphen eindeutig bestimmt. Wir nennen eine solche Sequenz von Kommunikationsgraphen ein *Kommunikationsmuster* und definieren einen konkreten Nachrichtengegner über die Menge an *zulässigen* Kommunikationsmustern, also jenen die er generieren bzw. gegen die Prozesse spielen kann. Ein *Präfix* (die ersten paar Kommunikationsgraphen) eines Kommunikationsmusters wird zulässig genannt, wenn er der Präfix eines zulässigen Kommunikationsmusters ist.

In dieser Zusammenfassung interessieren wir uns für die Lösbarkeit des Consensus Problems [FLP85] im Nachrichtengegner-Modell. Bei dieser klassischen Problemstellung starten alle Prozesse mit einem Eingangswert, (üblicherweise) aus der Menge $\{0,1\}$. Ihre Aufgabe ist es, sich irgendwann auf einen Ausgangswert zu einigen, welcher der Eingangswert mindestens eines Prozesses war. Dieses fundamentale Problem abstrahiert die Fähigkeit dass mehrere Prozesse das selbe Ergebnis liefern, wie es insbesondere der Fall ist wenn man Fehlertoleranz mit der Hilfe von Replikation erreichen möchte. Hier kann Consensus Replikadeterminismus, also ein identes Verhalten der Replikate, sicherstellen.

Für verschiedene Spezialfälle wurde diese Frage bereits früher beantwortet: Für zwei Prozesse wurde diese Frage von Fevat und Godard in [FG11] studiert und beantwortet. Santoro und Widmayer beschreiben im Hauptresultat von [SW89] eine Lösbarkeitscharakterisierung für Consensus mit mobilen Verbindungsfehlern. Hier wird der Nachrichtengegner über die Anzahl der Nachrichten definiert, die jede Runde unterdrückt werden dürfen. Dieses Resultat wurde später von Couloma, Godard und Peters in [CGP15] verallgemeinert, für die sogenannten *vergesslichen* Nachrichtengegner. Ein Nachrichtengegner dieser Klasse ist definiert über eine Menge an erlaubten Kommunikationsgraphen, wobei er jede Runde einen beliebigen Graphen aus dieser Menge auswählen darf und stellt wiederum einen Spezialfall eines abgeschlossenen Nachrichtengegners dar.

Diese Zusammenfassung bietet einen Überblick über die in [Wi19] etablierten Bedingungen, die ein Nachrichtengegner erfüllen muss, um Consensus lösbar zu machen. Zunächst präsentieren wir eine rein kombinatorische Bedingung für die Lösbarkeit von Consensus unter der Klasse der abgeschlossenen Nachrichtengegner. Anschließend erklären wir kurz die praktisch relevanten stabilisierenden Nachrichtengegner und geben exempla-

risch eine Charakterisierung für einen generischen Nachrichtengegner aus dieser Klasse. Schließlich zeigen wir, wie man Punktmengentopologie benutzen kann, um die Lösbarkeit von Consensus unter allgemeinen Nachrichtengegnern zu charakterisieren. Die Ergebnisse der Dissertation wurden in den Tagungsbändern mehrerer internationaler Konferenzen [Sc14, Bi15, SWS16, SSW18, WSM19] und Journalen [Bi18, WSS19] publiziert[3]; ein eingeladener Übersichtsartikel ist im EATCS Bulletin [WS19] erschienen.

2 Abgeschlossene Nachrichtengegner

Eine besonders nützliche fundamentale Eigenschaft von Nachrichtengegnern ist ihre Abgeschlossenheit (vgl. auch [LM95]). Intuitiv verständlich wird sie, wenn wir die zulässigen Kommunikationsmuster eines Nachrichtengegners als Baum darstellen, wie in Abb. 2.

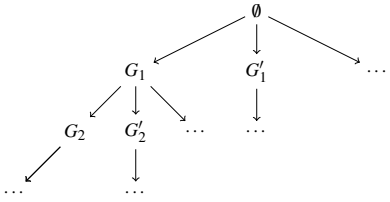


Abb. 2: Ein Nachrichtengegner, dargestellt als Baum. Jeder unendliche Pfad (z.B. $G_1, G_2', \ldots$) von der Wurzel aus korrespondiert zu einem zulässigen Kommunikationsmuster und umgekehrt.

Hier repräsentieren die Pfade von der Wurzel bis zu einer Ebene r die ersten r Runden eines zulässigen Kommunikationsmusters. Ein Nachrichtengegner ist nun abgeschlossen wenn jeder infinite Pfad in diesem Baum zu einem zulässigen Kommunikationsmuster korrespondiert. Dies ist keineswegs selbstverständlich: Der Nachrichtengegner etwa, der garantiert dass ein bestimmter Graph G nach einer unbestimmten, jedoch finiten Zeit gespielt wird, erfüllt diese Eigenschaft nicht. Hier gibt es nämlich für jede Runde einen Präfix, bzw. einen entsprechenden Pfad, auf dem G noch nicht gespielt wurde und darum beinhaltet der Baum den infiniten Pfad, in dem der Graph nie gespielt wird (der unter diesem Nachrichtengegner zu keinem zulässigen Kommunikationsmuster korrespondiert). Etwas präziser definiert man diese Eigenschaft folgendermaßen.

Definition 1. *Sei (σ_i) eine beliebige Sequenz von zulässigen Präfixen mit der der Eigenschaft dass σ_i ein Präfix von σ_{i+1} ist. Ein Nachrichtengegner ist abgeschlossen wenn der Grenzwert $\lim_{i\to\infty}(\sigma_i)$ ein zulässiges Kommunikationsmuster ist.*

Diese Eigenschaft schränkt den Nachrichtengegner stark ein, denn sie erlaubt es nicht, Dinge wie "irgendwann wird etwas Gutes passieren" auszudrücken (vgl. [AS85]). So ist etwa der relativ einfache Nachrichtengegner, der garantiert, dass irgendwann eine vollständig verbundener Kommunikationsgraph auftaucht, nicht abgeschlossen, obwohl ein Lösungsalgorithmus auf der Hand liegt: Ein Prozess wartet einfach bis er von allen anderen Prozessen gehört hat und entscheidet dann z.B. das Maximum aller Eingangswerte.

[3] Eine detaillierte Darstellung der Zusammenhänge dieser Publikationen mit der Dissertation befindet sich in Abschnitt 1.3 von [Wi19].

Nichtsdestotrotz wurden alle bisherigen Lösbarkeitscharakterisierungen von Consensus mit mehr als zwei Prozessen für abgeschlossene Nachrichtengegner formuliert (insbesondere sind die Nachrichtengegner von [CGP15] und [SW89] abgeschlossen). In Wahrheit sind abgeschlossene Nachrichtengegner sogar noch allgemeiner als die bisherigen Klassen, denn sie erlauben insbesondere Formulierungen der Form "innerhalb von x Runden wird etwas Gutes passieren", wobei x den Prozessen bekannt ist.

Interessanterweise gibt es eine relativ einfache Lösbarkeitscharakterisierung für Consensus unter abgeschlossenen Nachrichtengegnern, die wir im folgenden kurz vorstellen. Sie basiert auf dem sogenannten *Kern* eines Kommunikationsmusters bzw. dessen Präfix, geschrieben als $\mathrm{Ker}(\sigma)$, welcher jene Prozesse beinhaltet, von denen alle Prozesse gehört haben. In Abb. 1 sind zum Beispiel p_1, p_2 und p_3 im Kern des abgebildeten Präfixes. Desweiteren benutzen wir die transitive Hülle der Ununterscheidbarkeitsrelation, einer Äquivalenzrelation, die alle jene Paare von (Präfixen von) Kommunikationsgraphen beinhaltet, welche für irgendeinen Prozess gleich ausschauen. Formal definiert werden kann diese Relation mit Hilfe der *Sicht* eines Prozesses (die im Wesentlichem dem kausalen Kegel aus [BZM14] entspricht), die in Abb. 3 grün dargestellt ist.

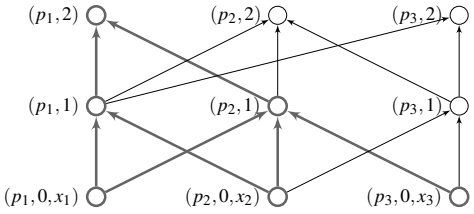


Abb. 3: Der Prozess-Zeit Graph PT^t für Runde $t = 2$ und die $n = 3$ Prozesse p_1, p_2, p_3 mit den Eingangswerten x_1, x_2, x_3. Er stellt den Eingangswert x von Prozess p als Knoten $(p, 0, x)$ dar und enthält eine Kante $((p,r,x),(q,r+1))$ für $r = 0$ bzw. $((p,r),(q,r+1))$ für $r > 0$ genau dann wenn $(p,q) \in G_{r+1}$. Die Sicht von p_1 am Ende von Runde 2 ist der grün markierte Teilgraph.

Die grundlegende Idee der Charakterisierung basiert auf der eher banalen Erkenntnis, dass der Kern eines zulässiges Kommunikationsmuster nicht für immer leer bleiben kann. Diese Erkenntnis folgt leicht aus der Definition des Consensus Problem mit Hilfe der Techniken von [FLP85] bzw. [SW89]. Nun ist die Existenz eines nicht trivialen Kerns zwar notwendig, aber keineswegs hinreichend, damit ein Prozess eine Entscheidung treffen kann. Klarerweise muss ein Prozess auch die Möglichkeit haben, festzustellen, ob der Kern tatsächlich nicht leer ist. Eine zusätzliche Komplikation ergibt sich durch die Eigenheit, dass es durchaus Präfixe von Kommunikationsmustern σ, ρ geben kann, sodass $\mathrm{Ker}(\sigma) = K$, während $\mathrm{Ker}(\rho) = K'$ und ρ ist für einen Prozess p ununterscheidbar von σ. Auf welchen Wert soll nun p entscheiden, wenn nun $K \cap K' = \emptyset$ ist und alle Prozesse aus K Eingangswert 0 haben, während alle Prozesse aus K' Eingangswert 1 haben? Man kann dieses Argument weiterführen und beweisen dass jedes Element der Äquivalenzklasse der transitiven Hülle der Ununterscheidbarkeitsrelation einen gemeinsamen Prozess im Kern haben muss.

Angenommen, wir haben nun einen Nachrichtengegner, der diese Bedingung erfüllt, also jedes zulässige Kommunikationsmuster irgendwann eine nicht triviale Durchschnittsmenge der Kerne der Elemente seiner Äquivalenzklasse besitzt, können die Prozesse hier

Consensus lösen? Die Antwort ist, dass es sogar einen Algorithmus gibt, der hier simultanen Consensus löst, bei dem alle Prozesse gleichzeitig entscheiden. Alles was ein Prozess tun muss, ist die Äquivalenzklasse des aktuellen Präfixes zu berechnen und zu überprüfen, ob deren Kern eine nicht triviale Durchschnittsmenge besitzt. Sobald das eintritt, kann der Prozess auf den Eingangswert eines bestimmten Prozesses (z.B. jenen mit der größten Identifikationsnummer) aus dieser Durchschnittsmenge entscheiden. Die Konstruktion der Äquivalenzklasse stellt sicher, dass alle Prozesse diese Durchschnittsmenge finden.

Das folgende Theorem gibt eine detailliertere, vollständige Charakterisierung der Lösbarkeit von Consensus für abgeschlossene Nachrichtengegner. Wir benutzen hier die Konvention $\sigma|_r$ für den Präfix, bestehend aus den ersten r Runden von σ. Weiters bezeichnen wir mit $[\sigma|_r]$ die Äquivalenzklasse von $\sigma|_r$ bezüglich der transitiven Hülle der Ununterscheidbarkeitsrelation.

Theorem 2 (vgl. [Wi19], Theorem 5.0.1)**.** *Consensus ist lösbar unter einem abgeschlossenen Nachrichtengegner dann und nur dann wenn es für jedes zulässige Kommunikationsmuster σ eine Runde r gibt sodass $\bigcap_{\rho\in[\sigma|_r]} \mathrm{Ker}(\rho) \neq \emptyset$.*

3 Stabilisierende Nachrichtengegner

Wie bereits angedeutet, ist eine der gravierendsten Limitationen der abgeschlossenen Nachrichtengegner dass sie keine Garantien der Form “irgendwann passiert etwas Gutes”, also Garantien, die zu einem finiten aber unbekannten Zeitpunkt eintreten, ausdrücken können. Für Systeme mit chaotischen Startsequenzen, bei der es eine unbekannte Zeit dauert, bis sich eine verlässliche synchrone Kommunikation etabliert hat, sowie in Systemen, in denen massive transiente Fehler von unbekannter Dauer auftreten können, wäre eine Modellierung solcher Eigenschaften jedoch äußerst nützlich. Aus diesem Grund beschreiben wir nun einen Nachrichtengegner, der eine generische Form von Stabilisierung ausdrücken kann und führen eine präzise Charakterisierung der Lösbarkeit von Consensus bezüglich der Dauer dieser Stabilität ein.

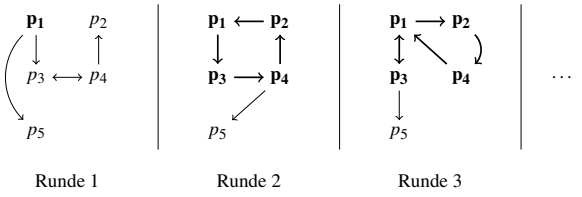


Abb. 4: Die Wurzelkomponente von Runde 1 besteht aus dem Knoten p_1. In Runde 2 und 3 existiert eine knotenstabile Wurzelkomponente, bestehend aus der Knotenmenge $\{p_1, p_2, p_3, p_4\}$.

Er basiert auf dem Konzept einer sogenannten *Wurzelkomponente* (siehe Abb. 4), einer starken Zusammenhangskomponente, bei der kein Knoten eine eingehende Kante von außerhalb der Komponente hat. Eine Wurzelkomponente wird *knotenstabil* genannt, wenn sie für mehrere aufeinanderfolgende Runden eines Kommunikationsmusters aus der selben Knotenmenge besteht. Dabei ist es erlaubt, dass sich die Konnektivität innerhalb der

Komponente stark ändert, sie muss jedoch stets eine Wurzelkomponente des entsprechenden Kommunikationsgraphen bleiben.

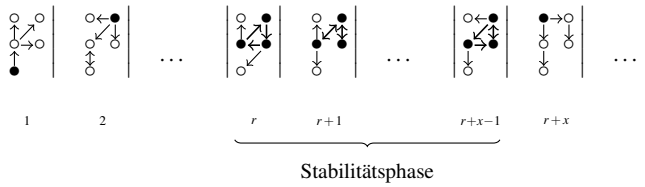


Abb. 5: Ein zulässiges Kommunikationsmuster des Nachrichtengegners $\Diamond\text{STABIL}(x)$. Die knotenstabile Wurzelkomponente (schwarz markiert) taucht in Runde r auf und dauert bis Runde $r+x-1$.

Vermutlich der grundlegendste Nachrichtengegner, basierend auf knotenstabilen Wurzelkomponenten, ist $\Diamond\text{STABIL}(x)$ und wird exemplarisch dargestellt in Abb. 5. Er unterliegt zwei Einschränkungen: 1. Jeder Kommunikationsgraph besitzt genau eine Wurzelkomponente und 2. Jedes zulässige Kommunikationsmuster besitzt eine Wurzelkomponente, die für mindestens x Runden knotenstabil bleibt.

Die zentrale Frage bezüglich dem Nachrichtengegner $\Diamond\text{STABIL}(x)$ ist, wie sich der Parameter x auf die Lösbarkeit von Consensus auswirkt. Wir fassen die Antwort auf diese Frage in Theorem 3 zusammen und geben im Folgenden eine intuitive Begründung. Wie in Abb. 5 skizziert, drückt der Parameter x die Dauer der Stabilitätsphase aus.

Theorem 3 (vgl. [Wi19], Theorem 6.3.6 und 8.4.4)**.** *In einem System aus n Prozessen ist Consensus lösbar unter* $\Diamond\text{STABIL}(x)$ *dann und nur dann wenn* $x \geq n$.

Der Grund, warum Consensus unlösbar ist, wenn $x < n$, ist im Wesentlichen, dass nach $n-1$ Runden Stabilitätsphase der Ausgangswert noch nicht feststehen kann. Die Ursache hierfür wird mit Hilfe von Abb. 6 im Folgenden illustriert:

G_1: $p_1 \longrightarrow p_2 \longrightarrow \cdots \longrightarrow p_n$ $\quad$ G_2: $p_1 \longleftrightarrow p_2 \longrightarrow \cdots \longrightarrow p_n$ $\quad$ G_3: $p_2 \longleftrightarrow p_1 \longrightarrow \cdots \longrightarrow p_n$ $\quad$ G_4: $p_2 \longrightarrow p_1 \longrightarrow \cdots \longrightarrow p_n$

Abb. 6: Kommunikationsgraphen für die Unmöglichkeit von Consensus mit $x < n$.

Nehmen wir an, dass p_1 mit Eingangswert 0 und alle anderen Prozesse mit Eingangswert 1 starten und, um einen Widerspruch abzuleiten, nehmen wir an, dass nach $n-1$ Runden Stabilitätsphase der Ausgangswert bereits feststünde. Wenn wir nun als Kommunikationsmuster G_1^{n-1}, also den Kommunikationsgraph G_1 aus Abb. 6, $n-1$ mal wiederholt, wählen, ist klar dass der Ausgangswert hier auf 1 festgelegt werden muss. Der Grund dafür ist schlicht dass der Prozess p_1 nur seinen eigenen Eingangswert gehört hat und deswegen, wenn er gezwungen wird, sich festzulegen, sich auf 1 festlegen muss (es könnte ja sein, dass alle anderen Prozesse auch 1 als Eingangswert hatten und hier wäre, aufgrund der Consensus Spezifikation, nur 1 ein zulässiger Ausgangswert).

Wenn wir uns nun weiters ansehen, was p_n über das Kommunikationsmuster gelernt hat, ist klar dass G_1^{n-1} für p_n ununterscheidbar ist von G_2^{n-1}. Deshalb muss auch in G_2^{n-1} der Ausgangswert auf 1 festgelegt sein. Ein ganz analoges Argument für G_4^{n-1} und G_3^{n-1} zeigt,

dass der Ausgangswert in G_3^{n-1} auf 0 festgelegt ist. Dies ist aber ein Widerspruch da G_2^{n-1} und G_3^{n-1} ununterscheidbar für p_1 und p_2 ist, wie man leicht überprüfen kann.

Ein klassisches Bivalenz-Argument, wie es bereits in [FLP85, SW89] eingeführt wurde, kann benutzt werden um zu zeigen, dass auch später die Entscheidungen nicht mehr festgelegt werden können.

Um zu zeigen, dass n Runden Stabilitätsphase hinreichend für die Lösbarkeit von Consensus sind, skizzieren wir im Folgenden einen Lösungsalgorithmus. Eine der größten Herausforderungen beim Entwickeln eines solchen Algorithmus ist, die kurze Dauer der Stabilitätsphase von nur n Runden effizient auszunutzen. Die Grundoperation unseres vorgeschlagenen Algorithmus ist es, in Runde r jedem möglichen Präfix σ einen Wert $\Delta(\sigma)$ zuzuordnen. Sobald in einer Runde r alle Präfixe, die für einen Prozess p in Betracht kommen (also jene, die für p vom tatsächlichen Präfix ununterscheidbar sind), denselben Wert Δ zugewiesen bekommen haben, entscheidet p auf den Wert von Δ. Die Berechnung von $\Delta(\sigma)$ selbst in Runde r erfolgt nach folgenden Regeln:

- Falls σ bereits einen Wert zugewiesen bekommen hat, übernimm diesen Wert.
- Ansonsten, weise $\Delta(\sigma)$ einen neuen Wert v zu wenn
  - es einen Präfix τ mit zugewiesenem Wert $\Delta(\tau) = v$ gibt, sodass die Stabilitätsphase von τ mindestens $2n-2$ Runden zurückliegt oder
  - die Stabilitätsphase von σ selbst $2n-2$ Runden zurückliegt und v der Eingangswert des Prozesses mit der größten Identifikationsnummer in der knotenstabilen Wurzelkomponente während dieser Stabilitätsphase ist.

4 Allgemeine Nachrichtengegner

Nach den bisher präsentierten Lösbarkeitscharakterisierungen für abgeschlossene Nachrichtengegner und den Nachrichtengegner $\Diamond\text{STABIL}(x)$, stellt sich die Frage, ob es auch eine solche Charakterisierung für allgemeine Nachrichtengegner gibt, die also sowohl abgeschlossene als auch nicht abgeschlossene Nachrichtengegner beinhaltet. Ein Weg, der zu einer solchen Charakterisierung führt, ist es, den topologischen Raum der Kommunikationsmuster zu betrachten, ganz analog zur Herangehensweise im klassischen Resultat von Alpern und Schneider [AS85]. Als Punkte im topologischen Raum eines Nachrichtengegners wählen wir im Wesentlichen[4] die Prozess-Zeit Graphen aus Abb. 3, die die Eingangswerte zusammen mit den Kommunikationsmustern des Nachrichtengegners beinhalten.

Wir definieren die Topologie der Punktmenge eines Nachrichtengegners mit Hilfe einer Distanzfunktion, die die "Nähe" von zwei Prozess-Zeit Graphen α, β quantifizieren soll. Intuitiv gesprochen bringen wir damit zum Ausdruck, wie ähnlich sich α, β sind im Bezug auf die Sicht einzelner Prozesse, also wie leicht bzw. schwer es die Prozesse haben α von β zu unterscheiden. Das macht Sinn, da der Extremfall der ewigen Ununterscheidbarkeit

[4] Aus technischen Gründen wählen wir genaugenommen die infiniten Produkte $PT^1 \times PT^2 \times \cdots$.

von α, β bedeutet dass in beiden Prozess-Zeit Graphen irgendwann dieselbe Entscheidung getroffen wird. Die folgenden Distanzfunktionen (vgl. Abb. 7), geben dieser nützlichen Einsicht gewissermaßen eine noch feinere Granularität, die es uns erlauben wird, eine Lösbarkeitscharaktscharakterisierung zu formulieren. Sei $d_p(\alpha, \beta) = 2^{-t}$ wobei t die letzte Runde ist sodass α und β für den Prozess p noch ununterscheidbar sind und sei $d_{\min}(\alpha, \beta) = \min_{p \in \Pi} d_p(\alpha, \beta)$, wobei Π für die Menge aller Prozesse im System steht.

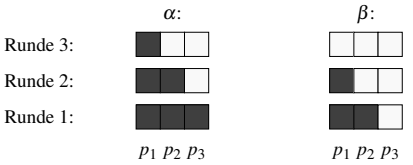


Abb. 7: Distanzfunktion für ein System aus den drei Prozessen $\{p_1, p_2, p_3\}$ mit den Prozess-Zeit Graphen α, β. Wir benutzen blau bzw. gelb für zwei verschiedene Sichten eines Prozesses. So hat p_3 bereits in Runde 1 eine unterschiedliche Sicht, weshalb $d_{p_3}(\alpha, \beta) = 2^0 = 1$. Analog sieht p_2 keinen Unterschied bis Runde 1 und $d_{p_2}(\alpha, \beta) = 2^{-1} = 1/2$ und p_1 sieht keinen Unterschied bis Runde 2, weshalb $d_{p_1}(\alpha, \beta) = 2^{-2} = 1/4$. Die minimale Distanzfunktion ist also $d_{\min}(\alpha, \beta) = \min_{p \in \{p_1, p_2, p_3\}} d_p = d_{p_1} = 1/4$.

Die Lösbarkeitscharakterisierung von Consensus für allgemeine Nachrichtengegner ist im folgenden Theorem 4 formuliert. Sie benutzt eine Handvoll grundlegender topologischer Begriffe, wie den topologischen Raum, der durch $d_{\min}$ induziert wird: Wenn X die Menge der Punkte ist, ist damit X gemeint, zusammen mit der Familie von *offenen* Teilmengen $U \subseteq X$, definiert durch $\forall x \in U \; \exists \varepsilon > 0 : B_\varepsilon(\alpha) \subseteq U$. Hierin bezeichnet $B_\varepsilon(\alpha)$ den offenen ε-Ball um α bzgl. der Distanzfunktion $d_{\min}$, der definiert ist als $B_\varepsilon(\alpha) = \{\beta : d_{\min}(\alpha, \beta) < \varepsilon\}$. Ein topologischer Raum ist *nicht zusammenhängend* wenn er in mehrere (nicht leere) offene Mengen partitioniert werden kann. Eine *Zusammenhangskomponente* eines topologischen Raumes ist eine maximale offene zusammenhängende Menge. Sei $PT(v)$ die Menge der Prozess-Zeit Graphen, in denen alle Prozesse Eingangswert v haben. Dann gilt:

Theorem 4. *Consensus ist lösbar unter einem Nachrichtengegner dann und nur dann wenn keine Zusammenhangskomponente des Raumes der Prozess-Zeit Graphen, der durch $d_{\min}$ induziert wird, sowohl $PT(0)$ als auch $PT(1)$ schneidet.*

Diese Charakterisierung sagt aus, dass, um eine Lösung von Consensus zu ermöglichen, der topologische Raum, der durch $d_{\min}$ induziert wird, in keiner Zusammenhangskomponente sowohl einen Prozess-Zeit Graph aus $PT(0)$ (also mit allen Eingangswerten auf 0 gesetzt) als auch einer aus $PT(1)$ auftauchen kann (siehe auch Abb. 8).

Wir erklären nun kurz, warum die Bedingung aus Theorem 4 nötig ist, damit Consensus gelöst werden kann. Um einen Widerspruch herzuleiten, nehmen wir also an, dass es eine Komponente S gibt, die sowohl $PT(0)$ als auch $PT(1)$ schneidet. Zunächst statten wir die Menge der Ausgangswerte $\{0, 1\}$ mit der diskreten Topologie aus, in der sowohl 0 als auch 1 offen sind. Man kann aus der Consensus Spezifikation folgern, dass die Funktion Δ, die jeden Prozess-Zeit Graphen aus S auf ein Element dieses topologischen Raums über

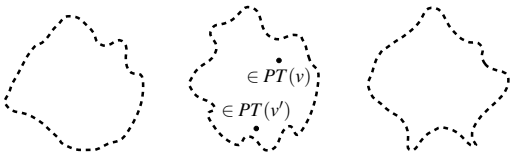


Abb. 8: Wenn eine Zusammenhangskomponente sowohl $PT(v)$ als auch $PT(v')$ schneidet, dann ist Consensus lösbar genau dann wenn $v = v'$.

$0,1$ abbildet, lokal konstant und deshalb stetig ist. Aufgrund der Stetigkeit von Δ ist das Bild von Δ zusammenhängend, jedoch folgt aus der Consensus Spezifikation, dass der Bildraum von Δ sowohl 0, als auch 1 beinhaltet. Dies ist in der diskreten Topologie jedoch nicht zusammenhängend, was obiger Aussage widerspricht.

Um zu sehen, dass die Erfüllung der Bedingung aus Theorem 4 ausreicht, um Consensus zu lösen, betrachten wir folgenden Lösungsalgorithmus. Wir nehmen also an, es existiert eine Partition P des topologischen Raumes, welche die Bedingung aus Theorem 4 erfüllt. Basierend auf jenen Prozess-Zeit Graphen, die ein Prozess p in Betracht zieht, berechnet er in jeder Runde t den ε-Ball bezüglich der d_p-Metrik, also $B_{\varepsilon,d_p}(\alpha) = \{\beta : d_p(\alpha,\beta) < \varepsilon = 2^{-t}\}$. Sobald $B_{\varepsilon,d_p}(\alpha)$ ganz in einer Menge der Partition P liegt, kann p auf einen Wert entscheiden: Wenn diese Menge $PT(v)$ beinhaltet, entscheidet p auf v, sonst auf 0.

5 Schlussfolgerungen

Wir haben hier kurz die wichtigsten der in [Wi19] vorgestellten Consensus-Charakterisierungen präsentiert. Wir haben gesehen, dass es für abgeschlossene Nachrichtengegner eine relativ einfache Charakterisierung gibt, während eine allgemeine Charakterisierung nur mit Hilfe topologischer Methoden erlangt werden konnte. Dazwischen liegt die Charakterisierung für stabilisierende Nachrichtengegner, die zwar nicht so allgemein ist wie letztere, jedoch ohne topologische Methoden auskommt. Wir hoffen, dass diese Resultate nützlich sind, um zu erklären wann bzw. warum Consensus in einem System, das durch einen Nachrichtengegner modelliert wird, (nicht) gelöst werden kann.

Literaturverzeichnis

[AS85] Alpern, Bowen; Schneider, Fred B.: Defining Liveness. Information Processing Letters, 21(4):181–185, 1985.

[Bi15] Biely, Martin; Robinson, Peter; Schmid, Ulrich; Schwarz, Manfred; Winkler, Kyrill: Gracefully Degrading Consensus and *k*-Set Agreement in Directed Dynamic Networks. In: NETYS'15. LNCS 9466. Springer, S. 109–124, 2015.

[Bi18] Biely, Martin; Robinson, Peter; Schmid, Ulrich; Schwarz, Manfred; Winkler, Kyrill: Gracefully degrading consensus and k-set agreement in directed dynamic networks. Theoretical Computer Science, 726:41–77, 2018.

[BZM14] Ben-Zvi, Ido; Moses, Yoram: Beyond Lamport's Happened-before: On Time Bounds and the Ordering of Events in Distributed Systems. J. ACM, 61(2):13:1–13:26, April 2014.

[CBS09] Charron-Bost, Bernadette; Schiper, André: The Heard-Of model: computing in distributed systems with benign faults. Distributed Computing, 22(1):49–71, April 2009.

[CGP15] Coulouma, Étienne; Godard, Emmanuel; Peters, Joseph G.: A characterization of oblivious message adversaries for which Consensus is solvable. Theoretical Computer Science, 584:80–90, 2015.

[FG11] Fevat, Tristan; Godard, Emmanuel: Minimal Obstructions for the Coordinated Attack Problem and Beyond. In: IPDPS'11. S. 1001–1011, 2011.

[FLP85] Fischer, Michael J.; Lynch, Nancy A.; Paterson, M. S.: Impossibility of Distributed Consensus with one Faulty Process. J. ACM, 32(2):374–382, April 1985.

[LM95] Lubitch, Ronit; Moran, Shlomo: Closed Schedulers: A Novel Technique for Analyzing Asynchronous Protocols. Distributed Computing, 8(4):203–210, Juni 1995.

[NSW19] Nowak, Thomas; Schmid, Ulrich; Winkler, Kyrill: Topological Characterization of Consensus Under General Message Adversaries. PODC'19. ACM, S. 218–227, 2019.

[Sc14] Schwarz, Manfred; Winkler, Kyrill; Schmid, Ulrich; Biely, Martin; Robinson, Peter: Brief Announcement: Gracefully Degrading Consensus and k-set Agreement under Dynamic Link Failures. PODC'14. ACM, S. 341–343, 2014.

[SSW18] Schmid, Ulrich; Schwarz, Manfred; Winkler, Kyrill: On the Strongest Message Adversary for Consensus in Directed Dynamic Networks. SIROCCO'18. Springer, S. 102–120, 2018.

[SW89] Santoro, Nicola; Widmayer, Peter: Time is not a healer. In: STACS'89. LNCS 349. Springer, S. 304–313, Februar 1989.

[SWS16] Schwarz, Manfred; Winkler, Kyrill; Schmid, Ulrich: Fast Consensus Under Eventually Stabilizing Message Adversaries. ICDCN'16. ACM, S. 7:1–7:10, 2016.

[Wi19] Winkler, Kyrill: Characterization of Consensus Solvability under Message Adversaries. Dissertation, TU Wien, 2019. `http://katalog.ub.tuwien.ac.at/AC15506161`.

[WS19] Winkler, Kyrill; Schmid, Ulrich: An Overview of Recent Results for Consensus in Directed Dynamic Networks. Bulletin of the EATCS, 128, 2019.

[WSM19] Winkler, Kyrill; Schmid, Ulrich; Moses, Yoram: A Characterization of Consensus Solvability for Closed Message Adversaries. OPODIS'19, 2019. (to appear).

[WSS19] Winkler, Kyrill; Schwarz, Manfred; Schmid, Ulrich: Consensus in rooted dynamic networks with short-lived stability. Distributed Computing, 32(5):443–458, Oct 2019.

Kyrill Winkler studierte Informatik an der technischen Universität Wien und promovierte im Jahr 2019 bei Prof. Ulrich Schmid mit Auszeichnung. Er absolvierte einen mehrmonatigen Forschungsaufenthalt an der Ecole Polytechnique in Paris und ist aktuell Universitätsassistent am Institut für Computer Engineering der TU Wien. Sein Forschungsschwerpunkt sind verteilte Algorithmen, insbesondere die Analyse von Consensus und verwandten Problemen in dynamischen Netzen.