Eigenschaften eines Algorithmus zur Zuordnung von nicht eindeutigen Sende- und Empfangsereignissen in Ereignisprotokollen

Bachelorarbeit

von

Benito van der Zander

aus

Düsseldorf

vorgelegt bei der

Arbeitsgruppe für Mobile und Dezentrale Netzwerke

Jun.-Prof. Dr. Björn Scheuermann

Heinrich-Heine-Universität Düsseldorf

August 2009

Betreuer:

Dr. Wolfgang Kiess

Jun.-Prof. Dr. Björn Scheuermann

Danksagung

Für ihre Unterstützung und das viele Lesen möchte ich hier vor allem meinen beiden Betreuern danken.

Inhaltsverzeichnis

Kapitel 1.

Einführung

In jedem Netzwerk oder verteilten System existieren gleichzeitig viele Prozesse in unterschiedlichen Knoten, die im wesentlichen voneinander unabhängig arbeiten. Einige dieser Prozesse kommunizieren jedoch untereinander, so dass der Ablauf der Prozesse in manchen Knoten von anderen Knoten geändert wird. Häufig muss man nun die globale Ereignisreihenfolge, also wann welche Knoten andere beeinflusst haben könnten, ermitteln.

Eine einfache Lösung zur Bestimmung des vollständigen Ablaufs scheint es zu sein, in jedem Knoten eine Logdatei zu führen, in der alle empfangenen und gesendeten Nachrichten verzeichnet werden. Doch man erkennt schnell, dass solche Logs alleine nicht ausreichen, um den globalen Ablauf nachvollziehen zu können. Denn es ist nicht trivial, die in Sende- und Empfangslog aufgezeichneten Nachrichten einander zuzuordnen, da viele der Nachrichten in einem LAN-Segment (wie zum Beispiel ARP-Requests [Che08]) nicht eindeutig sind. Außerdem können gesendete Nachrichten während der Übertragung verloren gehen, so dass sie nicht in jedem Log aufgezeichnet werden, wodurch für ein Empfangsereignis mehrere mögliche Sendeereignisse existieren können.

Eine vermeintlich einfache Lösung wäre es jedes Paket mit einer eindeutigen ID zu versehen, wie sie beispielsweise bei einem zuverlässigen Übertragungsprotokoll verwendet wird.

Dieser Ansatz ist jedoch häufig nicht praktikabel, denn er erfordert, die kommunizierenden Prozesse umzuprogrammieren und eventuell alte Experimente zu wiederholen. Wünschenswert wäre ein Verfahren mit dem sich eine eindeutige Zuordnung zwischen Sende- und Empfangsereignissen nachträglich aus unvollständigen Aufzeichnungen berechnen ließe, ohne dass man weitere Informationen benötigt.

Im Rahmen dieser Arbeit wird zuerst dieses Problem der Zuordnungen formal spezifiziert und auf seine Komplexität untersucht. Danach werden zwei Algorithmen zu dessen Lösung vorgestellt, eine schnelle, jedoch unzuverlässige Heuristik, sowie ein perfekt arbeitendes Verfahren mit entsprechender Laufzeit.

Kapitel 2.

Problemstellung

Man benötigt zuerst eine formale Darstellung für die aufgezeichneten Logdateien, sowie für die zu ermittelnde Zuordnung zwischen den Ereignissen. Hierzu werden im wesentlichen die Definitionen aus [SK09] übernommen, wo das Problem zum ersten Mal vorgestellt wurde.

Die einzelnen Knoten werden von 0 bis n_L durchnummeriert[1] und das Log eines Knoten i wird als Menge L_i von Ereignissen dargestellt. Zur Vereinfachung der Notation werden alle Ereignisse in einer Menge $L = \bigcup L_i$ zusammengefasst und eine Funktion $N : L \to \mathbb{N}$ definiert, die jedem Ereignis die Nummer der Logdatei zuweist, in der das Ereignis gespeichert ist, also $e \mapsto l$ für $e \in L_l$.

Jedes Ereignis ist dabei entweder ein Sende- oder ein Empfangsereignis, je nachdem ob der Knoten damit das Senden oder den Empfang einer Nachricht in der Logdatei aufgezeichnet hat. Diese Zuordnung wird als Funktion $D : L \to \{\tilde{s}, \tilde{r}\}$ dargestellt, wobei $\tilde{s}$ für »Senden« und $\tilde{r}$ für »Empfangen« steht.

Damit kann man die Menge aller Sendeereignisse $S = \{s \in L | D(s) = \tilde{s}\}$ und die Menge aller Empfangsereignisse $R = \{r \in L | D(r) = \tilde{r}\}$ definieren. Außerdem erweist es sich als

[1] Die 0-basierte Nummerierung ist sehr gut geeignet um ein bestimmtes Log L_0 von den anderen zu unterscheiden. In einigen späteren Kapiteln über die Algorithmen ist dies jedoch nicht nötig, so dass dort die intuitivere, bei 1 beginnende Nummerierung verwendet wird, indem O.B.d.A. $L_0 = \emptyset$ vorausgesetzt wird.

eine nützliche Vereinfachung der Notation für jedes Log L_l die Menge aller enthaltenen Sendeereignisse als $S_l = S \cap L_l$ und die Menge aller enthaltenen Empfangsereignisse als $R_l = R \cap L_l$ zu bezeichnen.

Um die Ereignisse in den Logdateien vergleichen zu können, wird jedem Ereignis mittels einer Funktion $P : L \to \mathscr{P}$ ein Pakettyp zugeordnet. Diese Pakettypen bilden die Menge $\mathscr{P}$ und müssen so gewählt sein, dass Ereignissen, die identische Nachrichten betreffen, der gleiche Pakettyp zugeordnet werden kann und unterschiedliche Nachrichten unterschiedliche Pakettypen erhalten. Dies lässt sich in einer Implementierung beispielsweise durch einen guten Hash des gesamten Paketinhaltes inklusive Header erreichen.

In jedem Log ist zudem implizit die Reihenfolge der Ereignisse gespeichert, was durch eine Ordnungsrelation $\prec_l$ für jedes Log L_l dargestellt wird, so dass $e \prec_l f$ für $e, f \in L_l$ gilt, falls e vor f aufgezeichnet wird. Für Ereignisse, die nicht im Log L_l sind, ist $\prec_l$ selbstverständlich undefiniert. Gilt zudem für zwei Ereignisse e, f entweder $e = f$ oder $e \prec_l f$, so wird dies zur Vereinfachung als $e \preceq_l f$ geschrieben.

Es lohnt sich hier auch eine anschauliche Darstellung der Logs einzuführen, die in den späteren Kapiteln vereinzelt verwendet wird. Dabei stellt man die Sendeereignisse durch die Großbuchstaben A, B, C, ... dar und die Empfangsereignisse durch die Kleinbuchstaben a, b, c, ..., wobei Ereignisse mit gleichen Buchstaben denselben Typ besitzen (also $P(a) = P(A)$ für Ereignisse a und A). Jedes Log wird nun als eine Reihe von solchen als Buchstaben kodierten Ereignissen dargestellt, die in der Reihenfolge ihrer Aufzeichnung notiert werden, und alle Logs werden dann untereinander geschrieben.

Eine Zuordnung wird dann gemäß [SK09] durch eine Abbildung $A : L \to S$ dargestellt, die jedem Empfangsereignis ein Sendeereignis und jedes Sendeereignis sich selbst zuordnet:

Definition 2.1. *Eine Abbildung $A : L \to S$ wird als globale Zuordnung (GA) bezeichnet, wenn gilt:*

1. $\forall s \in S : A(s) = s$
   Jedes Sendeereignis ist sich selbst zugeordnet

2. $\forall e \in L : P(e) = P(A(e))$
   Alle einander zugeordneten Ereignisse haben denselben Nachrichtentyp

3. $\forall r \in R : N(r) \neq N(A(r))$
   Kein Knoten empfängt seine eigenen Nachrichten

Mit dieser Definition alleine besitzt man aber noch keine Möglichkeit zu entscheiden, ob die durch die Zuordnung festgelegten Übertragungsvorgänge tatsächlich in einem Experiment aufgetreten sein könnten. Deswegen wird in [SK09] nun eine formale Darstellung für die globale, zeitliche Reihenfolge der Ereignisse eingeführt:

Die globale Transmissionordnung (GTO), eine irreflexive, partielle Ordnungsrelation der Sendeereignisse $\prec \subset S \times S$.

Damit kann man nun eine Konsistenzbedingung einführen, welche garantiert, dass die durch die GA gegebene Zuordnung zusammen mit der durch die GTO gegebene Reihenfolge nicht den in den Logs aufgezeichneten Reihenfolgen widerspricht:

Definition 2.2. *Ein Paar $(\prec, A)$ heißt konsistent, wenn gilt:*

$$\forall n \in N : \forall e_1, e_2 \in L_n : (e_1 \prec_n e_2 \Rightarrow A(e_1) \prec A(e_2))$$

Diese Definition ist deswegen sinnvoll, weil wir sämtliche Übertragungsverzögerungen ignorieren. Das heißt wir gehen davon aus, dass e und das zugeordnete Sendeereignis $A(e)$ immer gleichzeitig geschehen, weshalb aus $e_1 \prec_n e_2$ eben $A(e_1) \prec A(e_2)$ folgen muss.

Definition 2.3. *Eine GA A heißt konsistent, wenn es eine GTO $\prec$ gibt, so dass $(\prec, A)$ konsistent ist.*

Diese Definition macht Sinn, wenn man bedenkt, dass die Logdateien dadurch entstanden sind, dass die aufgezeichneten Nachrichten in einer bestimmten (durch die GTO ermittelten) Reihenfolge gesendet wurden und einander (durch die GA) zugeordnete Ereignisse gleichzeitig von ihren jeweiligen Knoten zu den Logs hinzugefügt wurden.

Man sieht auch, dass man zu einer gegebenen konsistenten GA A in Linearzeit eine GTO $\prec$ finden kann, so dass das Paar $(\prec, A)$ konsistent ist, indem man einfach mittels topologischem Sortieren ([Las61]) in Linearzeit eine Ordnungsrelation bestimmt, welche die für jedes Log durch die Konsistenzbedingung gegebenen Ordnungsrelationen enthält.

Damit ist die Formalisierung des Problems in seiner allgemeinsten Form abgeschlossen. Im Folgenden werden aber auch einige wichtige Spezialfälle betrachtet, für die nun zusätzliche Einschränkungen erforderlich sind.

Bei realen Experimenten tritt häufig der Spezialfall auf, dass Nachrichten eines bestimmten Typs nur von genau einem Knoten ausgesandt werden können, da üblicherweise jede gesendete Nachricht die eindeutige ID des Senders enthält. Wird beispielsweise Ethernet verwendet, ist diese Adresse die MAC-Adresse, die weltweit eindeutig sein sollte[2].

In [SK09] wird dieser Spezialfall durch folgende Einschränkung spezifiziert:

$$\forall s_1, s_2 \in S : (P(s_1) = P(s_2) \Rightarrow N(s_1) = N(s_2))$$

Damit wird die dritte Bedingung einer GA ($N(r) \neq N(A(r))$ für alle $r \in R$) im Grunde überflüssig, da je nach Struktur der Logs entweder jede oder keine »GA«, welche die ersten beiden Bedingungen erfüllt, die dritte erfüllt. Formaler ausgedrückt: Eine beliebige Abbildung $A : L \to S$, welche die ersten beiden Bedingungen einer GA erfüllt, ist in diesem Fall genau dann eine GA, wenn es keine $r \in R_i$, $s \in S_i$ gibt mit $P(r) = P(s)$ (denn aus $N(r) = N(A(r)) = i$ würde folgen $r \in R_i$ und $s = A(r) \in S_i$ mit $P(s) = P(r)$).

Bei der theoretische Analyse erweist es sich als nützlich, die Menge der möglichen Lösungen durch eine zusätzliche Anforderung einzuschränken, nämlich durch die Bedingung, dass jedem Sendeereignis maximal ein Empfangsereignis zugeordnet werden darf. In der Praxis könnte sich diese Situation beispielsweise bei reinen Punkt-zu-Punkt Verbindungen ergeben.

Hierzu definieren wir:

[2] Praktisch gibt es zwar Adresskollisionen, aber in einem Labornetzwerk lassen sie sich leicht beheben.

Definition 2.4. *Eine GA A heißt eindeutig, wenn jedes Sendeereignis nur ein einziges Mal empfangen wird, also wenn gilt:*

$$r_1, r_2 \in R : (A(r_1) = A(r_2) \Rightarrow r_1 = r_2)$$

Sind nun bestimmte Logdateien gegeben, gibt es verschiedene Fragestellungen über die möglichen Zuordnungen. Die im folgenden behandelten Problemstellungen sind:

SRC: Kann eine konsistente GA A gefunden werden?

uSRC: Kann eine konsistente und eindeutige GA A gefunden werden?

pSRC: Was ist für ein gegebene Ereignis $r \in R$ die Menge der möglichen Sendeereignisse $\{s \in S | \exists \text{konsistente GA } A : A(r) = s\}$?

Die drei Abkürzungen bedeuten jeweils »Send-Receive-Correlation«, »Unique-Send-Receive-Correlation« und »Possible-Send-Receive-Correlations«. Außerdem betrachten wir nur eine konkrete GA A als Lösung für SRC und uSRC, nicht das bloße Erkennen der Existenz einer solchen. Bei all diesen Problemen behandeln wir den Spezialfall, dass gleiche Pakete nur von demselben Sender stammen können. Für den allgemeinen Fall, dass jeder Knoten Pakete jeden Typs versenden kann, verwenden wir die Bezeichnung SRC*, uSRC* und pSRC*.

pSRC und pSRC* wurden erstmalig in [SK09] untersucht, allerdings wurden die Probleme dort noch als SRC bezeichnet, während in dieser Arbeit die Bezeichnung SRC für den einfachsten Fall der Suche nach einer konsistenten GA reserviert wurde.

Im nächsten Kapitel wird für diese Fragen bewiesen, dass sie NP-vollständig sind. Anschließend wird eine schnelle Heuristik sowie ein langsamer, exakter Algorithmus zur Beantwortung der Fragen vorgestellt.

Kapitel 3.

Komplexität

3.1. Eindeutige Empfänger und keine Senderambivalenz (uSRC)

In diesem Abschnitt wird das als 3-Exactcover (X3C) bezeichnete Problem, eine exakte Mengenüberdeckung zu ermitteln, auf das Finden einer konsistenten und eindeutigen GA reduziert. X3C ist als NP-vollständig bekannt und lautet folgendermaßen [GJ79]:

Sei eine Menge $M = \{1,\dots,m_T\}$ mit $m_T = 3q$ und $q \in \mathbb{N}$ sowie eine Familie von Untermengen $\{M_i\}_{1\leq i\leq n_L}$ mit $M_i \subseteq M$ und $|M_j| = 3$ gegeben
Existieren $\{T_i\} \subseteq \{M_i\}$, so dass $T_i \cap T_j = \emptyset$ für $i \neq j$ und $\bigcup_i T_i = M$?

Um dieses X3C-Problem nun auf uSRC zu reduzieren, müssen für gegebene Mengen M_j Logs L_j so konstruiert werden, dass jede zu ihnen konsistente und eindeutige GA zu einer Lösung von X3C umgewandelt werden kann. Die Konstruktion der Logs für uSRC erfolgt dabei ähnlich zu der in [Wan09] beschriebenen Reduktion von X3C auf SRC*. Der einzige Unterschied ist dabei, dass hier Empfangs- und Sendeereignisse im Vergleich zu denen in [Wan09] vertauscht sind und eben uSRC anstatt von SRC* betrachtet wird.

Bevor wir die Logs konstruieren können, müssen wir die Menge der möglichen Ereignistypen definieren: Es sei $\mathscr{P} = M \cup \{X\} \subset \mathbb{N}$, wobei X einen Ereignistyp bezeichnet, der nach der üblichen Ordnung über $\mathbb{N}$ größer ist als alle anderen in M (z.B.: $X = m_T + 1$).

Zudem benötigen wir eine Funktion $C : M \rightarrow \mathbb{N}$, die für jedes Element aus M angibt, in wie vielen Mengen M_i es enthalten ist, also $C(k) = |\{M_j | k \in M_j\}|$.

Damit können nun n_L Empfangslogs $L_1, \ldots, L_{n_L}$ folgendermaßen definiert werden: Für jede Menge M_i des X3C-Problems mit $M_i = \{\alpha, \beta, \gamma\}$ und $\alpha < \beta < \gamma$ sei nun

$$L_i = \{a, b, c, x\}$$

ein Log mit vier Empfangsereignissen für die gilt $P(a) = \alpha$, $P(b) = \beta$, $P(c) = \gamma$, $P(x) = X$ und $a \prec_i b \prec_i c \prec_i x$.

Als nächstes wird das Senderlog L_0 definiert, welches für jedes der in den anderen n_L Logs aufgezeichnete Empfangsereignis genau ein passendes Sendeereignis besitzt. Dieses Log L_0 besteht aus vier Teilen, die durch die Symbole ◀, ▲ und ▶ voneinander getrennt werden:

$$L_0 = (L_{0,\text{Teil 1}}, \blacktriangleleft, L_{0,\text{Teil 2}}, \blacktriangle, L_{0,\text{Teil 3}}, \blacktriangleright, L_{0,\text{Teil 4}})$$

Der erste Teil vor ◀ enthält von jedem Typ aus M genau ein Sendeereignis und entspricht der Menge, die von der X3C-Lösung überdeckt werden soll. Schreibt man diesen Teil als Tupel von Ereignistypen in einer festen Reihenfolge, so ergibt sich:

$$P(L_{0,\text{Teil 1}}) = (1, 2, 3, \ldots, m_T)$$

Da jede Menge M_i genau drei Elemente enthält und die Mengen einer Lösung untereinander disjunkt sein müssen, kennt man die Zahl der an einer Lösung beteiligten Mengen: $\frac{m_T}{3}$. Im zweiten Teil, zwischen ◀ und ▲, kann also für jede dieser Mengen ein X gesendet werden, um zu garantieren, dass in jedem Log jeweils allen Ereignissen nur Sendeereignisse vor oder nur nach ▲ zugewiesen werden:

$$P(L_{0,\text{Teil 2}}) = (X, X, \ldots, X) \text{ mit } |L_{0,\text{Teil 2}}| = \frac{m_T}{3}$$

Der dritte Teil, zwischen ▲ und ▶, enthält dann einfach genügend Sendeereignisse von jedem Typ, dass für jedes Empfangsereignis mit einem Typ aus M tatsächlich ein Sendeereignis existiert:

$$P(L_{0,\text{Teil }3}) = (\underbrace{1,1,\ldots,1}_{C(1)-1\text{-mal}}, \quad \underbrace{2,2,\ldots,2}_{C(2)-1\text{-mal}}, \quad \underbrace{3,3,\ldots,3}_{C(3)-1\text{-mal}}, \quad \ldots \quad \underbrace{m_T,m_T,\ldots,m_T}_{C(m_T)-1\text{-mal}})$$

Der vierte Teil, nach ▶, besteht aus den noch fehlenden Sendeereignisse vom Typ X:

$$P(L_{0,\text{Teil }4}) = (X,X,\ldots,X) \text{ mit } |L_{0,\text{Teil }2}| = n_L - \frac{m_T}{3}$$

Alles zusammen ergibt:

$$
\begin{aligned}
P(L_0) = (\quad & 1, \quad 2, \quad \ldots, \quad m_T - 1, \quad m_T, \\
& \qquad\qquad\qquad\qquad\qquad \blacktriangleleft \\
& \underbrace{X \quad X \quad \ldots \quad X}_{\frac{m_T}{3}\text{-mal}} \\
& \qquad\qquad\qquad\qquad\qquad \blacktriangle \\
& \underbrace{1 \quad 1 \quad \ldots \quad 1}_{C(1)-1\text{-mal}} \\
& \underbrace{2 \quad 2 \quad \ldots \quad 2}_{C(2)-1\text{-mal}} \\
& \qquad\quad \vdots \\
& \underbrace{m_T - 1 \quad m_T - 1 \quad \ldots \quad m_T - 1}_{C(m_T-1)-1\text{-mal}} \\
& \underbrace{m_T \quad m_T \quad \ldots \quad m_T}_{C(m_T)-1\text{-mal}} \\
& \qquad\qquad\qquad\qquad\qquad \blacktriangleright \\
& \underbrace{X \quad X \quad \ldots \quad X}_{n_L-\frac{m_T}{3}\text{-mal}}
\end{aligned}
$$

Nach dieser Definition erkennt man leicht, dass jede konsistente Lösung dieselbe GTO $\prec$ haben muss. Offensichtlich ist $S = L_0$ und bei jeder konsistenten Lösung gilt für alle

$e_1, e_2 \in L_0$ mit $e_1 \prec_0 e_2$, dass $A(e_1) \prec A(e_2)$ ist, also $e_1 \prec e_2$, da $e_1, e_2 \in L_0 = S$ sind. Da $\prec \subseteq S \times S$ ist, ist $\prec$ dadurch vollständig definiert und es gilt $\prec = \prec_0$.

Nun ist zu zeigen, dass das Finden einer zu diesen Logs konsistenten und eindeutigen GA äquivalent zum Lösen der gegebenen Instanz von X3C ist.

Lemma 3.1. *Existiert eine Lösung für uSRC, so existiert eine Lösung für X3C.*

Beweis. Dies folgt im wesentlichen daraus, dass es durch die Konstruktion von L_0 kein Log geben kann, dessen Ereignissen sowohl Sendeereignisse vor $\blacktriangle$ als auch nach $\blacktriangle$ zugeordnet sind.

Da zudem durch die geforderte Eindeutigkeit und die beschränkte Zahl der Sendeereignisse jedes Sendeereignis tatsächlich empfangen werden muss, gibt es eine Menge von Logs, die nur Ereignisse vor $\blacktriangle$ empfangen und die zugleich auch für sämtliche Sendeereignisse vor $\blacktriangle$ ein entsprechendes Empfangsereignis besitzen. Das heißt also, dass es eine Bijektion zwischen ihren Empfangs- und den Sendeereignissen vor $\blacktriangle$ gibt. Damit bilden die Mengen M_i, auf denen die entsprechenden Logs basieren, eine Lösung für X3C.

Die mathematischen Details finden sich im Anhang unter A.1. $\square$

Um die Äquivalenz der Probleme zu zeigen, muss das obige Lemma nun auch noch in der Gegenrichtung formuliert werden. Desweiteren werden einige Details der gesuchten GA spezifiziert, die in späteren Lemmas benötigt werden:

Lemma 3.2. *Existiert eine Lösung für X3C, so existiert eine Lösung für uSRC.*

Zudem gibt es zwei Indexmengen I und J mit $I \cap J = \emptyset$ und $I \cup J = \{1, \ldots, n_L\}$, so dass für alle $r \in R_i = L_i$ mit $i \in I$ gilt $A(r) \prec \blacktriangle$ und für alle $r \in R_j = L_j$ mit $j \in J$ gilt $\blacktriangle \prec A(r)$.

Außerdem sind die Ereignisse innerhalb dieser Mengen von Logs nach Typ und jeweiliger Lognummer sortiert, das heißt, ist $k, k' \in I$ mit $k < k'$ oder $k, k' \in J$ mit $k < k'$, so gilt für alle $q \in L_k, r \in L_{k'}$ mit $P(q) = P(r)$, dass $A(q) \prec A(r)$ ist.

Beweis. Dieser Beweis erfolgt im Grunde umgekehrt zum vorherigen Beweis. Da die Mengen der Lösung des Coverproblems von jedem Typ genau ein Element enthalten, kann man jedem Empfangsereignis aus den ihnen zugeordneten Logs eindeutig ein Sendeereignis vor ▲ zuweisen. Allen restlichen Empfangsereignissen kann man dann Sendeereignisse aus L_0 nach ▲ zuordnen.

Wieder finden sich die Details im Anhang A.1. □

Damit ist gezeigt, dass sich X3C auf das Problem des Findens einer eindeutigen, konsistenten GA und GTO reduzieren lässt und da X3C bekanntlich NP-vollständig ist [GJ79], muss auch unser Problem zumindest NP-schwer sein.

Da sich auch in Polynomialzeit überprüfen lässt, ob eine GA A konsistent und eindeutig ist, indem man mit topologischem Sortieren ([Las61]) eine dazugehörige GTO berechnet und die entsprechenden im Kapitel 2 angegebenen Bedingungen überprüft, liegt uSRC in NP ([Coo71]).

Damit ist gezeigt, dass uSRC NP-vollständig ist.

3.2. Keine Senderambivalenz (SRC)

Im vorherigen Abschnitt wurde bewiesen, dass das Problem, eine sowohl konsistente wie auch eindeutige GA zu finden, NP-vollständig ist. Nun stellt sich natürlich die Frage, ob das Finden einer GA, welche nur konsistent und nicht eindeutig zu sein braucht, nicht doch in Polynomialzeit möglich ist.

In diesem Abschnitt wird nun uSRC (in der Eingabe beschränkt auf die Logs, welche im obigen Beweis bei der Reduktion von X3C auf uSRC entstehen) wiederum auf SRC reduziert. Dazu werden die Logs so modifiziert, dass jede zu ihnen konsistente GA auch eine eindeutige sein muss. Dadurch gibt es für diese Eingabe keinen Unterschied zwischen einem Algorithmus, der nur eine konsistente GA sucht, und einem Algorithmus, der eine konsistente und zudem eindeutige GA sucht.

Wir betrachten dazu folgendes Konstrukt:

$$\begin{array}{cccc} A & b & \square & A \\ B & a & \boxdot & B \end{array}$$

$\square$ und $\boxdot$ stehen dabei für eine beliebige Folge von Ereignissen, welche weder A noch B enthalten

Es existieren nun genau zwei Möglichkeiten dem empfangenen a ein Sendeereignis zuzuordnen:

$$\begin{array}{ccccc|cccccc} & A & b & \square & A & A & b & \square & A & & \\ & \updownarrow & & & & & & & \updownarrow & & \\ B & a & \boxdot & B & & & & B & a & \boxdot & B \end{array}$$

Damit bleibt für das freie b jeweils nur exakt eine Möglichkeit der Zuordnung übrig:

$$\begin{array}{cccccc|cccccc} & A & & b & \square & A & A & b & \square & A & & \\ & \updownarrow & & \updownarrow & & & & \updownarrow & & \updownarrow & & \\ B & a & \boxdot & B & & & & B & & a & \boxdot & B \end{array}$$

Das bedeutet, dass $\square$ und $\boxdot$ niemals gleichzeitig geschehen können.

Sind $\square$ und $\boxdot$ einzelne Ereignisse gilt somit $A(\square) \neq A(\boxdot)$.

Um dieses Konstrukt anwenden zu können, sei noch einmal an die Definition der Logs im vorherigen Beweis erinnert:

Im vorherigen Abschnitt wurden die Logs L_i folgendermaßen definiert: Für jede Menge M_i des X3C-Problems mit $M_i = \{\alpha, \beta, \gamma\}$ und $\alpha < \beta < \gamma$ sei $L_i = \{a, b, c, x\}$ ein Log mit vier Empfangsereignissen für die gilt $P(a) = \alpha$, $P(b) = \beta$, $P(c) = \gamma$, $P(x) = X$ und $a \prec_i b \prec_i c \prec_i x$.

Damit jede konsistente GA auch eine eindeutige GA ist, umgeben wir nun jedes Paar von Empfangsereignissen desselben Typs in unterschiedlichen Logs mit dem zuvor veranschaulichten Konstrukt. Dabei müssen die Sendeereignisse jeweils am Anfang und Ende des Logs zusätzlich wiederholt werden, damit die Anwendung des Konstrukt um ein Paar von Ereignissen keine Zuordnungen zu anderen Ereignissen ausschließt. Zur besseren Lesbarkeit, führen wir diese Umformung schrittweise durch und beginnen mit Typ α:

$$^{-3}L_i' = (\mathbf{A}_{i\_}^1, \mathbf{A}_{i\_}^2, \mathbf{a}_{\_i}, a, \mathbf{A}_{i\_}^3, b, c, x, \mathbf{A}_{i\_}^4)$$

Dabei steht $\mathbf{A}_{i\_}^m$ für eine Menge von Sendeereignissen $\mathbf{A}_{i\_}^m = \{A_{ij}^m | i \neq j \wedge \exists a \in R_j : P(a) = \alpha\}$, die für jedes Log L_j, welches ein Empfangsereignis vom Typ α enthält, ein passendes Sendeereignis A_{ij}^m besitzt. Diese Ereignisse seien im Log L_i' aufsteigend sortiert gespeichert, also $A_{ij}^m \prec_i A_{ik}^m \Leftrightarrow j < k$.
Entsprechend bezeichnet $\mathbf{a}_{\_i} = \{a_{ji} | i \neq j \wedge \exists a \in R_j : P(a) = \alpha\}$ eine Menge von Empfangsereignissen. Auch hier seien diese Ereignisse in L_i' aufsteigend sortiert, also $a_{ji} \prec_i a_{ki} \Leftrightarrow j < k$.
Für alle m sei $P(A_{ij}^m) = P(a_{ij})$ und $P(A_{ij}^m) \neq P(a_{kl})$ für $i \neq k$ oder $j \neq l$ (Intuitiv bestimmt der erste Index also das sendende Log und der zweite Index das empfangende Log).
Der Index m zählt die Sendeereignisse durch und die zusätzlichen Ereignisse A_{ij}^1 und A_{ij}^4, welche im oben vorgestellten Konstrukt noch nicht enthalten waren, benötigt man wie schon erwähnt, damit sich die Konstrukte um unterschiedliche Ereignisse nicht gegenseitig blockieren.

Fügt man mit entsprechend definierten Mengen $\mathrm{B}_{i\_}^m$ und $\mathrm{b}_{\_i}$ das Konstrukt als nächstes um das Ereignis b, so ergibt sich:

$$^{-2}L_i' = (\mathbf{A}_{i\_}^1, \mathbf{B}_{i\_}^1, \mathbf{A}_{i\_}^2, \mathbf{a}_{\_i}, a, \mathbf{A}_{i\_}^3, \mathbf{B}_{i\_}^2, \mathbf{b}_{\_i}, b, \mathbf{B}_{i\_}^3, c, x, \mathbf{A}_{i\_}^4, \mathbf{B}_{i\_}^4)$$

Dies wiederholt man dann mit Ereignis c:

$$^{-1}L_i' = (\mathbf{A}_{i\_}^1, \mathbf{B}_{i\_}^1, \mathbf{C}_{i\_}^1, \mathbf{A}_{i\_}^2, \mathbf{a}_{\_i}, a, \mathbf{A}_{i\_}^3, \mathbf{B}_{i\_}^2, \mathbf{b}_{\_i}, b, \mathbf{B}_{i\_}^3, \mathbf{C}_{i\_}^2, \mathbf{c}_{\_i}, c, \mathbf{C}_{i\_}^3, x, \mathbf{A}_{i\_}^4, \mathbf{B}_{i\_}^4, \mathbf{C}_{i\_}^4)$$

Und schließlich mit Ereignis x:

$$\begin{aligned} L_i' = (&\mathbf{A}_{i\_}^1, \mathbf{B}_{i\_}^1, \mathbf{C}_{i\_}^1, \mathbf{X}_{i\_}^1, \mathbf{A}_{i\_}^2, \mathbf{a}_{\_i}, a, \mathbf{A}_{i\_}^3, \mathbf{B}_{i\_}^2, \mathbf{b}_{\_i}, b, \mathbf{B}_{i\_}^3, \\ &\mathbf{C}_{i\_}^2, \mathbf{c}_{\_i}, c, \mathbf{C}_{i\_}^3, \mathbf{X}_{i\_}^2, \mathbf{x}_{\_i}, x, \mathbf{X}_{\_i}^3, \mathbf{A}_{i\_}^4, \mathbf{B}_{i\_}^4, \mathbf{C}_{i\_}^4, \mathbf{X}_{i\_}^4) \end{aligned}$$

Für die nachfolgenden Beweise wird zudem eine Funktion B benötigt, die jedem Ereignis einen Basistyp zuweist, mit $B(A_{ij}^{m}) = B(a_{ij}) = B(a) = \alpha$ für alle i, j, m.

Das Sendelog $L'_0 = L_0$ bleibe unverändert.

Jedes Ereignis der $4n_L$ Empfangsereignisse von L wird dabei von maximal $5n_L$ Ereignismengen umgeben, die jeweils maximal n_L Ereignisse enthalten und bei denen sich für jedes potentiell enthaltene Ereignis in konstanter Zeit (durch einem Lookup in der vierelementigen Menge L_i) bestimmen lässt, ob es tatsächlich zur Menge gehört. Damit lässt sich die gesamte Konstruktion trivial in $O(n_L^3)$ erzeugen, also in Polynomialzeit.

Lemma 3.3. *Jede konsistente Lösung von L' ist eindeutig.*

Beweis. Dies folgt aus der genauen Untersuchung des oben angegebenen Konstrukts, welche sich im Anhang A.2 findet. □

Damit ist gezeigt, dass unser Ziel erreicht ist und das Finden einer eindeutigen, konsistenten und einer nur konsistenten GA für diese Logs äquivalent ist.

Allerdings ist noch nicht gesichert, dass die so modifizierten Logs weiterhin äquivalent zur gegebenen Instanz von X3C sind, was nun gezeigt wird:

Lemma 3.4. *Ist A eine zu L' konsistente GA, so ist A eingeschränkt auf L eine konsistente GA zu L.*

Beweis. Dies folgt im Grunde sofort dadurch, dass $L \subset L'$ ist.

Eine genauere Ausführung findet man im Anhang unter A.2. □

Lemma 3.5. *Existiert eine eindeutige, konsistente Lösung zu L, so gibt es eine Lösung für L'.*

Beweis. Sei $A, \prec$ eine solche Lösung zu L.

Wie im Lemma 3.2 werden die Logs wieder durch zwei Indexmenge I und J mit $I \cap J = \emptyset$ und $I \cup J = \{1, \dots, n_L\}$ aufgeteilt, so dass für alle $r \in L_i$ mit $i \in I$ gilt $A(r) \prec \blacktriangle$ und für alle $r \in L_j$ mit $j \in J$ gilt $\blacktriangle \prec A(r)$.

Aus diesen Indexmengen entstehen jeweils zwei Familien von Logs: $L_I = \{L_i | i \in I\}$ und $L_J = \{L_j | j \in J\}$.

Außerdem gilt $A(q) \prec A(r)$ für $K \in \{I, J\}$ und $k, k' \in K$ mit $k < k'$ und $q \in L_k, r \in L_{k'}$ mit $P(q) = P(r)$.

Nun wird eine neue GA A' konstruiert, indem man für jedes Empfangsereignis aus L die Zuordnung von A übernimmt und für jedes neu hinzugekommene Empfangsereignis aus $L' \setminus L$ eines der vier möglichen neuen Sendeereignisse wählt. Intuitiv betrachtet geschehen alle Empfangsereignisse der Logs L_I vor denen von L_J, also nimmt man für die Empfangsereignisse von L_I das frühestmögliche Sendeereignis in L_J und umgekehrt für die Empfangsereignisse in L_J das spätest mögliche in L_I. Innerhalb von L_I und L_J sind die Ereignisse nach ihren Typen und der Indexnummer des Logs sortiert, so dass man entsprechend dieser Sortierung jeweils das erste und letzte mögliche Sendeereignis (also das zweite und vorletzte im Sendelog, da das erste und letzte ja bereits zugeordnet sind) wählt.
Konkret bedeutet das:

Für alle $s \in S'$ sei

$$A'(s) = s$$

Für alle $e \in L$ sei

$$A'(e) = A(e)$$

Für $i \in I$ und $e_{ki} \in L' \setminus L$ sei

$$A'(e_{ki}) = \begin{cases} E^3_{ki} & \text{falls } k < i \land k \in I \\ E^1_{ki} & \text{falls } k \in J \lor (k > i \land k \in I) \end{cases}$$

Dabei bezeichnet $E^m_{ki} \in S_k$ ein Sendeereignis mit $P(E^m_{ki}) = P(e_{ki})$ an der entsprechenden, durch m bestimmten Position im Log.

Für $j \in J$ sei

$$A'(e_{kj}) = \begin{cases} E^4_{kj} & \text{falls } k \in I \\ E^3_{kj} & \text{falls } k < j \wedge k \in J \\ E^2_{kj} & \text{falls } k > j \wedge k \in J \end{cases}$$

Es ist nun leicht zu sehen, dass A' eine GA ist. Zu beweisen, dass A' auch konsistent ist, ist etwas komplizierter und lässt sich zeigen, indem man eine Reihenfolge der Empfangsereignisse festlegt, bei denen die Ereignisse aus L_I vor denen von L_J geschehen und die restlichen Ereignisse nach ihrem Typ und der Nummer des sie enthaltenden Logs sortiert sind.

Die Details hierzu findet man im Anhang A.2.

Da die oben angegeben GA A' also konsistent ist, stellt sie die gewünschte Lösung für die Logs L' dar. □

Damit haben wir nun gesehen, dass sich aus den Logs L des Beweis des vorherigen Abschnitts in Polynomialzeit eine Menge von neuen Logs L' konstruieren lässt, so dass das Finden einer beliebigen konsistenten Lösung für diese L' äquivalent zum Finden einer eindeutigen Lösung für die Logs L und damit zum Finden einer Lösung des X3C-Problems ist. Damit wurde letzeres, NP-vollständige Problem auf das Finden einer konsistenten Lösung reduziert.

Da sich jede konsistente Lösung auch offenkundig in Polynomialzeit als konsistent verifizieren lässt, ist das Finden einer konsistenten Lösung in NP, und nach dem obigen Beweis sogar NP-vollständig.

3.3. Möglicher Sender im Fall ohne Senderambivalenz (pSRC)

In den obigen Abschnitten wurden nur die neuen Probleme SRC und uSRC betrachtet, so dass die Untersuchung von pSRC noch aussteht. Es liegt jedoch nahe, dass es ebenfalls NP-vollständig ist.

Um dies zu zeigen, wird nun ein Verfahren beschrieben, mit dem sich in Polynomialzeit eine konsistente GA finden lässt, sofern man über einen schnellen Algorithmus für pSRC verfügt. Dabei nimmt der pSRC-Algorithmus die Rolle eines Orakels ein, welches zu jedem Empfangsereignis eine Menge der möglichen Sendeereignisse berechnet.

Es lassen sich mit diesem Orakel drei Fälle unterscheiden:

Fall 1: Es gibt ein Empfangsereignis, für das kein mögliches Sendeereignis existiert.

Dann existiert offensichtlich keine konsistente GA.

Fall 2: Für jedes Empfangsereignis gibt es nur genau ein mögliches Sendeereignis.

Das heißt, für jedes Empfangsereignis r existiert genau ein s, so dass es eine konsistente GA A mit $A(r) = s$ gibt und dass es für $s' \neq s$ keine konsistente GA mit $A(r) = s'$ geben kann. Ist A nun eine konsistente GA, so folgt, dass $A(r) = s$ für alle $r \in R$ und dem einzigen nach dem Orakel für r möglichen Sendeereignis s. Da nun für jedes Empfangsereignis ein Sendeereignis gegeben ist, ist A durch diese Zuordnung eindeutig bestimmt und man kann in Linearzeit (die für das Erkennen des Falles nötig ist) eine konsistente GA finden.

Fall 3: Es gibt ein Empfangsereignis r, dem sich mehrere Sendeereignisse zuordnen lassen.

Sei s eines dieses Sendeereignisse. Dann gibt es eine konsistente GA A mit $A(r) = s$. Man kann nun in den Logs das Ereignis s durch drei Sendeereignisse XsY ersetzten (bzw. es mit zwei umgeben) und zugleich r durch drei Empfangsereignisse xry ersetzten. Dabei sei $P(X) = P(x)$, $P(Y) = P(y)$ und $P(X) \neq P(e) \neq P(Y)$ für alle anderen Ereignisse e. Damit gibt es jeweils nur genau ein mögliches Sendeereignis und es gilt $A(x) = X$ und $A(y) = Y$ für jede GA. Aus den Konsistenzbedingungen $A(x) \prec A(r) \prec A(y)$ und $X \prec s \prec Y$ mit $s' \prec X$ oder $Y \prec s'$ für alle $s' \neq s$ im selben Log wie s folgt, dass es in den modifizierten Logs keine GA A mit $A(r) \neq s$ geben kann. Damit hat man offensichtlich in Polynomialzeit die Zahl der Empfangsereignisse ohne eindeutige Zuordnung um mindestens eins reduziert.

Wiederholt man dies, bis für die modifizierten Logs Fall 1 oder Fall 2 eintritt (was maximal eine Wiederholung für jedes Empfangsereignis erfordert), so erhält man schließlich eine konsistente GA.

Somit ist es möglich SRC in Polynomialzeit auf pSRC zu reduzieren.

Umgekehrt kann man überprüfen, ob eine Zuordnung zwischen einem Empfangs- r und Sendeereignis S möglich ist, indem man eine konsistente GA sucht, die diese Zuordnung enthält. Dazu ersetzt man einfach wie oben S durch XSY und r durch xry, so dass es keine konsistente GA ohne diese Zuordnung gibt, und ruft dann einen SRC-lösenden Algorithmus als Orakel auf. Da es nur n^2 mögliche Zuordnungen gibt, kann man also auch pSRC in Polynomialzeit auf SRC reduzieren.

Da SRC wie im vorherigen Abschnitt gezeigt NP-vollständig ist, muss also auch pSRC NP-vollständig sein.

Kapitel 4.

Intervallbasierte Zuordnung und Heuristik

In [SK09] wurde als Polynomialzeit-Algorithmus für das optimale Lösen von pSRC eine Heuristik vorgestellt, die für jedes Empfangsereignis ein Intervall von möglichen Sendeereignissen ermittelt, welche dem Ereignis zugeordnet werden können.

Nach den vorherigen Überlegungen ist aber klar, dass ein solches Verfahren diese Erwartungen nicht erfüllen kann (es sei denn, es gälte $P = NP$).

In diesem Kapitel wird nun zuerst – unabhängig vom eigentlichen Algorithmus – bewiesen, dass schon die Annahme, dass jedem Empfangsereignis ein geschlossenes Intervall von Sendeereignissen zugewiesen werden kann, nicht erfüllt ist; dann wird gezeigt, dass die von der Heuristik gefundenen Intervalle im Allgemeinen nicht einmal minimal sind, und schließlich einige Spezialfälle vorgestellt, in denen sie es doch sind.

4.1. Mangelhaftigkeit einer Intervallzuordnung

Lemma 4.1. *Die möglichen Sendeereignisse bilden nicht immer ein geschlossenes Intervall.*

Beweis. Dies wird mit einem Widerspruchsbeweis gezeigt, bei dem wir folgende Eingabe betrachten:

$$\begin{array}{cccccc}
\mathrm{B} & \mathrm{c} & \mathrm{a} & \mathrm{B} & & \\
\mathrm{C} & \mathrm{A} & \mathrm{C} & \mathrm{b} & \mathrm{C} & \mathrm{A}
\end{array}$$

Damit sind diese beiden Zuordnungen möglich:

$$\begin{array}{ccccccc|ccccccc}
\mathrm{B} & \mathrm{c} & \mathrm{a} & & \mathrm{B} & & & & & & \mathrm{B} & \mathrm{c} & \mathrm{a} & \mathrm{B} \\
 & \uparrow & \uparrow & & \downarrow & & & & & & \downarrow & \uparrow & \uparrow & \\
 & \mathrm{C} & \mathrm{A} & \mathrm{C} & \mathrm{b} & \mathrm{C} & \mathrm{A} & \mathrm{C} & \mathrm{A} & \mathrm{C} & \mathrm{b} & \mathrm{C} & \mathrm{A} &
\end{array}$$

Das empfangene *c* kann also sowohl dem ersten wie auch dem letzten gesendeten *C* zugeordnet werden. Also muss das Intervall der für *c* möglichen Sendeereignisse sämtliche *C*s enthalten. Würden die möglichen Sendeereignisse nun ein geschlossenes Intervall bilden, so gäbe es auch mit dem mittleren *c* eine gültige Zuordnung. Aber es ist trivial zu zeigen, dass es eben keine solche Zuordnung gibt, indem man versucht ein Sendeereignis für das übrig gebliebene *a* oder *b* zu finden:

$$\begin{array}{ccccccc|ccccccc}
 & \mathrm{B} & \mathrm{c} & \mathrm{a} & \mathrm{B} & & & & \mathrm{B} & \mathrm{c} & & & \mathrm{a} & \mathrm{B} \\
 & & \uparrow & ???? & \downarrow & & & & & \uparrow & ???? & & \uparrow & \\
\mathrm{C} & \mathrm{A} & \mathrm{C} & & \mathrm{b} & \mathrm{C} & \mathrm{A} & \mathrm{C} & \mathrm{A} & \mathrm{C} & \mathrm{b} & \mathrm{C} & \mathrm{A} &
\end{array}$$

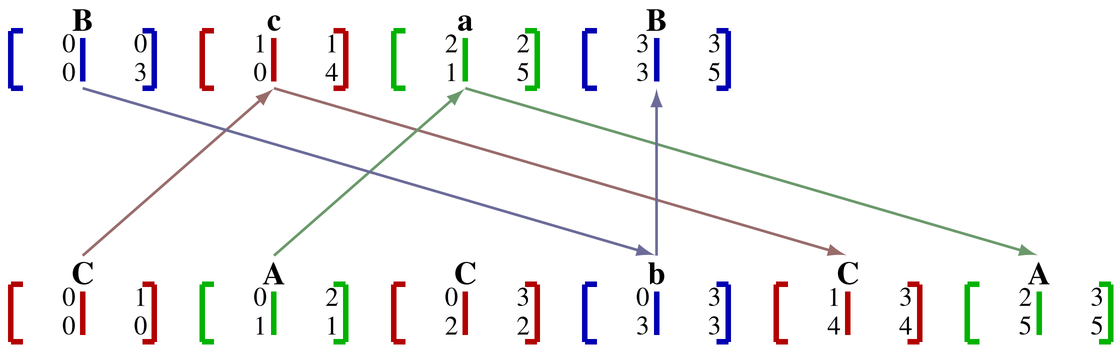
Offensichtlich ist es nicht möglich, diesen beiden Empfangsereignissen gleichzeitig ein Sendeereignis zuzuordnen. Also ist das mittlere *C* kein gültiges Sendeereignis für das empfangene *c*. Jeder auf Intervallen basierende Algorithmus liefert also immer einige falsche Zuordnungen

□

4.2. Probleme der Heuristik

Nach dem vorherigen Abschnitt wissen wir, dass der in [SK09] vorgestellte und dann in [Mar08] realisierte Algorithmus nicht garantieren kann, dass die von ihm gefundenen Sendeereignisse tatsächlich eine mögliche Zuordnung darstellen. Trotzdem könnte es sein, dass das Verfahren in dem Sinne optimal wäre, dass der Algorithmus die bestmöglichen (also kleinsten) Intervalle findet, die noch alle möglichen Sendeereignisse umfassen.

Leider ist dies nicht der Fall, was durch ein einfaches Gegenbeispiel gezeigt werden kann. Aber um dieses Gegenbeispiel zu verstehen, muss man die Funktionsweise der Heuristik kennen, weswegen sie zuvor beschrieben wird. Dabei beginnen wir mit der verwendeten Datenstruktur, die exemplarisch an Hand der Ausgabe erklärt wird, welche entsteht, wenn die Heuristik auf die Logs des vorherigen Abschnitts angewandt wird. [1]:



An diesem Bild kann man auf zwei Weisen die von der Heuristik gefundenen Intervalle ablesen: Die eine, visuelle Darstellung erfolgt in Form von Pfeilen, von denen sich bei jedem Empfangsereignis genau zwei treffen. Der erste Pfeil führt vom frühestmöglichen Sendeereignis zum Empfangsereignis und der zweite Pfeil führt vom Empfangsereignis zum spätestmöglichen Sendeereignis.

Diese intuitive Pfeildarstellung ist leicht verständlich, kann aber nur schlecht als interne Darstellung im Rechner verwendet werden, weshalb [Mar08] dazu Intervallvektoruhren einführt (siehe die formale Definition des Algorithmus im Anhang B.4). Diese Intervallvektoruhren basieren auf Matterns Vektoruhren [Mat89], die angeben, welche Ereignisse gleichzeitig geschehen können. Dies kann man leicht an Hand des obigen Bildes verstehen, wo die Intervallvektoruhren als Matrix unter den Ereignistypen dargestellt sind. Diese Matrix enthält zwei Zeilen, eine Zeile für jedes Log. Bei jedem Ereignis enthält die

[1] Diese sind nicht geeignet, um den Ablauf des Algorithmus zu beschreiben, da er bei dieser Eingabe sofort terminiert.

linke Spalte für jedes Log die Nummer[2] des frühesten Ereignisses im Log dieser Zeile, welches gleichzeitig mit dem Ereignis, dem die Matrix zugeordnet ist, aufgetreten sein kann. Die rechte Spalte gibt analog die Nummer des spätestmöglichen Ereignisses an.

Betrachtet man die Matrizen im Bild genauer, erkennt man, dass in der ersten Reihe von Ereignissen, der Matrixeintrag in der linken Spalte und ersten Zeile identisch zum Eintrag in der rechten Spalte ist. Dasselbe gilt für die zweite Zeile der Matrizen der zweiten Reihe. Die Erklärung hierfür ist simpel: Diese Matrixzeile gibt jeweils an, welche Ereignisse in diesem Log gleichzeitig zu diesem Ereignis passiert sein können, und das kann, da die Ereignisse in einem Log eine feste Reihenfolge besitzen, offensichtlich nur das Ereignis selbst sein. In dieser Zeile steht somit die eindeutige ID des Ereignisses bezüglich des eigenen Logs.

Die Pfeildarstellung lässt sich dabei trivial aus der Matrix auslesen: Bei jedem Empfangsereignis steht in der Zeile, welche nicht die eindeutige ID enthält, in der linken Spalte die ID des frühestmöglichen Sendeereignisses und in der rechten Spalte die ID des spätestmöglichen Sendeereignisses. Damit kann man im Sendelog (welches in dem von der Heuristik behandelten Spezialfall pSRC durch den Typ des Ereignisses eindeutig festgelegt ist) die entsprechenden Sendeereignisse finden und mit dem Empfangsereignis über die beiden Pfeile verbinden.

Die Matrizen der Sendeereignisse geben entsprechend ein Intervall der möglichen Empfangsereignisse an. Im Bild sieht man viele Sendeereignisse, bei denen diese Intervalle sich genau bis zum ersten und letzten Ereignis erstrecken. Dies liegt einfach daran, dass diese Sendeereignisse möglicherweise nicht empfangen werden und daher vor oder nach allen Ereignissen im Empfangslog aufgetreten sein könnten (man könnte hier für die Matrixeinträge ebenso gut $-\infty$ und $+\infty$ nehmen).

Sowohl in der Pfeil- wie auch in der Vektoruhrdarstellung kann man nun ein Kriterium dafür definieren, ob eine solche Zuordnung des frühesten oder spätesten Sendeereignisses möglich ist. Bei der von [SK09] verwendeten Pfeildarstellung basiert es auf der Beobachtung, dass die Ereignisse innerhalb eines Logs in einer strikten Reihenfolge angeordnet sind und es damit innerhalb des Logs keine Ereignisse geben kann, die gleich-

[2] Die Ereignisse seien in jedem Log mit Index 0 beginnend nummeriert.

zeitig geschehen können. Damit stellt die Reihenfolge im Log offenbar eine »kleiner«-Relation dar, während die eingezeichneten Pfeile eine »kleiner-gleich«-Relation darstellen. Es darf also keine Pfeile geben, deren Ende sich ausgehend von ihrem Anfang durch einen anderen Pfeil erreichen lässt, der später in ihrem Log beginnt. Außerdem darf es keinen Pfad von einem Pfeilanfang zu einem Pfeilende (oder sogar zu einem Ereignis vor dem Pfeilende) geben, der sich entlang der Pfeile und der Ereignisse in einem Log erstreckt.

In der inhaltlich äquivalenten Intervalldarstellung von [Mar08] lautet die Bedingung, dass in jeder Komponente das Minimum eines Empfangsereignisses größer-gleich dem Minimum des ersten Sendeereignisses sein muss und das Maximum kleiner-gleich dem des letztmöglichen Sendeereignisses. Außerdem muss das Maximum des frühestmöglich Sendeereignisses in jeder Komponente kleiner-gleich sein, als das des Empfangsereignisses, während das Minimum des spätestmöglichen Sendeereignisses größer-gleich dem des Empfangsereignisses sein muss. Außerdem muss in dieser Darstellung in einer zweiten Bedingung verlangt werden, dass die Werte einer Matrix immer kleiner-gleich den Werten der Matrix des nächsten Ereignisses dieses Logs sind.

Die Heuristik entfernt nun einfach sämtliche solchermaßen unmöglichen Zuordnungen, bis keine unmöglichen Zuordnungen mehr existieren. In der Pfeildarstellung entspricht dieses Entfernen dem Verschieben der Pfeilenden zum nächsten Ereignis vom selben Typ. In der Intervallvektoruhrdarstellung setzt man einfach die Matrixwerte auf das gewünschte Maximum/Minimum (so dass die Intervalle jeweils kleiner werden). Das Wiederherstellen der ersten Bedingung heißt »Kreuztausch«, dass der zweiten »Wischen«.

Leider funktioniert dieses Verfahren nicht, wie nun hier gezeigt wird:

Lemma 4.2. *Die von der Heuristik gefundenen Intervalle sind nicht minimal.*

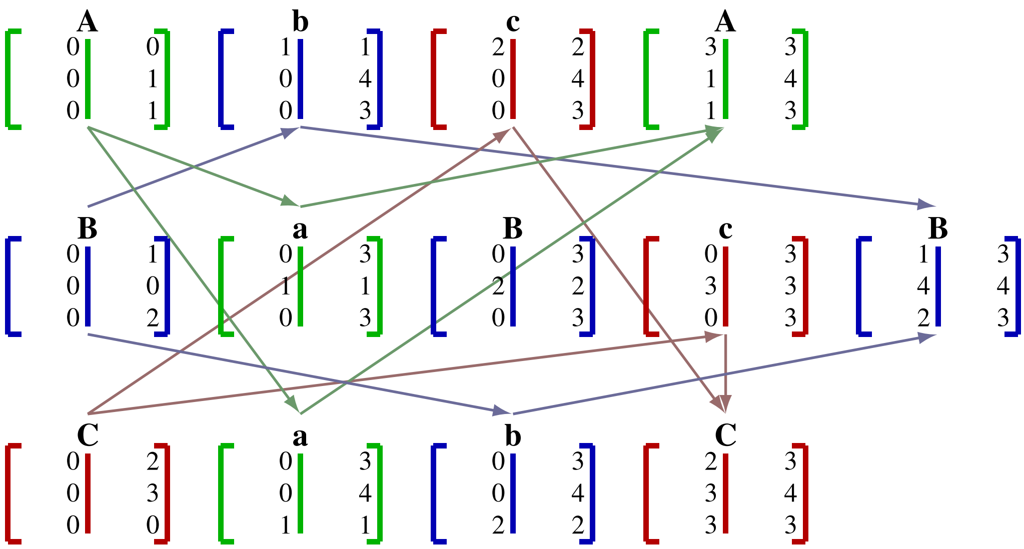
Beweis. Dazu sehen wir uns einfach folgende Eingabe an:

A	b	c	A	
B	a	B	c	B
C	a	b	C	

Dazu gibt es mindestens eine gültige Lösung, also handelt es sich tatsächlich um eine mögliche Eingabe:

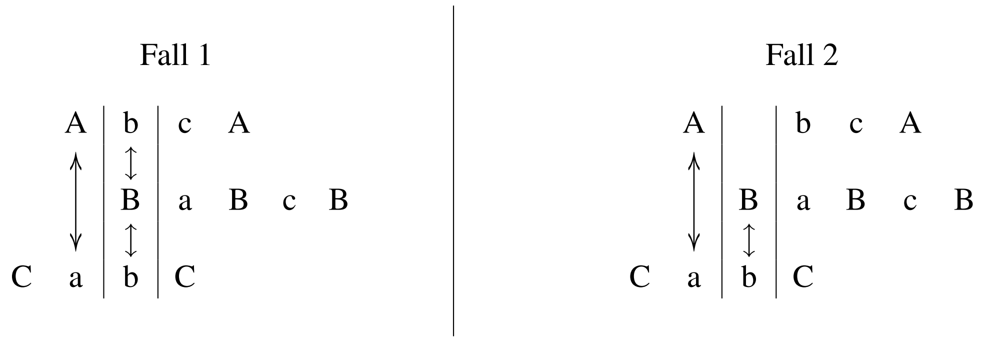
	A	b	c	A
	↕	↕	↕	
B	a	B	c	B
	↕	↕	↕	
C	a	b	C	

Dann liefert die Heuristik folgende Ausgabe: (man kann sich davon überzeugen, dass dies in der Tat die Ausgabe des Algorithmus ist, indem man feststellt, dass die angegebenen Intervallvektoruhren weder durch Kreuztausche noch durch Wischen verändert werden. Siehe auch die formale Definition des Algorithmus im Anhang B.4)

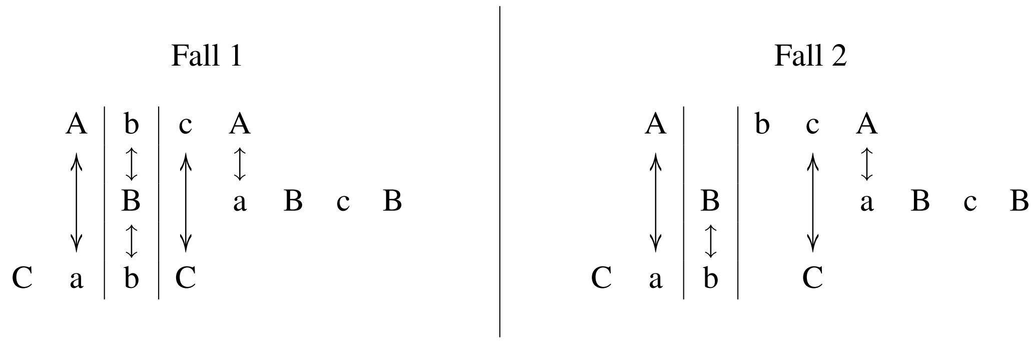


Wie man sieht wird das erste B als mögliches Sendeereignis für das b im dritten Log gefunden.

Um zu überprüfen, ob dies korrekt ist, nehmen wir nun diese Zuordnung als gegeben an. Dann gibt es zwei Möglichkeiten, entweder (Fall 1:) ist das empfangene b im ersten Log ebenfalls diesem Sendeereignis zugeordnet, oder (Fall 2:) einem der anderen gesendeten Bs: (Das erste A muss in diesem Fall offensichtlich vor dem ersten B gesendet werden, sonst gibt es keine gültige Zuordnung für das empfangene a im dritten Log)



In der rechten Hälfte gibt es genau ein gesendetes A und ein gesendetes C, also gibt es nur genau eine mögliche Zuordnung für das im ersten Log empfangene c und das im zweiten Log empfangene a:



Offensichtlich ist es nun nicht mehr möglich, dem in Log 2 empfangenen c ein Sendeereignis zuzuordnen.

Also liefert der Algorithmus für das b ein frühestmögliche Sendeereignis, welches nicht zu einer gültigen Lösung führt. Somit ist das gefundene Intervall nicht minimal.[3]

□

[3] Allerdings basiert dieses Gegenbeispiel auf der Annahme, dass Sende- und Empfangsereignis exakt zum gleichen Zeitpunkt aufgetreten sind. Gäbe es eine – von [SK09] jedoch explizit ausgeschlossene – Empfangsverzögerung wäre die obige Zuordnung gültig.

4.3. Korrekt gelöste Spezialfälle

In diesem Abschnitt werden drei Spezialfälle genannt, in denen das Ergebnis der Heuristik trotz der in den vorherigen Abschnitten aufgezeigten Probleme korrekt ist.

Prinzipiell muss dafür gezeigt werden, dass es für jede (oder einige) der durch die Intervalle bestimmten möglichen Zuordnungen eine gültige Lösung (in etwa eine konsistente GA und GTO) gibt, die diese Zuordnung enthält. Aus Platzgründen ist dieser Abschnitt aber relativ kurz gehalten und man findet die weiteren Details im Anhang.

Lemma 4.3. *Die Heuristik funktioniert fehlerfrei, wenn es nur ein einziges Log gibt, welches Empfangsereignisse enthält.*

Beweis. Es ist klar, dass keinen zwei Empfangsereignissen innerhalb eines Logs dieselben Sendeereignisse zugewiesen werden (sonst gäbe es zwei Pfeile zu bzw. von diesem Sendeereignis und somit zwei Pfade).

Damit lässt sich nun zeigen, dass eine gültige Lösung entsteht, wenn man jedem Empfangsereignis das nach der Heuristik früheste Sendeereignis zuweist. Dasselbe gilt äquivalent für das spätestmögliche Sendeereignis.

Daraus folgt wiederum, dass jede Zuordnung innerhalb des gefundenen Intervalls tatsächlich möglich ist. Dazu nimmt man einfach eine Zuordnung als gegeben an und weist jedem früheren Empfangsereignis (innerhalb des einzigen Empfangslogs) das frühestmögliche und jedem späteren Empfangsereignis das spätestmögliche Sendeereignis zu, wodurch man eine gültige Lösung enthält.

Siehe Anhang B.6 für Details. □

Lemma 4.4. *Die Heuristik funktioniert fehlerfrei, wenn es nur ein einziges Log gibt, welches Sendeereignisse enthält.*

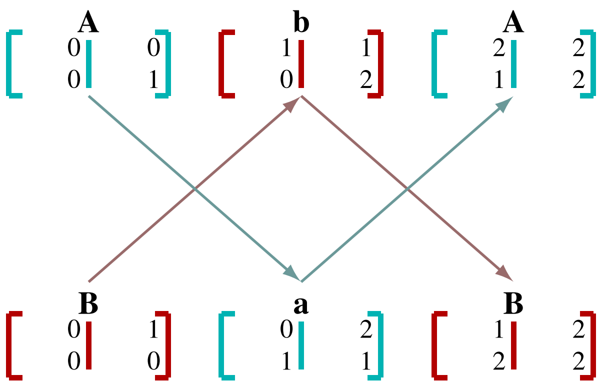
Beweis. Dies kann man relativ direkt aus dem vorherigen Lemma 4.3 folgern, indem man zeigt, dass sich die Ereignisse in den unterschiedlichen Logs nicht gegenseitig beeinflussen. Das heißt, dass der Algorithmus genau dieselben Zuordnungen findet, unabhängig davon, ob man ihn auf alle Logs gemeinsam anwendet, oder einzeln für jedes Empfangslog zusammen mit dem Sendelog.

Gibt es aber nur ein Empfangslog und ein Sendelog, so wurde im vorherigen Lemma gezeigt, dass jedes Sendeereignis innerhalb des Intervalls zu einer gültigen Zuordnung führt, und infolgedessen gilt dies auch für die gesamten Logs.

Siehe Anhang B.7 für Details. □

Lemma 4.5. *Die von der Heuristik gefundenen Intervalle sind minimal, wenn es insgesamt nur zwei Logs gibt.*

Beweis. In diesem Fall muss man eine andere Beweisstrategie wählen, als in den vorherigen Lemmas, da sich leider keine gültige Lösung ergibt, wenn man für alle Ereignisse das frühestmögliche Sendeereignis wählt. Dies zeigt folgendes Trivialbeispiel:



Stattdessen benutzt man die Tatsache, dass jede Zuordnung zweier Ereignisse in diesem Spezialfall die Logs in genau zwei Hälften teilt. Eine linke Hälfte, in der sich alle früheren Ereignisse befinden, und in eine rechte Hälfte in der sich alle späteren befinden (früher und später bezieht sich dabei auf die lokale Reihenfolge in jedem Log).

Dann lässt sich zeigen, dass, wenn man einem Empfangsereignis entweder das im gefundenen Intervall früheste oder späteste Sendeereignis zuweist, in jeder Hälfte einem Ereignis wiederum entweder das früheste oder späteste Sendeereignis zugewiesen werden kann, und zwar so, dass eine der dadurch entstehenden Hälften (geschnitten mit der Hälfte in der man die Zuordnung durchgeführt hat) kein Empfangsereignis enthält. Veranschaulicht heißt dies: (B sei ein nach der Heuristik frühest- oder spätestmögliche Sendeereignis, $\square$ bezeichnet irgendein Ereignis)

$$\begin{array}{ccccccccc} \ldots & \square & \square & \square & \mathrm{B} & \square & \square & \square & \ldots \\ & & & & & \updownarrow & & & \\ \ldots & \square & \square & \square & \mathrm{b} & \square & \square & \square & \ldots \end{array}$$

Die obrige, als Text schwer verständliche Aussage bedeutet nun im Prinzip, dass es Ereignisse a, A, c, C gibt, so dass die Sendeereignisse A und C frühest- oder spätestmögliche Sendeereignisse sind und kein $\boxdot$ ein Empfangsereignis ist: (das Bild ist dabei vereinfacht, es könnte auch sein, dass die Sende- und Empfangsereignisse im jeweils anderen Log liegen, so dass sich *nicht* alle Sendeereignisse im selben Log befinden)

$$\begin{array}{ccccccccccccc} \ldots & \square & \mathrm{A} & \boxdot & \ldots & \boxdot & \mathrm{B} & \boxdot & \ldots & \boxdot & \mathrm{C} & \square & \ldots \\ & & \updownarrow & & & & \updownarrow & & & & \updownarrow & & \\ \ldots & \square & \mathrm{a} & \boxdot & \ldots & \boxdot & \mathrm{b} & \boxdot & \ldots & \boxdot & \mathrm{c} & \square & \ldots \end{array}$$

Damit kann man nun in der rechten und linken Hälfte rekursiv neue Ereignisse einander zuordnen, wodurch eine gültige Lösung entsteht. (Man könnte auf die Idee kommen, dieses Beweisverfahren vereinfachen zu wollen, indem man links mit einem ersten Ereignis beginnt und dann induktiv neue Zuordnungen hinzufügt. Dies funktioniert jedoch nicht, da man dann nicht für jedes Empfangsereignis beide Sendeereignisse der Intervallgrenzen für möglich erklärt.)

Siehe Anhang B.8 für Details. $\square$

Ein weiterer denkbarer Spezialfall ist es, dass es eine beliebige Anzahl von reinen Senderlogs und Empfängerlogs gibt; oder dass es mehrere Sender und Empfängerlogs, sowie maximal ein oder zwei gemischte Logs gibt. Diese Fälle werden aber in dieser Arbeit nicht weiter untersucht.

Kapitel 5.

Alternativer Algorithmus

5.1. Spezialfall: $n_L = 2$

In diesem Kapitel wird ein Algorithmus entworfen, der uSRC* exakt löst, ohne dafür sämtliche Kombinationen von möglichen Zuordnungen auszuprobieren. Auf Grund der NP-Vollständigkeit hat er aber weiterhin exponentielle Laufzeit, allerdings nicht bezüglich der Zahl der Ereignisse, sondern bezüglich der Zahl der Logs. Zur besseren Verständlichkeit beginnen wir dabei mit dem Spezialfall, dass nur genau 2 Logs existieren[1] (wir gehen in diesem Kapitel davon aus, dass $L_0 = \emptyset$ ist, da $n_L = 2$ als eine intuitivere Darstellung für zwei Logs erscheint, als etwa $n_L = 1$).

Als nächstes wird ein neues Kriterium für die Existenz einer Lösung für uSRC* formuliert (welches in ähnlicher Form in [Mat89] als sogenannter *Schnitt* benutzt wird):

Da keinen zwei Empfangsereignissen dieselben Sendeereignisse zugewiesen werden dürfen, kann man, wenn es eine eindeutige und konsistente GA gibt, alle Empfangsereignisse als $r_1, r_2, \ldots, r_{n_R}$ durchnummerieren, so dass für $k \in \{1,2\}$ und alle $r_i, r_j \in R_k$ mit $i < j$ gilt $r_i \prec_k r_j$ und $A(r_i) \prec_{3-k} A(r_j)$

Umgekehrt folgt aus der Existenz einer solchen Nummerierung, dass eine eindeutige und konsistente GA existiert (vergleiche das ähnliches Lemma A.2 im Anhang).

[1] In diesem Fall gilt zudem uSRC* = SRC* , da kein Sender seine eigenen Nachrichten empfangen darf.

Das bedeutet, dass eine Instanz von uSRC* genau dann lösbar ist, wenn es ein erstes Empfangsereignis $r_1 \in R_l$ mit einer Zuordnung $A(r_1)$ gibt, so dass die Logs eingeschränkt auf $L' = \{e \in L | r_1 \prec_l e \vee A(r_1) \prec_{3-l} e\}$ lösbar sind (da es genau dann dort weitere $r_2, \ldots, r_{n_R}$ gibt)[2].

Damit kann man zwei beliebige Ereignisse $e_1 \in L_1$ und $e_2 \in L_2$ herausgreifen und untersuchen unter welchen Bedingungen es eine Lösung für ihre rechte Hälfte gibt: Der erste, triviale Fall ist, dass es in der rechten Hälfte keine Empfangsereignisse gibt und sie damit offensichtlich lösbar ist. Der zweite Fall ist, dass man ein frühestes Empfangsereignis[3] finden kann, dem ein Sendeereignis so zugeordnet werden kann, dass keine Empfangsereignisse zwischen diesen beiden Ereignissen und e_1, e_2 liegen. Bezeichnet man diese beiden neuen, einander zugeordneten Ereignisse als e'_1 und e'_2, so lässt sich die rechte Hälfte von e_1 und e_2 offenbar genau dann lösen, wenn sich die rechte Hälfte von e'_1 und e'_2 lösen lässt, da man deren (Zwischen-)Lösung um e'_1 und e'_2 erweitern kann, um eine Lösung für die rechte Hälfte von e_1 und e_2 zu erhalten. Formal ergibt sich dafür die folgende rekursive Definition einer Funktion, die angibt, ob die rechte Hälfte lösbar ist:

$$\begin{aligned}
\mathrm{Solvable}_R\ & (e_1, e_2) \\
= \ & (\nexists r \in R_1 \text{ mit } e_1 \prec_1 r \wedge \nexists r \in R_2 \text{ mit } e_2 \prec_2 r) \\
& \vee(\exists e'_1 \in L_1, e'_2 \in L_2, \text{ so dass gilt: } \quad e_1 \prec_1 e'_1 \text{ und } e_2 \prec_2 e'_2 \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge P(e'_1) = P(e'_2) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge D(e'_1) \neq D(e'_2) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \nexists r \in R_1 \text{ mit } e_1 \prec_1 r \prec_1 e'_1 \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \nexists r \in R_2 \text{ mit } e_2 \prec_2 r \prec_2 e'_2 \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \mathrm{Solvable}_R\ (e'_1, e'_2))
\end{aligned}$$

Entsprechend kann man auch eine analoge Bedingung für die Existenz einer Lösung in der linken Hälfte angeben:

[2] Dies ähnelt dem Vorgehen im Beweis B.17 des Spezialfalls $n_L = 2$ bei der Heuristik.

[3] Also ein r_2, wenn man eines der Ereignisse e_1, e_2 als r_1 bezeichnet.

$$
\begin{array}{lll}
\text{Solvable}_L & (e_1, e_2) & \\
& = \quad (\nexists r \in R_1 \text{ mit } r \prec_1 e_1 \wedge \nexists r \in R_2 \text{ mit } r \prec_2 e_2) & \\
& \quad\ \vee(\exists e_1' \in L_1, e_2' \in L_2, \text{ so dass gilt:} & e_1' \prec_1 e_1 \text{ und } e_2' \prec_2 e_2 \\
& & \wedge P(e_1') = P(e_2') \\
& & \wedge D(e_1') \neq D(e_2') \\
& & \wedge \nexists r \in R_1 \text{ mit } e_1' \prec_1 r \prec_1 e_1 \\
& & \wedge \nexists r \in R_2 \text{ mit } e_2' \prec_2 r \prec_2 e_2 \\
& & \wedge \text{Solvable}_L\ (e_1', e_2'))
\end{array}
$$

Dann ist $s_2 \in S_2$ offensichtlich genau dann ein gültiges Sendeereignis für $r_1 \in R_1$, wenn gilt $P(r_1) = P(s_2)$, $\text{Solvable}_L(r_1, s_2)$ und $\text{Solvable}_R(r_1, s_2)$.

Der Trick ist nun, dass sich $\text{Solvable}_{L/R}$ in Polynomialzeit berechnen lässt, da e_1 eines der Ereignisse aus L_1 und e_2 eines der Ereignisse aus L_2 sein muss. Damit gibt es offenbar nur maximal $O(n^2)$ mögliche Werte für die Parameter von $\text{Solvable}_{L/R}$, also braucht man maximal $O(n^2)$ Aufrufe, da man das Ergebnis entsprechend cachen kann ($n = |L|$ sei die Zahl der Ereignisse).

Die neuen Ereignisse e_1' und e_2' lassen sich mittels Hashing in konstanter Zeit finden, da man vor Berechnung von $\text{Solvable}_{L/R}$ für jedes Log in linearer Zeit für jede Position das jeweils nächste Empfangsereignis und das nächste Sendeereignisse von jedem Typ speichern kann. Damit hat dieser Algorithmus maximal eine Laufzeit von $O(n^2)$ zusammen mit einem Speicherverbrauch von $O(n^2)$.

Statt durch Caching kann man die Mehrfachberechnung von Werten auch mittels dynamischer Programmierung – also durch ein iteratives statt rekursives Vorgehen – verhindern. Zur Demonstrierung dieses Vorgehens wenden wir den alternativen Algorithmus auf die Logs des Gegenbeispiel aus Abschnitt 4.1 an, welches von keinem auf Intervallen basierenden Algorithmus gelöst werden könnte und folgendermaßen lautete:

B	c	a	B		
C	A	C	b	C	A

Für ein iteratives Vorgehen muss man für alle möglichen Eingabewerten von Solvable_R (also für alle Paare von Ereignissen) das resultierende Ergebnis speichern können. Dazu wird eine Matrix benötigt, die in folgender Grafik dargestellt wird (dabei sind alle Trennlinien nach Sendeereignissen ausgelassen):

	C	A	C	b	C	A
B				?		
c	?		?		?	
a		?				?
B				?		

Dabei sind nur die ein ? enthaltenen Felder relevant, da es offensichtlich eine gültige Lösung nur dann geben kann, wenn einem Empfangsereignis ein Sendeereignis von demselben Typ zugeordnet wird.

In der Definition von Solvable$_R$ werden zwei Fälle unterschieden, der erste, triviale Fall ist, dass es rechts von den als Parameter übergebenen Ereignissen keine Empfangsereignisse gibt. Alle Einträge, bei denen dieser Fall auftritt, lassen sich in der Matrix leicht mit * als möglich markieren:

	C	A	C	b	C	A
B				?		
c	?		?		?	
a		?				*
B				*		

Im zweiten Fall wird in jedem Log das nächste Empfangsereignis gesucht und überprüft, ob es dafür ein mögliches Sendeereignis innerhalb der betrachteten Hälfte gibt. Umgekehrt formuliert heißt dies, dass die Information innerhalb jeder Zeile und Spalte nach links bzw. oben bis zum nächsten, durch eine Trennlinie dargestellten Empfangsereignis propagiert wird (weil dazwischen liegende Sendeereignisse keine Zuordnung zerstören können). In jeder mit ? markierten Zelle, die bekanntlich für eine potentielle Zuordnung zwischen zwei Ereignissen steht, wird zudem der Wert aus der nach rechts unten diagonal benachbarten Zelle übernommen, da diese Zelle für die rechte Hälfte – ohne diese beiden Ereignisse – die Lösbarkeit angibt. Dieser Informationsfluss lässt sich folgendermaßen mittels Pfeile darstellen:

	C	A	C	b	C	A
B				?		
c	?		?		?	
a		?				*
B				*		

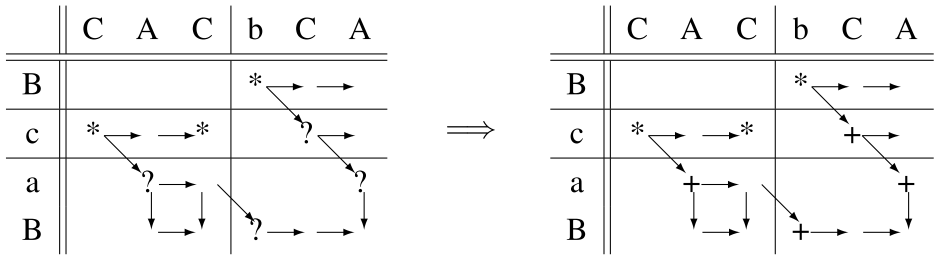
Jede Zuordnung ist genau dann bezüglich ihrer rechten Hälfte möglich, wenn es eine beliebige Zuordnung rechts (bzw. im Bild auch unterhalb) von ihr gibt, die zu lösbaren Logs führt. Das heißt, jede Zuordnung ist genau dann möglich, wenn (mindestens) ein Pfad von einer anderen möglichen Zuordnung zu ihr existiert[4]. Markiert man die dadurch möglichen Zuordnungen mit + und die unmöglichen mit 0, so ergibt sich:

	C	A	C	b	C	A
B				+		
c	+		0		+	
a		+				*
B				*		

Also ist die einzige rechte Hälfte, welche nicht lösbar ist, diejenige, die zur Zuordnung zwischen dem empfangenen c und dem mittleren C gehört.

Um nun tatsächlich die möglichen Zuordnungen zu finden, muss man dasselbe noch in umgekehrter Reihenfolge für die linke Hälfte jeder Zuordnung durchführen. Dabei werden in jedem Log alle Zuordnungen mit frühesten Empfangsereignisse nach der Randbedingung als *-möglich markiert und diese Möglichkeitsmarkierungen dann nach rechts und unten verbreitet:

[4]Hierbei unterscheidet sich das oben beschriebene, iterative Vorgehen geringfügig von der rekursiven Variante, bei der nur die Pfade eine Rolle spielen, die im iterativen Vorgehen mit einem diagonalen Pfeil enden. Man sieht aber leicht, dass dies im Ergebnis keinen Unterschied bewirkt, da es zu jedem Pfad, der nicht mit einem diagonalen Pfeil endet, ein analoger Pfad mit Diagonalpfeilende existieren muss, der (stückweise) parallel verläuft.



In der linken Hälfte sind also alle Zuordnungen möglich und, da die insgesamt möglichen Zuordnungen sich aus der $\wedge$-Verknüpfung beider Matrizen ergeben, sind – wie wir bereits wissen – alle Zuordnungen abgesehen von der zwischen dem empfangenen c und dem mittleren C möglich.

Man kann dieses Verfahren optimieren, indem man nur die tatsächlich relevanten (im Bild nicht leeren) Felder speichert. Außerdem fällt auf, dass man während der Berechnung von Solvable_L und Solvable_R nur jeweils die letzte Zeile/Spalte benötigt, so dass man prinzipiell alle Matrixeinträge nacheinander ohne Laufzeiteinbußen in $O(n)$-Speicher berechnen kann. Allerdings benötigt man leider am Schluss alle Werte von beiden Funktionen zugleich, um entscheiden zu können, ob eine Zuordnung zu einer Lösung führt. Man könnte aber gleich eine Datenstruktur verwenden, die angibt, welche Zuordnungen insgesamt möglich sind, und dann während der Berechnung von $\text{Solvable}_{L/R}$ alle ungültigen Zuordnungen streichen. Je nach dem wie gut diese Datenstruktur komprimiert, verringert sich der Speicherverbrauch, möglicherweise aber zum Preis erhöhter Rechenzeit (ein speichermäßig optimaler Fall wären zum Beispiel die Eingabedaten selbst). Als Kompromiss scheint sich eine verkettete Liste von möglichen Intervallen oder eine Baumstruktur zu eignen, was aber in dieser Arbeit nicht weiter untersucht wird.

5.2. Allgemeiner Fall

Nun betrachten wir den allgemeinen Fall von uSRC*, in dem es auch mehr als zwei Logs geben darf. Mit denselben Argumenten wie im vorherigen Abschnitt folgt auch hier, dass eine konsistente und eindeutige GA genau dann existiert, wenn man die Empfangsereignisse mit $r_1, ..., r_{n_R}$ durchnummerieren kann, so dass für $r_i, r_j \in R_k$ mit $i < j$ gilt $r_i \prec_k r_j$ und, falls $A(r_i), A(r_j) \in S_l$ sind, $A(r_i) \prec_l A(r_j)$ (vergleiche wieder das Lemma A.2).

Nunmehr werden die Ereignisse in jedem Log aber durch eine beliebige Zuordnung nicht mehr in zwei Hälften aufgeteilt, sondern ein solcher Schnitt kann nur noch durch $e_1, \ldots, e_{n_L}$ Ereignisse mit $e_i \in L_i$ erfolgen.

Man sieht aber leicht, dass es eine Lösung für die rechte Hälfte eines Schnitts genau dann gibt, wenn es in dieser Hälfte ein erstes Empfangsereignis r_1 mit einem zugehörigen Sendeereignis gibt, so dass die Menge der Ereignisse rechts von diesen beiden und den Ereignissen des Schnitts wiederum lösbar ist (da dort dann $r_2, \ldots, r_{n_R}$ sortierte Empfangsereignisse existieren müssen, so dass $r_1, \ldots, r_n$ ebenfalls eine Lösung bildet).

Ein Beispiel hierzu ist:

$$\begin{array}{cccccccc}
\ldots & e_1 & \ldots & & & & \\
\ldots & e_2 & S & \ldots & S & e'_2 & \ldots \\
\ldots & e_3 & S & \ldots & S & e'_3 & \ldots \\
\ldots & e_4 & \ldots & & & &
\end{array}$$

Dabei sind die beiden neuen, einander zugeordneten Ereignisse mit e'_2 und e'_3 bezeichnet und S steht für ein beliebiges Sendeereignis. Ist nun die rechte Hälfte des Schnitts e_1, e'_2, e'_3, e_4 lösbar, so ist es offensichtlich auch die rechte Hälfte des Schnitts e_1, e_2, e_3, e_4. Umgekehrt, existieren in keinem Log zwei solche neuen Ereignisse, so ist die rechte Hälfte eben nicht lösbar.

Formuliert man dies wieder als rekursive Funktion ergibt sich:

$$
\begin{array}{lll}
\text{Solvable}_R & (e_1, e_2, \ldots, e_{n_L}) & \\
= & (\nexists i : \exists r \in R_i \text{ mit } e_i \prec_i r & \\
& \vee(\exists k, l, e'_1, e'_2, \ldots, e'_{n_L}, \text{ so dass:} & e'_i \in L_i \\
& & \wedge \quad e_i = e'_i \; \forall i \text{ mit } i \neq k,\, i \neq l \\
& & \wedge \quad e_k \prec_k e'_k \wedge e_l \prec_l e'_l \\
& & \wedge \quad P(e'_k) = P(e'_l) \\
& & \wedge \quad D(e'_k) \neq D(e'_l) \\
& & \wedge \quad \nexists i : \exists r \in R_i \text{ mit } e_i \prec_i r \prec_i e'_i \\
& & \wedge \quad \text{Solvable}_R\ (e'_1, e'_2, \ldots, e'_{n_L}))
\end{array}
$$

Und analog für die linke Hälfte:

$$
\begin{array}{lll}
\text{Solvable}_L & (e_1, e_2, \ldots, e_{n_L}) & \\
= & (\nexists i : \exists r \in R_i \text{ mit } r \prec_i e_i & \\
& \vee(\exists k, l, e'_1, e'_2, \ldots, e'_{n_L}, \text{ so dass:} & e'_i \in L_i \\
& & \wedge \quad e_i = e'_i \; \forall i \text{ mit } i \neq k,\, i \neq l \\
& & \wedge \quad e'_k \prec_k e_k \wedge e'_l \prec_l e_l \\
& & \wedge \quad P(e'_k) = P(e'_l) \\
& & \wedge \quad D(e'_k) \neq D(e'_l) \\
& & \wedge \quad \nexists i : \exists r \in R_i \text{ mit } e'_i \prec_i r \prec_i e_i \\
& & \wedge \quad \text{Solvable}_L\ (e'_1, e'_2, \ldots, e'_{n_L}))
\end{array}
$$

Offenbar gibt es maximal n^{n_L} mögliche Parameter und n_L^2 Werte für k und l, so dass sich die Funktionen in $O(n_L^2 \cdot n^{n_L})$ mit $O(n^{n_L})$-Speicher berechnen lassen. (Die e'_i lassen sich für jedes Log vorberechnen.)

Damit r und s mit $P(r) = P(s)$ einander zugeordnet werden können, benötigt man wieder zumindest $e_1, \ldots, e_{n_l}$ Ereignisse, für die gilt $e_i = r$ und $e_j = s$ für ein i, j sowie $\text{Solvable}_L(e_1, \ldots, e_{n_l}) \wedge \text{Solvable}_R(e_1, \ldots, e_{n_l})$.

Im Gegensatz zum vorherigen Abschnitt ist diese Bedingung aber nicht mehr hinreichend, da zu den Ereignissen des Schnittes $e_1, \ldots, e_{n_L}$ nun Empfangsereignisse gehören können, zu denen es keine Sendeereignisse gibt. Es muss daher für jedes Empfangsereignis in der Schnittmenge auch ein passendes Sendeereignis in ihr geben, was man folgendermaßen formulieren kann: Ist $R' = R \cap \{e_1, \ldots, e_{n_L}\}$ und $S' = S \cap \{e_1, \ldots, e_{n_L}\}$, so muss es eine Menge $S'' \subseteq S'$ geben, so dass für jedes $t \in \mathscr{P}$ gilt $|\{r \in R' | P(r) = t\}| = |\{s \in S'' | P(s) = t\}|$.

Dabei spielt die Reihenfolge keine Rolle (denn im Schnitt sind alle Ereignisse gleichzeitig und früher bzw. später als die Ereignisse in den beiden Hälften), so dass diese Bedingung mit einem geeigneten Hash in $O(n_L)$ überprüft werden kann. Für zwei Ereignisse gibt es offenbar $O(n^{n_L-2})$ mögliche Schnitte, welche diese enthalten, so dass man mit den Ergebnissen von Solvable$_{L/R}$ für zwei Ereignisse in $O(n_L \cdot n^{n_L-2})$ überprüfen kann, ob die Zuordnung möglich ist. Insgesamt kann man dann also alle Zuordnungen in $O(n_L^2 \cdot n^{n_L} + n_L \cdot n^{n_L}) = O(n_L^2 \cdot n^{n_L})$ berechnen. Die Berechnung der eigentlichen Zuordnung aus den Matrizen von Solvable$_{L/R}$ spielt also bei der Laufzeit asymptotisch keine Rolle. Es könnte aber aus Geschwindigkeitsgründen trotzdem wünschenswert sein, den Algorithmus ohne eine solche Nachberechnungsphase zu implementieren, indem man ähnlich wie im vorherigen Abschnitt während der Berechnung von Solvable$_{L/R}$ aus einem Set der möglichen Zuordnungspaare alle als ungültig erkannten Paare streicht. Auch die dynamische Programmierung ist weiterhin möglich, indem man die Werte in einem n_L-dimensionalen Hyperwürfel speichert und für jeden Würfeleintrag alle zweidimensionalen Ebenen scannt, die diesen Eintrag enthalten.

Für größere n_L lässt sich dieses Verfahren zwar nicht mehr in realistischer Zeit durchführen, trotzdem ist es um einiges schneller als das naive Durchprobieren aller Kombinationen, welches exponentiell zu n statt n_L wäre. Eine Möglichkeit zur Optimierung, könnte es sein, zuerst ein anderes und schnelleres, aber nicht fehlerfreies, Verfahren (z.B.: die Heuristik aus Kapitel 4) zu verwenden, um die möglichen Lösungsmengen einzuschränken. Dies wird hier aber nicht weiter betrachtet.

Außerdem kann man den Algorithmus auch erweitern, um SRC* statt uSRC* zu lösen. Dazu kann man prinzipiell entweder versuchen in jedem Funktionschritt mehrere Empfangsereignisse zuzuordnen (was die Laufzeit um einen Faktor von $2^{(n_L-1)}$ verlangsamt) oder man wandelt das SRC* in ein uSRC* Problem um, indem man jedes Sendeereignis S eines bestimmten Typs durch $2n_L$-Sendeereignisse $S_1 S_2 \dots S_{2n_L}$ ersetzt. Ersetzt man dann noch jedes Empfangsereignis $r \in R_i$ mit $P(r) = P(s)$ durch $r_i r_{2n_L-i+1}$, wobei $P(S_j) = P(r_k) \Leftrightarrow j = k$ gelten solle, so kann jedes neue Sendeereignis nur von genau einem Knoten empfangen werden. Da in keinem Log ein einzelnes Sendeereignis mehrmals empfangen werden darf, ist damit das SRC*-Problem auf uSRC* reduziert.[5]

[5] Inwiefern sich diese Verdopplung der Zahl der Empfangsereignisse und die »Ver$n_L$fachung« der Sendeereignisse auf die Laufzeit auswirkt, müsste genauer untersucht werden. Das einfache Ersetzten von n durch $n_L \cdot n$ in der O-Notation würde die Laufzeit mit dem Faktor $n_L^{n_L}$ multiplizieren, was schlechter ist als eine Modifikation von Solvable$_{L/R}$.

Kapitel 6.

Zusammenfassung

In dieser Arbeit wurde das Problem des Findens von möglichen Zuordnungen zwischen Sende- und Empfangsereignissen untersucht. Dazu wurden zuerst untere Grenzen für die Laufzeit aller Algorithmen, welche diese Zuordnungen ermitteln können, hergeleitet und anschließend zwei entsprechende Algorithmen präsentiert.

Die Komplexitätsbetrachtungen haben gezeigt, dass alle drei der hier behandelten Probleme (SRC, pSRC und uSRC) NP-vollständig sind. Dazu wurde das für seine NP-Vollständigkeit bekannte X3C-Problem zuerst auf uSRC reduziert und der dadurch entstehende Spezialfall von uSRC wiederum zu einem SRC-Problem umgewandelt. Anschließend wurden noch zwei Reduktionen von SRC und pSRC auf das jeweils andere Problem beschrieben.

Anschließend wurde ein aus der Literatur ([SK09], [Mar08]) bekannter Algorithmus untersucht, von dem vermutet wurde, dass er pSRC in Polynomialzeit löst. Mittels Gegenbeispielen ließ sich aber zeigen, dass es sich nur um eine Heuristik handelt, die keine exakte Lösung finden kann. Nichtsdestotrotz wurden einige Spezialfälle, bei denen die Zahl der vorhandenen Logs stark eingeschränkt wurde, entdeckt, die zu einem sinnvollen Ergebnis führen.

Danach wurde aus einer neuen Definition für die korrekte Lösung ein alternativer Algorithmus entworfen, der uSRC* (eine Verallgemeinerung von uSRC) exakt löst. Obwohl dieser Algorithmus eine exponentielle Laufzeit besitzt, wurde gezeigt, dass er um einiges schneller ist, als das naive Durchprobieren aller Lösungen. Er wurde dabei zuerst für den Spezialfall zweier Logs in rekursiver und iterativer Form beschrieben und anschließend verallgemeinert.

Für die zukünftige Forschung in diesem Gebiet dürfte es vor allem interessant sein, nach schnelleren Lösungsalgorithmen zu suchen sowie die Probleme zu verallgemeinern.

Man könnte beispielsweise versuchen, die beiden Algorithmen dieser Arbeit zu kombinieren. Da die Heuristik bekanntermaßen keine möglichen Zuordnungen ausschließt ([SK09]), kann man mit ihr den Lösungsraum verkleinern. Wendet man dann den exakten Algorithmus auf diesen verkleinerten Lösungsraum an, kann man womöglich trotz dessen exponentieller Laufzeit schnell die korrekte Lösung finden.

Desweiteren enthalten die Logs auch zusätzliche Informationen, nämlich Zeitstempel, die hier vollständig ignoriert wurden. Diese basieren auf – im allgemeinen nicht synchronisierten – physikalischen Uhren. In vielen Fällen kann man jedoch eine maximale Abweichung der Uhren voneinander angeben ([Lam78]), so dass von Anfang an für eine Zuordnung nur die Ereignisse eines gewissen Intervalls möglich sind. Prinzipiell scheint es zudem auch möglich, aus einer teilweisen Lösung zu mehreren Zeitpunkten die exakte zeitliche Abweichung zwischen allen Uhren zu berechnen. Daraus könnte man eine (womöglich nicht-lineare) Funktion konstruieren, welche die Zeiten aller vorhandenen Uhren ineinander umrechnet ([SKR+09]), woraus man eine Lösung für weitere Bereiche der Logs ermitteln kann.

Bei allen Untersuchungen in dieser Arbeit wurde vorausgesetzt, dass es keine Übertragungs- oder Ausbreitungsverzögerung gibt und jede gesendete Nachricht unmittelbar beim Empfänger ankommt. Dies gilt sicherlich für LANs, in größeren Systemen lassen sich diese Verzögerungen jedoch nicht mehr ignorieren. Hierzu sollte man versuchen entsprechende allgemeinere Algorithmen zu entwickeln und die Komplexität des so verallgemeinerten Problems analysieren. [1]

[1] Das hier bezüglich der Heuristik präsentierte Gegenbeispiel würde im allgemeinen Fall nicht funktionieren.

Anhang A.

Beweise für Kapitel Komplexität

A.1. Eindeutige Empfänger und keine Senderambivalenz (uSRC)

In diesem Abschnitt des Anhangs werden die im Abschnitt 3.1 skizzierten Beweise im Detail vorgestellt, wodurch gezeigt wird, dass X3C auf das Finden einer konsistenten und eindeutigen GA reduziert werden kann.

Die Notation von X3C und die Definition der Logs, auf die das Problem reduziert wurde, kann man folglich dort nachlesen.

Wiederholtes Lemma 3.1 *Existiert eine Lösung für uSRC, so existiert eine Lösung für X3C.*

Beweis. Angenommen es existiert eine konsistente und eindeutige GA A.
Da A eindeutig ist, gilt für alle $r_1, r_2 \in R$ mit $A(r_1) = A(r_2)$, dass $r_1 = r_2$ ist, beziehungsweise dass für $r_1 \neq r_2$ dementsprechend $A(r_1) \neq A(r_2)$ gilt.

Betrachten wir nun die Zuordnung der gesendeten und empfangenen Ereignisse vom Typ X:

Zur Vereinfachung des Formalismus heiße $x \in L_i \neq L_0$ mit $P(x) = X$ ab sofort x^i (eindeutig, da es in jedem Log L_i nur ein einziges solches Ereignis gibt)
Offensichtlich gilt für jedes x^i, entweder $A(x^i) \prec \blacktriangle$ oder $\blacktriangle \prec A(x^i)$.
Dann lassen sich die Logs nach dieser Eigenschaft aufteilen, wodurch zwei Indexmengen $I = \{i|A(x^i) \prec \blacktriangle\}$ und $J = \{j|\blacktriangle \prec A(x^j)\}$ definiert werden, für die $I \cap J = \emptyset$ und $I \cup J = \{1, 2, \ldots, n_L\}$ gilt.

Es existieren genau n_L gesendete X und n_L empfangene X, die in jeweils unterschiedlichen Logs liegen, also folgt aus der Eindeutigkeit von A, dass für jedes gesendete X ein Log in der entsprechenden Indexmenge liegt. Da es genau $\frac{m_T}{3}$ gesendete X vor $\blacktriangle$ gibt, muss $|I| = \frac{m_T}{3}$ sein und $|J| = n_L - \frac{m_T}{3}$.

Für jedes $r \in L_i$ mit $i \in I$ und $P(r) \neq X$ gilt, $r \prec_i x^i$, also folgt aus der Konsistenz von A, dass $A(r) \prec A(x^i) \prec \blacktriangle$ gilt. Da sich nur X-se zwischen $\blacktriangleleft$ und $\blacktriangle$ befinden, also $P(s) = X$ für alle $s \in L_0$ mit $\blacktriangleleft \prec s \prec \blacktriangle$ gilt, folgt $A(r) \prec \blacktriangleleft$.

Seien $i, i' \in I$ und $r \in L_i$, $r' \in L_{i'}$ mit $P(r) = P(r') \neq X$ für diesen Absatz fest.
Es gilt offensichtlich $P(A(r)) = P(r) = P(r') = P(A(r'))$ und $A(r), A(r') \prec \blacktriangleleft$.
Da vor $\blacktriangleleft$ jeder Ereignistyp nur ein einziges Mal vorkommt, also $\forall t \neq X : |\{s \in L_0 | s \prec \blacktriangleleft \wedge P(s) = t\}| = 1$ formal geschrieben, folgt daraus $A(r) = A(r')$ und aus der Eindeutigkeit von A folgt $r = r'$ und somit $i = i'$.

Das heißt die Logs sind disjunkt bezüglich der Typen ihrer Ereignisse, also $\{P(r) \neq X | r \in L_i\} \cap \{P(r) \neq X | r \in L_j\} = \emptyset$ für $i, j \in I$ mit $i \neq j$. Da die Typen der Logs den Sets entsprechen, gilt $M_i \cap M_j = \emptyset$ für die ursprünglichen Sets und $i, j \in I$ mit $i \neq j$.

Sei $\hat{L}_i = L_i \setminus \{x^i\}$.
Aus $|\hat{L}_i| = 3$ und der Disjunktivität aller $\hat{L}_i$ untereinander bezüglich der Ereignistypen für alle i, folgt $|\bigcup_{i \in I} \hat{L}_i| = |\hat{L}_i| \cdot |I| = 3 \cdot \frac{m_T}{3} = m_T$.
Da $P(r) \neq P(r')$ für $r \in \hat{L}_i \neq \hat{L}_j \ni r'$ gilt, ist $\bigcup_{i \in I} \{P(r) | r \in \hat{L}_i\} = \{1, 2, \ldots, m_T\}$, woraus folgt $\bigcup_{i \in I} M_i = \{1, 2, \ldots, m_T\}$.

Setzt man nun $\{T_i\} = \{M_i | i \in I\}$, so besitzt man eine gültige Lösung für X3C. $\square$

Wiederholtes Lemma 3.2 *Existiert eine Lösung für X3C, so existiert eine Lösung für uSRC.*

Zudem gibt es zwei Indexmengen I und J mit $I \cap J = \emptyset$ *und* $I \cup J = \{1,\ldots,n_L\}$*, so dass für alle* $r \in R_i = L_i$ *mit* $i \in I$ *gilt* $A(r) \prec \blacktriangle$ *und für alle* $r \in R_j = L_j$ *mit* $j \in J$ *gilt* $\blacktriangle \prec A(r)$*.*

Außerdem sind die Ereignisse innerhalb dieser Mengen von Logs nach Typ und jeweiliger Lognummer sortiert, das heißt, ist $k,k' \in I$ *mit* $k < k'$ *oder* $k,k' \in J$ *mit* $k < k'$*, so gilt für alle* $q \in L_k, r \in L_{k'}$ *mit* $P(q) = P(r)$*, dass* $A(q) \prec A(r)$ *ist.*

Beweis. Sei $\{T_i\} \subseteq \{M_i\}$ mit $T_i \cap T_j = \emptyset$ für $i \neq j$ und $\bigcup_i T_i = \bigcup_i M_i$ eine gegebene Lösung des Coverproblems.

Dann existiert ein I, so dass $\{T_i\} = \{M_i | i \in I\}$ gilt. Sei zudem $J = \{1,\ldots,n_L\} \setminus I$.

Wir bezeichnen wie im vorherigen Beweis $r \in R_i$ mit $P(r) = X$ als x^i und $\hat{L}_i = L_i \setminus x^i$.

Es muss nun gezeigt werden, dass eine eindeutige GA A mit folgenden Eigenschaften existiert:

- Für $i \in I$ muss $A(r) \prec\blacktriangleleft$ für $r \in \hat{L}_i$ und zudem $\blacktriangleleft\prec A(x^i) \prec \blacktriangle$ sein.
- Für $j \in J$ muss $\blacktriangle \prec A(r) \prec\blacktriangleright$ für $r \in \hat{L}_j$ und zudem $\blacktriangleright\prec A(x^j)$ sein.
- Für $i,i' \in I$ mit $i < i'$ muss für alle $q \in L_i, r \in L_{i'}$ mit $P(q) = P(r)$: $A(q) \prec A(r)$ gelten.
- Für $j,j' \in J$ mit $j < j'$ muss für alle $q \in L_j, r \in L_{j'}$ mit $P(q) = P(r)$: $A(q) \prec A(r)$ gelten.

Für die ersten beiden Eigenschaften muss man sich nur überzeugen, dass es in den angegebenen Bereichen genügend Sendeereignisse jedes Typs gibt, wodurch auch folgt, dass die entsprechende Abbildung eine GA darstellt:

Da die T_i disjunkt sind, gibt es für jeden Typ t mit $1 \leq t \leq m_T$ genau ein einziges i und $r \in L_i$ mit $P(r) = i$. Da für jedes t auch ein $s \in L_0$ mit $P(s) = t$ und $s \prec\blacktriangleleft$ existiert,

existiert für alle $r \in L'_i$ mit $i \in I$ die geforderte Zuordnung.

Es gilt zudem $|\bigcup_i T_i| = |\bigcup_i M_i| = m_T$, also ist $|I| = |\{T_i\}| = \frac{m_T}{3}$, da $|T_i| = 3$ für alle i ist und die T_i disjunkt sind. Daraus folgt $\{x^i | i \in I\} = |I| = \frac{m_T}{3} = |\{x \in L_0 | P(x) = X \wedge \blacktriangleleft \prec x \prec \blacktriangle\}|$, weshalb es für die geforderten Zuordnungen von I genügend Sendeereignisse gibt.

Betrachten wir nun die Zuordnungen für die durch J festgelegten Logs: Für jeden Typ t existieren insgesamt $C(t)$ Empfangsereignisse. Da eines dieser Ereignisse in $\bigcup_{i \in I} L_i$ enthalten ist, muss $|\{j \in J | \exists r \in L_j : P(r) = t\}| = C(t) - 1$ sein.

Dies ist genau die Zahl der Sendeereignisse vom Typ t in L_0 im Intervall zwischen $\blacktriangle$ und $\blacktriangleright$ ist, also ist $|\{s \in L_0 | P(s) = t \wedge \blacktriangle \prec s \prec \blacktriangleright\}| = |\{j \in J | \exists r \in L_j : P(r) = t\}|$ und es gibt genügend Sendeereignisse (da von jedem Typ maximal ein Ereignis in jedem Log L_j sein kann).

Schließlich gilt auch $|\{x \in L_0 | P(x) = X \wedge \blacktriangleright \prec x\}| = n_L - \frac{m_T}{3} = |J|$, also existiert für jedes von einem Log aus J empfangene X ein entsprechendes Sendeereignis.

Also existiert eine eindeutige GA A mit den ersten beiden oben verlangten Eigenschaften.

Die letzten beiden Eigenschaften folgen trivial daraus, dass die Sendeereignisse mit gleichem Typ in L_0 vor und nach $\blacktriangle$ nur in Blöcken auftauchen. Gibt es also zwei $i, i' \in I$ mit $i < i'$ und $q \in L_i, r \in L_{i'}$ mit $P(q) = P(r)$, so dass gilt $A(r) \prec A(q)$, so kann man die Zuordnungen von r und q vertauschen, also eine neue GA A' mit $A'(r) = A(q)$ und $A'(q) = A(r)$ definieren, die dann die Bedingung erfüllt (und offensichtlich ist A' weiterhin konsistent, sofern es A war).

Nun ist nur noch zu zeigen, dass A zusammen mit der festgelegten GTO $\prec = \prec_0$ tatsächlich konsistent ist, also dass aus $e_1 \prec_n e_2$ für $e_1, e_2 \in L_n$ folgt $A(e_1) \prec A(e_2)$.

Für alle $s_1, s_2 \in L_0$ mit $s_1 \prec_0 s_2$ gilt $A(s_1) = s_1 \prec_0 s_2 = A(s_2)$ und somit wie erfordert $A(s_1) \prec A(s_2)$.

Für $i \in I$ gilt: Ist $e_1, e_2 \in L_i$ mit $e_1 \prec_i e_2$ gegeben, so ist $P(e_1) < P(e_2)$. Also gilt $P(A(e_1)) < P(A(e_2))$.

Offensichtlich gilt für alle $s_1, s_2 \in L_0$ mit $s_1, s_2 \prec \blacktriangle$, dass $s_1 \preceq_0 s_2 \Leftrightarrow P(s_1) \leq P(s_2)$ gilt. Also folgt aus $P(A(e_1)) < P(A(e_2))$, dass $A(e_1) \preceq_0 A(e_2)$ gilt und somit mit der Bedingung der Eindeutigkeit, $A(e_1) \prec A(e_2)$.

Für $j \in J$ gilt ebenfalls: Ist $e_1, e_2 \in L_j$ mit $e_1 \prec_j e_2$ gegeben, so ist $P(e_1) < P(e_2)$. Also gilt $P(A(e_1)) < P(A(e_2))$. Auch für alle $s_1, s_2 \in L_0$ mit $\blacktriangle \prec s_1, s_2 \prec \blacktriangleright$ gilt offenbar $s_1 \preceq_0 s_2 \Leftrightarrow P(s_1) \leq P(s_2)$. Damit folgt $A(e_1) \preceq_0 A(e_2)$ und mit der Eindeutigkeit wieder $A(e_1) \prec A(e_2)$.

Somit ist A konsistent.

Es existiert also eine eindeutige, konsistente GA als Lösung für uSRC.

□

A.2. Keine Senderambivalenz (SRC)

Hier werden nun detailliert die Beweise von Abschnitt 3.2 präsentiert, die zeigen, dass X3C auch auf das Finden einer konsistenten GA reduziert werden kann.

Die dazu konstruierten Logs kann man im Abschnitt 3.2 nachlesen, die Notation von X3C im Abschnitt 3.1, außerdem wurden dort die Logs definiert, auf denen die Logs von 3.2 basieren.

Da für die Reduktion eine große Anzahl von Sendeereignissen nötig ist, durch die es kompliziert wird, eine GTO zu definieren, ersetzten wir die GTO, als Ordnungsrelation über den Sendeereignissen, durch eine Ordnungsrelation über den Empfangsereignissen: Diese heißt dementsprechend global receiving order (GRO), wird mit dem Symbol $\lessdot$ dargestellt und ist eine partielle Ordnungsrelation über $R \times R$. Sie entspricht der GTO, indem sie eine globale Reihenfolge der Ereignisse festlegt, allerdings eben die Reihenfolge der Empfangsereignisse anstatt der Sendeereignisse.

Wie bei der GTO gibt es auch für die GRO eine Konsistenzbedingung, bevor man diese aber formulieren kann, benötigt man den Begriff einer inversen GA:

Lemma A.1. *Ist A eine eindeutige GA, so existiert eine inverse GA $A^{-1} : R \cup A(R) \to R$, so dass für alle $r \in R$ gilt:*

$$A^{-1}(A(r)) = A^{-1}(r) = r$$

Beweis. Es ist nur zu zeigen, dass die obige, implizite Definition von A^{-1} widerspruchsfrei ist, also dass nach der Definition aus $e_1 = e_2$ tatsächlich folgt $A^{-1}(e_1) = A^{-1}(e_2)$.
Für $e_1 = e_2 \in R$ ist dies offensichtlich der Fall.
Sind $e_1, e_2 \in A(R) \subseteq S$, so gibt $r_1, r_2 \in R$ mit $e_1 = A(r_1)$ und $e_2 = A(r_2)$.
Da A eindeutig ist, ist $r_1 = r_2$ und somit $A^{-1}(e_1) = A^{-1}(A(r_1)) = r_1 = r_2 = A^{-1}(A(r_2)) = A^{-1}(e_2)$ □

Nun kann man die Konsistenzbedingung für eine GRO angeben, die lautet:

Lemma A.2. *Ist eine eindeutige GA A mit ihrer Umkehrfunktion A^{-1} gegeben und erfüllt $\lessdot$ die Eigenschaft*

$$e_1, e_2 \in L_n \cap (R \cup A(R)) : (e_1 \prec_n e_2 \Rightarrow A^{-1}(e_1) \lessdot A^{-1}(e_2))$$

so ist A konsistent.

Beweis. Es wird mittels Induktion über die Zahl der nicht zugeordneten Sendeereignisse gezeigt, dass sich aus der GRO $\lessdot$ eine GTO $\prec$ konstruieren lässt, so dass das Paar $A, \prec$ konsistent ist:

Induktion über $|M| = |\{s \in S | \forall r : A(r) \neq s\}|$

Induktionsanfang $|M| = 0$:
Dann ist A^{-1} eingeschränkt auf die Menge der Sendeereignisse (Also $A^{-1} : S \to R$) eine bijektive Abbildung zwischen allen Sende- und allen Empfangsereignissen, und somit lässt sich $\prec$ wohldefinieren als: Für $s_1, s_2 \in S$ gilt $s_1 \prec s_2$ genau dann falls $A^{-1}(s_1) \lessdot A^{-1}(s_2)$ erfüllt ist.

Äquivalent dazu ist folgende Formulierung: Für $r_1, r_2 \in R$ sei $A(r_1) \prec A(r_2) \Leftrightarrow r_1 \lessdot r_2$

Damit erhält man tatsächlich eine Ordnungsrelation:

Da A eindeutig und $\lessdot$ irreflexive ist, gibt es keine r_1, r_2 mit $A(r_1) = A(r_2)$ und $r_1 \lessdot r_2$, also ist $\prec$ irreflexive.

Aus $A(r_1) \prec A(r_2)$ folgt $r_1 \lessdot r_2$ und, da $\lessdot$ antisymmetrisch ist, folgt daraus, dass nicht $r_2 \lessdot r_1$ gilt und somit auch nicht $A(r_2) \prec A(r_1)$. Daher ist $\prec$ ebenfalls antisymmetrisch.

Gilt $s_1 \prec s_2$ und $s_2 \prec s_3$, so gibt es eindeutige $r_1, r_2, r_3 \in R$ mit $r_1 \lessdot r_2$ und $r_1 \lessdot r_3$

sowie $s_1 = A(r_1)$, $s_2 = A(r_2)$ und $s_3 = A(r_3)$. Da $\lessdot$ transitiv ist, gilt $r_1 \lessdot r_3$ und somit $s_1 \prec s_3$, weshalb auch $\prec$ transitiv ist.

Gilt $e_1 \prec_n e_2$ so ist $A^{-1}(e_1) \lessdot A^{-1}(e_2)$, woraus folgt $A(A^{-1}(e_1)) \prec A(A^{-1}(e_2))$
Ist $e_1 \in R$, so ist $A^{-1}(e_1) = e_1$ und $A(A^{-1}(e_1)) = A(e_1)$.
Ist dagegen $e_1 \in S$, so ist $A(A^{-1}(e_1)) = e_1 = A(e_1)$.
Entsprechend ist auch immer $A(A^{-1}(e_2)) = A(e_2)$ also folgt aus $e_1 \prec_n e_2$, dass $A(e_1) \prec A(e_2)$ gilt und das Paar $(\prec, A)$ konsistent ist.

Also ist A konsistent.

Induktionsschritt: $|M| \to |M| + 1$
Sei $s \in S_n$ mit $A(r) \neq s$ für alle $r \in R$ ein Sendeereignis, welches niemals empfangen wurde. Sei $\bar{L} = L \setminus \{s\}$ die Menge der Ereignisse ohne dieses s. Die GA A eingeschränkt auf $\bar{L}$ heiße $\bar{A}$ und ist offensichtlich eindeutig, da die Einschränkung bloß die Abbildung $A(s) = s$ entfernt. Dementsprechend existiert eine Umkehrabbildung $\bar{A}^{-1}$.
Alle weiteren Bezeichner wie $\bar{S}$, $\bar{S}_n$, ... seien analoge Einschränkungen von $S, S_n, \ldots$

Weiterhin erfüllt $\lessdot$ offenkundig die Bedingung, dass für $e_1, e_2 \in \bar{L}_n \cap (R \cup A(R)) \subset L_n \cap (R \cup A(R))$ mit $e_1 \prec_n e_2$ gilt $\bar{A}^{-1}(e_1) \lessdot \bar{A}^{-1}(e_2)$, da dies für $e_1, e_2 \in L_n \cap (R \cup A(R))$ gilt.
Also existiert nach der Induktionsannahme eine GTO $\bar{\prec}$, so dass das Paar $A, \bar{\prec}$ konsistent ist, da $|\bar{M}| = |M| - 1 < |M|$ ist.

Wir definieren nun eine neue Ordnungsrelation $\prec$, für die erstmal gelte $a \prec b \Leftrightarrow a \bar{\prec} b$ für $a \neq s \neq b$
Es lassen sich nun drei Fälle unterscheiden:

Fall 1: $s \prec_n e$ für alle $e \in \bar{L}_n$
Dann sei $s \prec e$ für alle $t \in \bar{S}$
Daraus folgt automatisch $A(s) \prec A(e')$ für alle $e' \in \bar{L}_n$ mit $s \prec_n e'$

Fall 2: $e \prec_n s$ für alle $e \in \bar{L}_n$
Dann sei $t \prec s$ für alle $t \in \bar{S}$
Daraus folgt automatisch $A(e') \prec A(s)$ für alle $e' \in \bar{L}_n$ mit $e' \prec_n s$

Fall 3: Es gibt $e \prec_n s \prec_n e'$ mit $e, e' \in \bar{L}_n$

Sei e maximal und e' minimal, also es gebe keine $f \in \bar{L}_n$ mit $e \bar{\prec}_n f \bar{\prec}_n s$ oder $s \bar{\prec}_n f \bar{\prec}_n e'$

Dann sei $A(e) \prec s \prec A(e')$.

Außerdem sei $A(f) \prec s$ für alle $f \in \bar{L}$ mit $A(f) \bar{\prec} A(e)$ und $s \prec f'$ für alle $f' \in \bar{L}$ mit $A(e') \bar{\prec} A(f')$

Daraus folgt: Ist ein beliebiges $f \in L_n$ mit $f \prec_n s$ gegeben, so gilt $A(f) \prec A(s)$ und ist ein beliebiges f' mit $s \prec_n f'$ gegeben, so gilt $A(s) \prec A(f')$

In jedem Fall ist das definierte $\prec$ mit A offensichtlich konsistent, da $\bar{\prec}$ konsistent ist und auch das eingefügte s die Konsistenzbedingung erfüllt. Es ist auch wohldefiniert, da in $\prec$ bisher keine Aussagen über s getroffen wurden und die neuen Bedingungen widerspruchsfrei sind.

Zu zeigen ist nur noch, dass $\prec$ auch tatsächlich eine Ordnungsrelation ist:

Irreflexiv und antisymmetrisch ist trivial, wenn man bedenkt, dass $\bar{\prec}$ eine Ordnungsrelation ist und in allen drei Fällen nur irreflexive und antisymmetrische Beziehungen hinzugefügt wurden.

Auch die Transitivität ist in den ersten beiden Fällen offensichtlich, aber für den letzten muss man diese drei Möglichkeiten betrachten: ($e, e' \in L_n$ seien dabei wie oben das maximale und minimale Ereignis mit $e \prec_n s \prec_n e'$, und s_1, s_2 seien in $\bar{S}$)

Fall 1: $s \prec s_1$ und $s_1 \bar{\prec} s_2$

Ist $s_1 = A(e')$, so ist $A(e') \bar{\prec} s_2$ und somit $s \prec s_2$.

Ansonsten ist $A(e') \bar{\prec} s_1 \bar{\prec} s_2$ und somit $A(e') \bar{\prec} s_2$ nach der Transitivität von $\bar{\prec}$, also gilt $s \prec s_2$

Fall 2: $s_1 \prec s$ und $s \prec s_2$

Es gilt entweder $s_1 \bar{\prec} A(e)$ oder $s_1 = A(e)$ sowie $A(e') \bar{\prec} s_2$ oder $A(e') = s_2$

Aus $A(e) \bar{\prec} A(e')$ folgt somit $s_1 \prec s_2$

Fall 3: $s_1 \bar{\prec} s_2$ und $s_2 \prec s$

Ist $s_2 = A(e)$, so ist $s_1 \bar{\prec} A(e)$ und somit $s_1 \prec s$.

Ansonsten ist $s_1 \bar{\prec} s_2 \prec A(e)$ und somit $s_1 \bar{\prec} A(e)$ nach der Transitivität von $\bar{\prec}$, also gilt $s_1 \prec s$

Also ist die so definierte Ordnungsrelation $\prec$ tatsächlich eine GTO und zusammen mit A auch konsistent. $\square$

Damit können wir nun die im Abschnitt 3.2 nur zusammengefassten Beweise konkretisieren. Das erste Lemma lautete:

Wiederholtes Lemma 3.3 *Jede konsistente Lösung von L' ist eindeutig.*

Beweis. Angenommen A ist nicht eindeutig, das heißt es gibt $r_1, r_2 \in R'$ mit $r_1 \neq r_2$ und $A(r_1) = A(r_2)$. Daraus folgt $P(r_1) = P(r_2)$ und somit $r_1, r_2 \in R$, da es keine zwei Empfangsereignisse in $R' \setminus R$ gibt, die denselben Typ haben. Außerdem haben offensichtlich alle Empfangsereignisse innerhalb desselben Log einen unterschiedlichen Typ, so dass zwei Logs L_i, L_j mit $i \neq j$ existieren, so dass $r_1 \in L_i$ und $r_2 \in L_j$ ist.

Es gibt nun $e_{ij} \in L'_j, e_{ji} \in L'_i$, für die jeweils genau vier Sendeereignisse existieren, nämlich: $E^1_{ij}, E^2_{ij}, E^3_{ij}, E^4_{ij}$ sowie $E^1_{ji}, E^2_{ji}, E^3_{ji}, E^4_{ji}$.

Diese sind nach der Definition der Logs folgendermaßen angeordnet:

$$E^1_{ij} \prec_i E^2_{ij} \prec_i e_{ji} \prec_i r_1 \prec_i E^3_{ij} \prec_i E^4_{ij} \in L_i$$

$$E^1_{ji} \prec_j E^2_{ji} \prec_j e_{ij} \prec_j r_2 \prec_j E^3_{ji} \prec_j E^4_{ji} \in L_j$$

Da es keine weiteren Sendeereignisse gibt, ist offensichtlich $A(e_{ij}) \in \{E^1_{ij}, E^2_{ij}, E^3_{ij}, E^4_{ij}\}$ und $A(e_{ji}) \in \{E^1_{ji}, E^2_{ji}, E^3_{ji}, E^4_{ji}\}$

Fall 1: $A(e_{ij}) \in \{E^1_{ij}, E^2_{ij}\}$ und $A(e_{ji}) \in \{E^1_{ji}, E^2_{ji}\}$
Aus $E^2_{ij} \prec_i e_{ji}$ folgt $E^2_{ij} = A(E^2_{ij}) \prec A(e_{ji}) \preceq_j E^2_{ji}$, also $E^2_{ij} \prec E^2_{ji}$
Aus $E^2_{ji} \prec_j e_{ij}$ folgt $E^2_{ji} = A(E^2_{ji}) \prec A(e_{ij}) \preceq_i E^2_{ij}$, also $E^2_{ji} \prec E^2_{ij}$
Aber es kann nicht gleichzeitig $E^2_{ij} \prec E^2_{ji}$ und $E^2_{ji} \prec E^2_{ij}$ gelten. $\lightning$

Fall 2: $A(e_{ij}) \in \{E^1_{ij}, E^2_{ij}\}$ und $A(e_{ji}) \in \{E^3_{ji}, E^4_{ji}\}$
Aus $r_2 \prec_j E^3_{ji}$ folgt $A(r_2) \prec A(E^3_{ji}) = E^3_{ji} \preceq_j A(e_{ji})$, also $A(r_2) \prec A(e_{ji})$

Aus $e_{ji} \prec_i r_1$ folgt $A(e_{ji}) \prec A(r_1)$
Also $A(r_2) \prec A(r_1)$ und somit $A(r_1) \neq A(r_2)$ ↯

Fall 3: $A(e_{ij}) \in \{E_{ij}^3, E_{ij}^4\}$ und $A(e_{ji}) \in \{E_{ji}^1, E_{ji}^2\}$
Symmetrisch zum vorherigen Fall

Fall 4: $A(e_{ij}) \in \{E_{ij}^3, E_{ij}^4\}$ und $A(e_{ji}) \in \{E_{ji}^3, E_{ji}^4\}$
Mit $e_{ji} \prec_i E_{ij}^3$ folgt $E_{ji}^3 \preceq_j A(e_{ji}) \prec A(E_{ij}^3) = E_{ij}^3$, also $E_{ji}^3 \prec E_{ij}^3$.
Mit $e_{ij} \prec_i E_{ji}^3$ folgt $E_{ij}^3 \preceq_i A(e_{ij}) \prec A(E_{ji}^3) = E_{ji}^3$, also $E_{ij}^3 \prec E_{ji}^3$.
Aber es kann nicht gleichzeitig $E_{ij}^3 \prec E_{ji}^3$ und $E_{ji}^3 \prec E_{ij}^3$ gelten.↯

Daraus folgt, dass die Annahme falsch war, und somit jede konsistente GA A zu L' auch eine eindeutige GA ist. □

Die beiden letzten Lemmas diesen Abschnitts zeigen die Äquivalenz der Logs L aus Abschnitt 3.1 und der Logs L' aus Abschnitt 3.2:

Wiederholtes Lemma 3.4 *Ist A eine zu L' konsistente GA, so ist A eingeschränkt auf L eine konsistente GA zu L.*

Beweis. Sei ein konsistentes Paar $A, \prec$ gegeben. Da $\prec$ eine Ordnungsrelation ist, ist die Einschränkung von $\prec$ auf L ebenfalls eine.

Sei $L'' = L' \setminus L$ die Menge, der zu L hinzugefügten Ereignisse. Offensichtlich haben die in L' hinzugefügten Ereignisse einen anderen Packettyp als alle Ereignisse in L, also $P(e_1) \neq P(e_2)$ für $e_1 \in L, e_2 \in L''$. Daraus folgt, ist $e_1 \in L$ gegeben, so muss $P(A(e_1)) = P(e_1)$ sein und somit ist $A(e_1) \in L$. Daraus folgt, dass A eingeschränkt auf L eine Abbildung von L nach S ist (und nicht nach S').

Dass A eingeschränkt auf L auch die anderen Eigenschaften einer konsistenten GA hat, ist ebenfalls trivial zu zeigen:

$$\forall s \in S' : A(s) = s \Rightarrow \forall s \in S' \setminus L'' : A(s) = s$$

$$\forall e \in L' : P(e) = P(A(e)) \Rightarrow \forall e \in L' \setminus L'' : P(e) = P(A(e))$$

$$\forall r \in R' : N(r) \neq N(A(r)) \Rightarrow \forall r \in R' \setminus L'' : N(r) \neq N(A(r))$$

$$(\forall n \in N : \forall e_1, e_2 \in L'_n : (e_1 \prec_n e_2 \Rightarrow A(e_1) \prec A(e_2)))$$
$$\Rightarrow (\forall n \in N : \forall e_1, e_2 \in L'_n \setminus L'' : (e_1 \prec_n e_2 \Rightarrow A(e_1) \prec A(e_2))) \qquad \square$$

Wiederholtes Lemma 3.5 *Existiert eine eindeutige, konsistente Lösung zu L, so gibt es eine Lösung für L'.*

Beweis. Sei $A, \prec$ die eindeutige, konsistente Lösung für L und A' die im Beweisanfang von Lemma 3.5 definierte Abbildung.

Offensichtlich ist dieses A' eindeutig, da es für alle $r_1, r_2 \in R' \setminus R$ mit $A'(r_1) = A'(r_2)$ ein Tripel m, i, j gibt, so dass $A'(r_1) = A'(r_2) = E^m_{ij}$ und somit $r_1 = e_{ij} = r_2$ ist

Außerdem ist A' eine GA, da es alle unter der entsprechenden Definition 2.1 aufgeführten Eigenschaften erfüllt:

1. $\forall s \in S' : A'(s) = s$
   Dies ist trivial zu sehen

2. $\forall e \in L' : P(e) = P(A'(e))$
   Dies folgt daraus, dass A eine GA ist und $P(e_{ij}) = P(E_{ij})$ für alle i, j gilt

3. $\forall r \in R' : N(r) \neq N(A'(r))$
   Dies folgt daraus, dass A eine GA ist und, dass für $e_{ij} \in L'$ gilt $i \neq j$ sowie $e_{ij} \in L'_j$ und $E_{ij} \in L'_i$

Um zu zeigen, dass A' konsistent ist, muss eine Ordnungsrelation über den Ereignissen definiert werden. Leider ist es durch die vielen, nicht zugeordneten Sendeereignisse kompliziert direkt eine GTO anzugeben. Daher wird zuerst eine GRO $\lessdot$ definiert, welche die Empfangsereignisse in L'_I und L'_J getrennt nach Typ und Nummer des sie enthaltenden Logs sortiert:

Seien $r_1, r_2 \in R'$ gegeben. Dann gelte $r_1 \lessdot r_2$ genau dann wenn eine der folgenden Bedingungen erfüllt ist (mindestens oder genau eine, macht dabei keinen Unterschied):

1. $r_1 \in L'_i, r_2 \in L'_j$ mit $i \in I,\ j \in J$
2. $r_1 \in L'_i, r_2 \in L'_{i'}$ mit $i, i' \in I$ und $B(r_1) < B(r_2)$
3. $r_1 \in L'_i, r_2 \in L'_{i'}$ mit $i, i' \in I$ und $B(r_1) = B(r_2)$ sowie $i < i'$
4. $r_1 \in L'_i, r_2 \in L'_{i'}$ mit $i, i' \in I$ und $B(r_1) = B(r_2)$ sowie $i = i'$ und $r_1 \prec_i r_2$
5. $r_1 \in L'_j, r_2 \in L'_{j'}$ mit $j, j' \in J$ und $B(r_1) < B(r_2)$
6. $r_1 \in L'_j, r_2 \in L'_{j'}$ mit $j, j' \in J$ und $B(r_1) = B(r_2)$ sowie $j < j'$
7. $r_1 \in L'_j, r_2 \in L'_{j'}$ mit $j, j' \in J$ und $B(r_1) = B(r_2)$ sowie $j = j'$ und $r_1 \prec_j r_2$

Geht man diese Bedingungen durch, sieht man leicht, dass $\lessdot$ tatsächlich eine Ordnungsrelation ist:
Für jedes Paar Empfangsereignisse r_1, r_2 kann genau eine Bedingung erfüllt sein, jede Bedingung ist selbst antisymmetrisch und keine andere Bedingung ist symmetrisch zu einer anderen, so dass $\lessdot$ insgesamt antisymmetrisch ist.
Außerdem ist es irreflexiv, da im Fall $r_1 = r_2$ keine Bedingung erfüllt ist.
Außerdem ist $\lessdot$ transitiv, da jede der sieben Bedingungen einer transitiven Ordnungsrelation entspricht, wobei bei Gleichheit in dieser Relation die anderen Bedingungen zur Unterscheidung verwendet werden. Detaillierter lässt sich die Transitivität zeigen, indem man alle Möglichkeiten aufzählt: Sei $r_1, r_2, r_3 \in R'$ mit $r_1 \lessdot r_2$ und $r_2 \lessdot r_3$ gegeben. Aus $r_1 \lessdot r_2$ folgt, dass eine der obigen Bedingungen erfüllt ist:

Fall 1: $r_1 \in L'_i, r_2 \in L'_j$ mit $i \in I,\ j \in J$.
Dann folgt aus $r_2 \in L'_j$ und $r_2 \lessdot r_3$, dass für r_2, r_3 eine der letzten drei Bedingungen erfüllt ist und $r_3 \in L'_{j'}$ mit $j' \in J$ gilt. Also ist $r_1 \lessdot r_3$.

Fall 2: $r_1 \in L'_i, r_2 \in L'_{i'}$ mit $i, i' \in I$ und $B(r_1) < B(r_2)$.
Ist $r_3 \in L'_j$ mit $j \in J$, so ist offensichtlich $r_1 \lessdot r_3$ erfüllt.

Ist dagegen $r_3 \in L'_{i''}$ mit $i'' \in I$, so muss $r_2 \lessdot r_3$ durch eine der Bedingungen 2 bis 4 gegeben sein und es gilt $B(r_3) \geq B(r_2) > B(r_1)$.
Also ist $B(r_1) < B(r_3)$ und $r_1 \lessdot r_3$.

Fall 3: $r_1 \in L'_i, r_2 \in L'_{i'}$ mit $i, i' \in I$ und $B(r_1) = B(r_2)$ sowie $i < i'$.
Ist $r_3 \in L'_j$ mit $j \in J$, so ist offensichtlich $r_1 \lessdot r_3$ erfüllt.
Ist dagegen $r_3 \in L'_{i''}$ mit $i'' \in I$, so muss $r_2 \lessdot r_3$ durch eine der Bedingungen 2 bis 4 gegeben sein.
Ist $B(r_3) > B(r_2) = B(r_1)$ so ist $r_1 \lessdot r_3$ nach Bedingung 2.
Ansonsten ist $B(r_3) = B(r_2) = B(r_1)$ und $i' \leq i''$, also $i < i' \leq i''$ und $r_1 \lessdot r_3$ nach Bedingung 3.

Fall 4: $r_1 \in L'_i, r_2 \in L'_{i'}$ mit $i, i' \in I$ und $B(r_1) = B(r_2)$ sowie $i = i'$ und $r_1 \prec_i r_2$.
Ist $r_3 \in L'_j$ mit $j \in J$, so ist offensichtlich $r_1 \lessdot r_3$ erfüllt.
Ist dagegen $r_3 \in L'_{i''}$ mit $i'' \in I$, so muss $r_2 \lessdot r_3$ durch eine der Bedingungen 2 bis 4 gegeben sein.
Ist $B(r_3) > B(r_2) = B(r_1)$ so ist $r_1 \lessdot r_3$ nach Bedingung 2.
Ist $B(r_3) = B(r_2) = B(r_1)$ und $i' < i''$, so ist $i = i' < i''$ und $r_1 \lessdot r_3$ nach Bedingung 3.
Ist $B(r_3) = B(r_2) = B(r_1)$, $i' = i''$ und $r_2 \prec_i r_3$, so ist $r_1 \prec_i r_2 \prec_i r_3$ und somit $r_1 \lessdot r_3$ nach Bedingung 4.

Fall 5: $r_1 \in L'_j, r_2 \in L'_{j'}$ mit $j, j' \in J$ und $B(r_1) < B(r_2)$.
Dann muss $r_3 \in L'_{j''}$ mit $j'' \in J$ sein, also ist $r_2 \lessdot r_3$ durch eine der Bedingungen 5 bis 7 gegeben. Also gilt $B(r_3) \geq B(r_2) > B(r_1)$ und $r_1 \lessdot r_3$.

Fall 6: $r_1 \in L'_j, r_2 \in L'_{j'}$ mit $j, j' \in J$ und $B(r_1) = B(r_2)$ sowie $j < j'$
Dann muss $r_3 \in L'_{j''}$ mit $j'' \in J$ sein, also ist $r_2 \lessdot r_3$ durch eine der Bedingungen 5 bis 7 gegeben.
Ist $B(r_3) > B(r_2) = B(r_1)$ so ist $r_1 \lessdot r_3$ nach Bedingung 5.
Ansonsten ist $B(r_3) = B(r_2) = B(r_1)$ und $j' \leq j''$, also $j < j' \leq j''$ und $r_1 \lessdot r_3$ nach Bedingung 6.

Fall 7: $r_1 \in L'_j, r_2 \in L'_{j'}$ mit $j, j' \in J$ und $B(r_1) = B(r_2)$ sowie $j = j'$ und $r_1 \prec_j r_2$.
Dann muss $r_3 \in L'_{j''}$ mit $j'' \in J$ sein, also ist $r_2 \lessdot r_3$ durch eine der Bedingungen

5 bis 7 gegeben.
Ist $B(r_3) > B(r_2) = B(r_1)$ so ist $r_1 \lessdot r_3$ nach Bedingung 5.
Ist $B(r_3) = B(r_2) = B(r_1)$ und $j' \leq j''$, so ist $j < j' \leq j''$ und $r_1 \lessdot r_3$ nach Bedingung 6.
Ist $B(r_3) = B(r_2) = B(r_1)$, $j' = j''$ und $r_2 \prec_j r_3$, so ist $r_1 \prec_j r_2 \prec_j r_3$ und somit $r_1 \lessdot r_3$ nach Bedingung 7.

$\lessdot$ ist also tatsächlich eine Ordnungsrelation.

Es bleibt nur zu zeigen, dass $\lessdot$ zusammen mit A' konsistent ist, das heißt für alle $e_1, e_2 \in L'_n \cap (R' \cup A'(R'))$ gilt $e_1 \prec_n e_2 \Rightarrow A'^{-1}(e_1) \lessdot A'^{-1}(e_2)$.
Je nachdem, ob es sich bei e_1, e_2 um Sende- oder Empfangsereignisse handelt, zerfällt der obige Ausdruck in diese vier Fälle:

- $r_1, r_2 \in R'_n : r_1 \prec_n r_2 \Rightarrow r_1 \lessdot r_2$
- $s_1 \in S'_n, r_2 \in R'_n : s_1 \prec_n r_2 \wedge s_1 = A'(r_1) \Rightarrow r_1 \lessdot r_2$
- $s_2 \in S'_n, r_1 \in R'_n : r_1 \prec_n s_2 \wedge s_2 = A'(r_2) \Rightarrow r_1 \lessdot r_2$
- $s_1, s_2 \in S'_n : s_1 \prec_n s_2 \wedge s_1 = A'(r_1) \wedge s_2 = A'(r_2) \Rightarrow r_1 \lessdot r_2$

Für jeden dieser Fälle muss nun gezeigt werden, dass die Eigenschaft tatsächlich erfüllt ist:

- Sind $r_1, r_2 \in R'_n$ mit $n \in I \cup J$ gegeben, so folgt aus $r_1 \prec_n r_2$ nach der Konstruktion der Logs, dass $B(r_1) \leq B(r_2)$ ist, und somit nach Bedingung 2, 4, 5 oder 7 der Definition von $\lessdot$ gilt $r_1 \lessdot r_2$.

- Sind $r_1 \in R'_k$ und $s_2 \in S'_k$ mit $r_1 \prec_k s_2$ gegeben, so hat s_2 die Form $s_2 = E^m_{kl}$ mit $A'^{-1}(E^m_{kl}) = e_{kl} \in R'_l$
  Offenbar ist $m > 1$, da es keine Empfangsereignisse vor dem Block der E^1-Sendeereignisse gibt.
  Ist $m = 2$, so ist $B(r_1) < B(s_2) = B(e_{kl})$ nach der Konstruktion von L'_k und $k, l \in J$ nach der Konstruktion von A', also ist $r_1 \lessdot e_{kl}$.

Ist $m = 3$, so ist $B(r_1) \leq B(s_2) = B(e_{kl})$ nach der Konstruktion von L'_k und $k < l$ sowie $k \in I \Leftrightarrow l \in I$ nach der Definition von A', also ist $r_1 \lessdot e_{kl}$.
Ist $m = 4$, so ist $k \in I$, $l \in J$ und somit $L'_k \ni r_1 \lessdot e_{kl} \in L'_l$.

- Sind $s_1 \in S'_k$ und $r_2 \in R'_k$ mit $s_1 \prec_k r_2$ gegeben, so hat s_1 die Form $s_1 = E^m_{kl}$ mit $A'^{-1}(E^m_{kl}) = e_{kl} \in R'_l$.
  Offenbar ist $m < 4$, da es keine Empfangsereignisse nach dem Block der E^4-Sendeereignisse gibt.
  Ist $m = 1$, so ist gilt $l \in I$ und entweder $k \in J$ oder $k \in I$ mit $k > l$. Im ersten Fall ist $e_{kl} \lessdot r_2$ trivial, im zweiten ist $B(e_{kl}) = B(r_2) = X$, da die Ereignistypen aller durch I bestimmten Logs bis auf die abschließenden X disjunkt sind. Damit und $k > l$ folgt $e_{kl} \lessdot r_2$.
  Ist $m = 2$, so ist $B(e_{kl}) = B(s_1) \leq B(r_2)$ nach der Konstruktion von L' sowie $k, l \in J$ und $k > l$ nach der Konstruktion von A', also ist $L'_l \ni e_{kl} \lessdot r_2 \in L_k$.
  Ist $m = 3$, so ist $B(e_{kl}) = B(s_1) < B(r_2)$ nach der Konstruktion von L' und $k \in I \Leftrightarrow l \in I$ nach der Konstruktion von A', also ist $e_{kl} \lessdot r_2$.

- Sind $s_1, s_2 \in S'_n$ mit $n \in I \cup J$, $s_1 = A'(r_1)$, $s_2 = A'(r_2)$ und $s_1 \prec_n s_2$ gegeben, so haben sie die Form $s_1 = E^m_{nk}$ und $s_2 = E^{m'}_{nl}$. Die dazugehörigen Empfangsereignisse haben dementsprechend die Form $r_1 = e_{nk} \in R'_k$ und $r_2 = e_{nl} \in R'_l$.

  Ist $m = 1$ so ist $k \in I$ und es gilt entweder $n \in J$ oder $n \in I$ mit $n > k$ nach der Konstruktion von A'. Im Unterfall $m' = 1$ folgt $l \in I$ nach der Definition von A' und aus $s_1 \prec_n s_2$ folgt nach der Konstruktion von L'_n, dass entweder $B(E^1_{nk}) < B(E^1_{nl})$ oder $B(E^1_{nk}) = B(E^1_{nl})$ mit $k < l$ gilt, da alle Sendeereignisse mit Index 1 in einem sortierten Block liegen. Daraus folgt $e_{nk} \lessdot e_{nl}$.
  Ist dagegen $m' = 2$, so ist $n, l \in J$ und $n > l$ nach der Konstruktion von A'. Da $k \in I$ ist, gilt $L'_k \ni e_{nk} \lessdot e_{nl} \in L'_l$.
  Ist $m' = 3$, so ist entweder $l \in J$ (woraus sofort $e_{nk} \lessdot e_{nl}$ folgt) oder $l, n \in I$ mit $n < l$. Daraus folgt wiederum $n > k$, also $k < l$. Da alle Ereignisse der Logs von I außer X bloß disjunkte Basistypen besitzen, folgt aus $B(e_{nk}) = B(E^1_{nk})$ mit $n, k \in I$, dass $B(e_{nk}) = B(E^1_{nk}) = X$ ist. Analog folgt $B(e_{nl}) = B(E^3_{nl}) = X = B(e_{nk})$, woraus mit $k, l \in I$ und $k < l$ folgt, dass $e_{nk} \lessdot e_{nl}$ ist.
  Ist dagegen $m' = 4$, so ist $n \in I$ und $l \in J$ nach der Konstruktion von A'. Da $k \in I$

ist, gilt $L'_k \ni e_{nk} \lessdot e_{nl} \in L'_l$.

Der nächste Fall ist $m = 2$, woraus folgt $n,k \in J$ und $n > k$. Für m' bleiben nach der Konstruktion von A' die Werte $\{1,2,3\}$ übrig, was die Konstruktion von L' auf $\{2,3\}$ reduziert, was zu $l \in J$ führt.
Aus $m' = 2$ folgt mit der Konstruktion von L' entweder $B(E^2_{nk}) < B(E^2_{nl})$ oder $B(E^2_{nk}) = B(E^2_{nl})$ mit $k < l$. In beiden Fällen ergibt sich $e_{nk} \lessdot e_{nl}$.
Aus $m' = 3$ folgt mit der Konstruktion von A', dass $n < l$ und somit $k < n < l$ gilt, und mit der Konstruktion von L' folgt $B(E^2_{nk}) \leq B(E^3_{nl})$ und somit $e_{nk} \lessdot e_{nl}$.

Ist $m = 3$, so muss m' nach der Konstruktion von L' mindestens 2 sein:
Ist $m' = 2$, so ist $n,l \in J$ nach der Konstruktion von A', woraus wiederum folgt $k \in J$. Aus der Konstruktion von L'_n folgt durch $E^3_{nk} \prec_n E^2_{nl}$, dass $B(e_{nk}) = B(E^3_{nk}) < B(E^2_{nl}) = B(e_{nl})$ gilt und somit $e_{nk} \lessdot e_{nl}$.
Ist $m' = 3$, so gilt entweder $B(E^3_{nk}) < B(E^3_{nl})$ oder $B(E^3_{nk}) = B(E^3_{nl})$ mit $k < l$. Außerdem ist entweder $n,k,l \in I$ oder $n,k,l \in J$, woraus folgt $e_{nk} \lessdot e_{nl}$.
Ist $m' = 4$, so ist $n \in I$ und $l \in J$ nach der Konstruktion von A'. Wäre $k \in J$ so wäre $n \in J$ ↯, also ist $k \in I$ und es folgt sofort $e_{nk} \lessdot e_{nl}$.

Ist $m = 4$ so muss $m' = 4$ sein, also gilt $k,l \in J$, $n \in I$ nach der Konstruktion von A'. Nach der Definition von L'_n sind alle 4er Sendeereignisse in einem Block, also gilt entweder $B(E^4_{nk}) < B(E^4_{nl})$ oder $B(E^4_{nk}) = B(E^4_{nl})$ mit $k < l$. Daraus folgt $e_{nk} \lessdot e_{nl}$.

- Sind $s_1, s_2 \in L_0$ gegeben, so gibt $r_1 \in R_k$ mit $s_1 = A'(r_1) = A(r_1)$ und $r_2 \in R_l$ mit $s_2 = A'(r_2) = A(r_2)$. Außerdem findet man folgende Möglichkeiten vor:

  $s_1 \prec_0 \blacktriangleleft$ und $s_2 \prec_0 \blacktriangle$, woraus nach der Konstruktion von L_0 folgt $k,l \in I$ und $P(s_1) < P(s_2)$. Daraus folgt mit $B(r_1) = P(r_1) = P(s_1)$ und $B(r_2) = P(r_2) = P(s_2)$ wiederum $r_1 \lessdot r_2$.

  Ist $\blacktriangleleft \prec_0 s_1 \prec_0 s_2 \prec_0 \blacktriangle$, so ist $k,l \in I$ und $X = P(s_1) = P(s_2) = P(r_1) = P(r_2) = B(r_1) = B(r_2)$. Daraus folgt $k < l$ und $r_1 \lessdot r_2$, da die Anordnung der Sendeereignis in L_0 für Ereignisse mit dem gleichen Typ als aufsteigend bezüglich der Nummerierung der Logs vorausgesetzt wurde.

Ist $\blacktriangle \prec_0 s_1 \prec_0 s_2$, so ist $k, l \in J$ und es gilt entweder $B(r_1) < B(r_2)$ oder $B(r_1) = B(r_2)$ mit $k < l$, abermals begründet durch die Konstruktion von L_0 und der Sortierung bezüglich der Lognummerierung. Daraus folgt sofort $r_1 \lessdot r_2$.

Ist $s_1 \prec_0 \blacktriangle \prec_0 s_2$, so ist $k \in I$ und $l \in J$, woraus sofort folgt $R_k \ni r_1 \lessdot r_2 \in R_l$.

Damit wurde gezeigt: Für alle $e_1, e_2 \in L'_n \cap (R' \cup A'(R'))$ gilt $e_1 \prec_n e_2 \Rightarrow A'^{-1}(e_1) \lessdot A'^{-1}(e_2)$.

Mit Lemma A.2 folgt daraus, dass A' wie gewünscht konsistent ist. $\square$

Anhang B.

Heuristik

B.1. Erweiterte Notation

Da die Arbeitsweise der Heuristik eng mit Intervallvektoruhren verknüpft ist, macht es Sinn, für alle Beweise eine Notation zu verwenden, die auf diesen Uhren basiert. Dazu werden alle Ereignisse über einer Indexmenge I durchnummeriert und jedem Ereignis e wird eine Intervallvektoruhr $V(e) \in I^{n_L} \times I^{n_L}$ zugewiesen. Statt $V(e)$ wird zudem im folgenden oft nur e geschrieben, sofern die Bedeutung aus dem Kontext klar ersichtlich ist.

Die Einträge einer Intervallvektoruhr $V(e)$ bzw. e haben die Form $e_{i,min}$ bzw. $e_{i,max}$.
Gilt $e_{i,min} = e_{i,max}$, so wird zur Klarheit einfach e_i anstatt $e_{i,min}$ verwendet.
Außerdem sei e_{min} der n_L-Vektor $(e_{1,min}, e_{2,min}, \ldots, e_{n_L,min})$, und e_{max} sei der n_L-Vektor $(e_{1,max}, e_{2,max}, \ldots, e_{n_L,max})$.
Es gelte $e < f$ falls $e_{i,m} < f_{i,m}$ für alle $1 < i < n_L$ und $m \in \{min, max\}$.
Es gelte $e \leq f$ falls $e_{i,m} \leq f_{i,m}$ für alle $1 < i < n_L$ und $m \in \{min, max\}$.
Es gelte $e = f$ falls $e_{i,m} = f_{i,m}$ für alle $1 < i < n_L$ und $m \in \{min, max\}$.
$>$ und $\geq$ gelt entsprechend, $e \neq f$ gelte, wenn nicht $e = f$ gilt.
Achtung: Aus $e \leq f, e \neq f$ folgt dann nicht $e < f$

Durch die Intervallvektoruhren wird jedem Empfangsereignis r eine Menge von möglichen Sendeereignissen

$$S(r) = \{s \in S | V(s)_{min} \leq V(r)_{max} \wedge V(s)_{max} \geq V(r)_{min} \wedge P(r) = P(s)\}$$

zugewiesen.

Das erste dieser Sendeereignisse heiße $S_f(r) \in S(r)$ und das letzte $S_l(r) \in S(r)$. Es gilt also für alle $s \in S(r)$, dass $S_f(r) \leq s \leq S_l(r)$ ist.

In manchen Beweisen werden mehrere unterschiedliche Belegungen V, V' oder V'' betrachtet, in diesen Fällen bezeichnen S, S' und S'' die Sendeereignisse, die einem Empfangsereignis durch diese neuen (Intervall-)Vektoruhren zugewiesen werden. Dazu muss in der obigen Definition einfach das V durch eben diese V' oder V'' ersetzt werden.
Die abkürzende Schreibweise $e = V(e)$ wird dabei wirklich nur für $V(e)$ und nicht etwa für $V'(e)$ verwendet. (sofern $V(e) \neq V'(e)$ gilt)

Außerdem gehen wir zur besseren Übersichtlichkeit in diesem Anhang B davon aus, dass $L_0 = \emptyset$ ist, so dass wir eine mit 1 beginnende Nummerierung der Logs verwenden und n_L die Zahl der Logs angibt.

B.2. Eigenschaften der Lösungen

In diesem Abschnitt werden Eigenschaften der auf Matterns Vektoruhren [Mat89] basierenden Intervallvektoruhren betrachtet, die auch unabhängig von der eigentlichen Heuristik gelten.

Dazu wird als nächstes der Begriff einer gültigen Lösung eingeführt, der im wesentlichem dem der konsistenten GA entspricht, allerdings nur auf den Werten der Vektoruhren basiert und weder eine GA noch eine GTO erfordert:

Definition B.1. *Eine Belegung der Vektoruhren $V : \bigcup_i L_i \to I^{n_L}$ zusammen mit einer bijektive Sortierfunktion $e : \{1, 2, \ldots, n_t\} \to L$ heißt gültige Lösung zu den Logs L_i, falls sie folgende Bedingungen erfüllt:*

1. $V(e_1) \leq V(e_2) \leq V(e_3) \leq \ldots \leq V(e_{n_{t-1}}) \leq V(e_{n_t})$

2. $V(e_i) \leq V(e_j), V(e_i) \neq V(e_j)$ *für alle* $e_i, e_j \in L_k$ *und* $i < j$

3. $|S(r)| = 1$ *für alle* $r \in R$*, wobei* $S(r) = \{s \in S | V(s) = V(r) \wedge P(s) = P(r)\}$ *wie üblich*

4. $|\{r \in R \cap L_m | s \in S(r)\}| \leq 1$ *für alle* $s \in S$*,* $1 \leq m \leq n_l$

Es ist nun einfach zu zeigen, dass dieser Begriff zu dem einer konsistenten GA äquivalent ist:

Lemma B.2. *Für jede gültige Lösung* V, e *existiert ein konsistenter GTO* $\prec$ *und GA* A*, die jedem Empfangsereignis dieselben Sendeereignisse zuweisen.*
Entsprechendes gilt für die Rückrichtung.

Beweis. Die Zuordnung der Sendeereignisse zu einem Empfangsereignis r, wird in den beiden Notationen jeweils entweder durch $S(r)$ oder durch $A(r)$ angegeben. Dabei gilt $S(r) = \{A(r)\}$ für alle $r \in R$.

"⇒": Sei (V, e) eine gültige Lösung.

Sei $A(l) = \begin{cases} l & \text{für } l \in S \\ s \in S(l) & \text{für } l \in R \quad \text{(eindeutig, da } |S(l)| = 1) \end{cases}$

Daraus folgt:

1. $\forall s \in S : A(s) = s$

2. $\forall f \in L : P(f) = \begin{cases} P(A(f)) & \text{für } f \in S \\ P(s) = P(A(f)) & \text{für } f \in R \text{ und } s \in S(f) \end{cases} = P(A(f))$

3. $\forall r \in R : N(r) \neq N(A(r))$, da $V(r)_i$ eindeutig ist für alle $r \in L_i$, so dass aus $N(r) = N(A(r))$ folgen würde $A(r) = r$, was ein Widerspruch zu $r \notin S$ wäre.

Damit ist A tatsächlich eine GA nach der Definition von [SK09].
Dann sei eine GTO folgendermaßen definiert:
Sei $e_i \prec e_j$ für zwei durch die Sortierfunktion bestimmte Ereignisse $e_i, e_j \in S$ genau dann, wenn gilt $i < j$.
Damit ist $\prec$ eine irreflexive Ordnungsrelation.
Ist dann $f_1, f_2 \in L_k$ mit $f_{1_k} < f_{2_k}$ (entspricht $f_1 \prec_k f_2$ nach der Notation von [SK09]), so ist $V(A(f_1)) = V(f_1) \leq V(f_2) = V(A(f_2))$ und somit gilt $A(f_1) \prec A(f_2)$, also ist $\prec$ konsistent.
Damit existiert ein konsistentes Paar $(\prec, A)$

"$\Leftarrow$" Sei $(\prec, A)$ ein konsistentes Paar eines GTO und GA.
Aus der partiellen Relation $\prec$ kann man durch topologischen Sortieren eine globale Relation berechnen. Fügt man nun noch die Empfangsereignisse hinzu, so dass sie direkt nach ihrem Sendeereignisse gelistet sind, erhält man eine globale Reihenfolge $e : \{1, .., n_T\} \to L$
Indem man dem ersten Ereignis eine beliebige Vektoruhr zuweist, deren Werte man für das nächste Sendeereignis sukzessiv vergrößert, erhält man leicht Vektoruhren $V : L \to I^{n_L}$ mit $V(e_i) \leq V(e_j)$ für $e_i \leq e_j$.
Man kann dabei auch erreichen, dass $V(e_i)_l = k$ für das k-te Ereignis e_i im Log L_l gilt, denn aus $e_i \prec_l e_j$ folgt $i < j$, da in der GTO dann gilt $A(e_i) \prec A(e_j)$. Dadurch kann man beim sukzessiven Erhöhen der Vektoruhren den Wert eines Eintrags in der l-ten Zeile jeweils nur bei Erreichen eines Ereignisses im l-ten Log inkrementieren.
Dadurch erreicht man auch automatisch $V(e_i) \leq V(e_j), V(e_i) \neq V(e_j)$ für $e_i, e_j \in L_k$ mit $i < j$ und ebenso $V(e_i) = V(A(e_i))$. Daraus folgt dann auch $S(e_i) = \{A(e_i)\}$.
Außerdem gilt für zwei $q, r \in R_k$ mit $q \leq r, q \neq r$, dass $A(q) \prec A(r)$ ist, also ist $S(q) \neq S(r)$ und es gilt $|\{r \in R_k | s \in S(r)\}| \leq 1$.
Also gibt es eine gültige Lösung, im oben definierten Sinne, mit derselben Zuordnung als sie durch das Paar $(\prec, A)$ definiert wurde. $\square$

Da man nun weiß, dass diese Definition sinnvoll ist, kann man mit ihr weiterarbeiten. Es existieren verschiedene Möglichkeiten eine Lösung, bei der nur einige Ereignisse die Bedingungen für eine gültige Lösung erfüllen, fortzusetzen, so dass man weitere Ereignisse hinzufügen kann, welche dann auch diese Bedingungen erfüllen.

Das nächste Lemma behandelt den Fall, dass man eine gültige Lösung für zumindest alle Empfangsereignisse und alle diesen zugeordneten Sendeereignissen besitzt (natürlich kann die teilweise Lösung unter diesen Bedingungen auch noch weitere Sendeereignisse umfassen).
Dann kann man zu dieser Lösung immer ein bisher noch nicht enthaltenes Sendeereignis hinzufügen[1]:

Lemma B.3. *Ist $X \subseteq L$ und existiert eine gültige Lösung (V', e) für $L'_i = L_i \cap (R \cup S(R) \cup X)$ und sei $s \in L_j \setminus L'_j$, so existiert eine gültige Lösung (V'', f) für $L''_i = \begin{cases} L'_j \cup \{s\} & i = j \\ L'_i & sonst \end{cases}$, so dass $V''(d) = V'(d)$ für $d \in L'$ ist.*

Beweis. O.B.d.A. habe $s \in L'_j$ bereits einen eindeutigen Index s_j (wenn nicht könnte man einfach die Indexmenge als $\mathbb{Q}$ betrachten und einen entsprechenden Bruch als Index wählen).
Für alle $d \in L'$ sei nun $V''(d) = V'(d)$.
Dann sind die Bedingungen 3 und 4 automatisch erfüllt, da $s \notin R \cup S(R)$.
Desweitern lassen sich folgende Fälle unterscheiden:

1. Es gibt kein $a \in L'_j$ mit $V'(a)_j < s_j$

   Dann kann man $V''(s) < V'(e)$ für alle $e \in L'$ wählen und $f_i = e_{i+1}, f_1 = s$ setzen, wodurch man offensichtlich alle Bedingungen erfüllt

2. Es gibt kein $b \in L'_j$ mit $V'(b)_j > s_j$

   Dann kann man $V''(s) > V'(e)$ für alle $e \in L'$ wählen und $f_i = e_i, f_{|L'|+1} = s$ setzen, wodurch man offensichtlich alle Bedingungen erfüllt

3. Es gibt $a, b \in L'_j$ mit $V'(a)_j < s_j < V'(b)_j$
   O.B.d.A. sei a maximal und b minimal, so dass es kein $c \in L'$ gibt mit $V'(a)_j < V'(c)_j < s_j$ oder $s_j < V'(c)_j < V'(b)_j$.
   Dann gibt es ein i mit $a = e_i$ und es gilt offenbar $e_{i+1} \leq b$. Sei $V''(s)_{k,m} = V''(a)_{k,m}$

[1] Dieser Beweis in der Notation der Vektoruhren entspricht ungefähr dem Induktionsschritt der GRO im Lemma A.2.

für alle $k \neq j$ und $V''(s)_j = s_j$.

Sei $f_k = \begin{cases} e_k & \text{für } k \leq i \\ s & k = i+1 \\ e_{k-1} & k > i+1 \end{cases}$

Damit erfüllt das neue, keinem Ereignis zugeordnete Sendeereignis die Bedingungen einer gültigen Lösung (da nur die Bedingungen bezüglich der Vektoruhr für es relevant sind). Außerdem wurde die Reihenfolge der Elemente in L' nicht verändert, so dass diese anderen Ereignisse ebenfalls weiterhin alle Bedingungen erfüllen.

□

Der nächste betrachtete Fall ist, dass eine gültige Lösung für alle Empfangsereignisse existiert. In diesem Fall kann man allerdings nicht gleich die Axiome für eine gültige Lösung verwenden, da weniger Aussagen über die Sendeereignisse getroffen sind. Stattdessen verlangt man, dass jedem Empfangsereignis ein Sendeereignis zugeordnet ist und Empfangsereignisse mit unterschiedlichen Vektoruhren nicht vom gleichen Sendeereignis stammen können.
Dies ergibt in mathematischer Formulierung:

Lemma B.4. *Seien Intervallvektoruhren V gegeben, so dass für $e \in L_i$ der Wert $V(e)_i$ eindeutig ist und der lokalen Reihenfolge entspricht.*
Existiert eine bijektiven Sortierfunktion $f : \{1,2,\ldots n_R\} \rightarrow R$ und Vektoruhren $V' : R \rightarrow I^{n_L}$ mit folgenden Eigenschaften:

$$V'(f_1) \leq V'(f_2) \leq \ldots \leq V'(f_{n_R})$$

$$f_{i_{min}} \leq V'(i) \leq f_{i_{max}}$$

$$|S'(f_i)| = 1$$

$$S'(f_i) \neq S'(f_j) \text{ für } V'(f_i) \neq V'(f_j)$$

So existiert eine gültige Lösung (V'', e) mit $V''(f_i) = V'(f_i)$ für alle f_i.

Beweis. Sei $V''(f_i) = V'(f_i)$, so dass man im folgenden V' statt V'' schreiben kann.
Sei $V'(s) = V'(r)$ für $\{s\} = S'(r)$. (Eindeutig da $S'(f_i) \neq S'(f_j)$ für $V'(f_i) \neq V'(f_j)$)
Da $s \in S'(r)$ ist, gilt damit offenbar $s_{min} \leq V'(s) \leq s_{max}$.

Damit existiert eine gültige Lösung zu $L'_i = L_i \cap (R \cup S(R))$, denn:

1. Es gibt eine Sortierfunktion $e : \{1,2,\dots,|R \cup S(R)|\} \to R \cup S(R)$ mit $V'(e_i) \leq V'(e_j)$ für $i \leq j$, da es eine solche Funktion für die Menge $\{V'(r)|r \in R\}$ gibt und nach der Definition von $V'(s)$ gilt: $\{V'(e)|e \in R \cup S(R)\} = \{V'(r)|r \in R\}$.

2. Für $e_i, e_j \in L_k$ mit $i < j$, gilt $e_{i_k} < e_{j_k}$, da der k-te Index eindeutig ist und $e_{min} \leq V'(e) \leq e_{max}$ gilt. Also ist $V'(e_i)_k < V'(e_j)_k$ und somit $V'(e_i) \neq V'(e_j)$.

3. $|S''(r)| = |S'(r)| = 1$, da $S''(r) = \{s\} = S'(r)$ ist, da $V'(s) = V'(r)$ für $s \in S'(r)$ gesetzt wurde und $|S'(r)| = 1$ für alle r vorausgesetzt wurde.

4. Sind $p, q \in L_k$ mit $q \neq p$, so ist $V'(p) \neq V'(q)$, also ist nach den Voraussetzungen $S'(p) \neq S'(q)$. Da $|S'(r)| = 1$ ist $S'(p) \cap S'(q) = \emptyset$, also gibt es kein s mit $s \in S(p)$ und $s \in S(q)$, somit gibt es maximal ein $r \in R_k$ mit $s \in S(r)$ für alle s. Also gilt $|\{r \in R \cap L_k | s \in S(r)\}| \leq 1$ für alle $s \in S$.

Daraus folgt das eine gültige Lösung für $R \cup S(R)$ mit $V''(f_i) = V'(f_i)$ existiert. Mit Lemma B.3 folgt dann induktiv, dass auch eine gültige Lösung für L mit $V''(f_i) = V'(f_i)$ existiert, da man beliebig oft ein $s \in L \setminus (R \cup S(R))$ zur existierenden Lösung hinzufügen kann. □

Nun definieren wir ein weiteres Kriterium für die Existenz einer gültigen Lösung, bei dem man die Logs in zwei Hälften aufteilt: Dabei ist intuitiv klar, dass wenn man die Ereignisse in den Logs in zwei Hälften aufteilen und dann für jede Hälfte eine gültige Lösung finden kann, daraus eine Zuordnung für sämtliche Ereignisse berechnet werden kann. (Eine ähnliche Aufteilung wurde von [Mat89] als Schnitt bezeichnet)

Lemma B.5. *Gibt es gültige Lösungen (V^1, F^1) und (V^2, F^2) für die Logs L_i^1 und L_i^2 mit $L^1 \cap L^2 = \emptyset$, so gibt es auch eine gültige Lösung (V, F) für $L_i = L_i^1 + L_i^2$ mit $V(e^1) =$*

$V^1(e^1)$ für alle $e^1 \in L^1$ und $V(e^2) = V^2(e^2)$ für alle $e^2 \in L^2$, falls gilt $V(e^1) \leq V(e^2)$ und $V(e^1)_i < V(e^2)_i$ für alle $e^1 \in L_i^1$ und $e^2 \in L_i^2$

Beweis. Sei $L = \bigcup_j L_i$, $L^1 = \bigcup_j L_j^1$ und $L^2 = \bigcup_j L_j^2$
Seien die Vektoruhren für alle $e \in L$ folgendermaßen gesetzt:

$$V(e) = \begin{cases} V^1(e) & \text{falls } e \in L^1 \\ V^2(e) & \text{falls } e \in L^2 \end{cases}$$

Außerdem kann man für die Sortierfunktion setzen:

$$F(i) = \begin{cases} F^1(i) & \text{falls } i \leq |L^1| \\ F^2(i - |L^1|) & \text{falls } i > |L^1| \end{cases}$$

Dies erfüllt, wie man leicht erkennen kann, alle an eine gültige Lösung gestellten Bedingungen.
Die ersten beiden sind erfüllt, weil die Sortierung der Ereignisse in L^1 und L^2 untereinander in L nicht geändert wurde und die Vektoruhren von allen $e^2 \in L^2$ größer als die von allen $e^1 \in L^1$ sind.
Die letzen beiden sind erfüllt, weil die Vektoruhren selbst nicht geändert wurden und somit auch die Zuordnungen innerhalb der Logs unverändert bleiben. Zudem kann es durch die unterschiedlichen Vektoruhren für beide Logs keine Zuordnungen zwischen Ereignissen aus den beiden unterschiedlichen Logs geben. □

B.3. Eingabe

Dieser Abschnitt beschreibt die initiale Belegung der Intervallvektoruhren vor Ausführung des Algorithmus:

Damit die Ereignisse eindeutig unterscheidbar sind, muss jedes Ereignis einen eindeutigen Index im jeweiligen Log besitzen, also:

$$e_{l,min} = f_{l,max} \text{ für alle } e \in L_l$$

$$e_l \neq f_l \text{ für alle } e, f \in L_l \text{ mit } e \neq f$$

Jede Art von Sendeereignis muss eindeutig einem Sendeknoten zugeordnet sein, also: Für jeden Typ t gibt es ein Log L_l, so dass

$$\{s \in S | P(s) = t\} \subseteq L_l$$

Außerdem werden die restlichen Werte der Vektoruhren folgendermaßen initialisiert (∞ muss dabei nicht unbedingt unendlich bedeuten, es kann auch ein so großer endlicher Wert sein, dass es keine größeren Vektoruhreinträge gibt):

$$e_{i,min} = -\infty \text{ für alle } e \in L_l \text{ und } i \neq l$$

$$e_{i,max} = \infty \text{ für alle } e \in L_l \text{ und } i \neq l$$

Es wird im folgenden davon ausgegangen, dass zu dieser Eingabe mindestens eine gültige Lösung existiert.

B.4. Pseudocode

Dieser Abschnitt gibt einen Pseudocode für die Arbeitsweise der Heuristik an, der auf den in den vorherigen Abschnitten eingeführten Intervallvektoruhrnotationen basiert:

repeat

    //Kreuztausch: Änderung der Empfänger
    while $\exists r \exists i : r_{i,min} < S_f(r)_{i,min} \vee r_{i,max} > S_l(r)_{i,max}$ **do**
        $r_{min} \leftarrow \max(r_{min}, S_f(r)_{min})$
        $r_{max} \leftarrow \min(r_{max}, S_l(r)_{max})$
    end while

    //Kreuztausch: Änderung der Sender

while $\exists r \exists i : S_f(r)_{i,max} > r_{i,max} \vee S_l(r)_{i,min} < r_{i,min}$ **do**
  $S_f(r)_{max} \leftarrow \min(S_f(r)_{max}, r_{max})$
  $S_l(r)_{min} \leftarrow \max(S_l(r)_{min}, r_{min})$
end while

//Wischen: Minimum
while $e, f \in L_l$ mit $e_l < f_l$ und $\exists i : e_{i,min} > f_{i,min}$ **do**
  $f_{i,min} \leftarrow e_{i,min}$
end while

//Wischen: Maximum
while $e, f \in L_l$ mit $e_l < f_l$ und $\exists i : e_{i,max} > f_{i,max}$ **do**
  $e_{i,max} \leftarrow f_{i,max}$
end while

until keine Änderung

Eine andere Beschreibung kann man in [Mar08] finden.

B.5. Rückgabe des Algorithmus

In diesem Abschnitt werden einige Eigenschaften der vom Algorithmus berechneten Intervallvektoruhren aufgezählt.

Die Ausgabe besteht grundsätzlich aus den Vektoruhren und den eigentlich relevanten möglichen Sendermengen. Letztere sind für ein Empfangsereignis $r \in R$ gegeben durch:

$$S(r) = \{s \in S | s_{max} \geq r_{min} \wedge s_{min} \leq r_{max} \wedge P(s) = P(r)\}$$

Es gibt dabei kein Empfangsereignis r mit einem Index i, so dass gilt $r_{i,min} \leq S_f(r)_{i,min}$, $r_{i,max} \geq S_l(r)_{i,max}$, $S_f(r)_{i,max} \geq r_{i,max}$ oder $S_l(r)_{i,min} \leq r_{i,min}$
Es gibt keine Ereignisse e, f in Log L_l mit $e_l < f_l$ und $e_{i,min} > f_{i,min} \vee e_{i,max} > f_{i,max}$

Dies folgt daraus, dass die Kreuztausche solange durchgeführt werden, wie die ersten Bedingungen für r nicht erfüllt sind und das Wischen solange bis die Bedingungen für e, f erfüllt sind. Somit terminiert der Algorithmus erst wenn diese Bedingungen erfüllt sind.

Positiv formuliert ergibt sich mit den entsprechenden Abkürzungen[2]:

Lemma B.6. *Für die Ausgabe des Algorithmus gilt:*

S:

$$r_{min} \leq s_{max} \wedge s_{min} \leq r_{max} \textit{ für alle } s \in S(r)$$

KT:

$$S_f(r) \leq r \leq S_l(r) \textit{ für jedes } r \in R$$

WI:

$$e \leq f \textit{ für alle } e, f \in L_l \textit{ mit } e_l \leq f_l$$

Das nächste Lemma zeigt, dass die minimalen und maximalen Grenzen eines Empfangsereignisses scharf und ausreichend sind. Jedes Sendeereignis im k-ten Log dessen eindeutiger k-ter Index zwischen Minimum und Maximum des Empfangsereignis liegt, ist also nach dem Algorithmus ein mögliches Sendeereignis und bei jedem möglichen Sendeereignis liegt der Index dazwischen (Also muss man nicht die Vektoruhr des Sendeereignis beachten, sondern bloß dessen Position im Log).
Formal schreibt sich das als:

Lemma B.7. *Für ein Ereignis $r \in R$ mit $s \in L_l$ für alle $s \in S$ mit $P(s) = P(r)$ lässt sich $S(r)$ schreiben als*

$$S(r) = \left\{ s \in L_l | r_{l,min} \leq s_l \leq r_{l,max} \wedge P(s) = P(r) \right\}$$

[2] Sie stehen für »Sender«, »Kreuztausch« oder »Wischen«, und werden in den folgenden Beweisen als Hochindizes bei Vergleichen benutzt, um diese zu erklären und zu begründen.

Beweis. Nach der Definition von $S(r)$ und der Einschränkung der versendeten Pakettypen ist

$$S(r) = \{s \in L_l | s_{max} \geq r_{min} \wedge s_{min} \leq r_{max} \wedge P(s) = P(r)\}$$

"$\subseteq$": Zu zeigen ist: $s_{i,max} \geq r_{i,min} \wedge s_{i,min} \leq r_{i,max} \forall i \Rightarrow r_{l,min} \leq s_l \leq r_{l,max}$.

Da $s_l = s_{l,min} = s_{l,max}$ ist, gilt $s_l \geq r_{l,min} \wedge s_l \leq r_{l,max}$, also $r_{l,min} \leq s_l \leq r_{l,max}$.

"$\supseteq$": Zu zeigen ist: $r_{l,min} \leq s_l \leq r_{l,max} \Rightarrow s_{i,max} \geq r_{i,min} \wedge s_{i,min} \leq r_{i,max} \forall i$.

Angenommen es gibt ein s mit $r_{l,min} \leq s_l \leq r_{l,max}$, aber $s_{i,max} < r_{i,min}$ für ein bestimmtes $i \neq l$.
Offenbar ist dieses $s \notin S(r)$ und es gilt $S_f(r)_{i,max} \geq r_{i,min} > s_{i,max}$, also $S_f(r) \geq^W s$, da sich $S_f(r)$ und s im selben Log befinden. Da $s \neq S_f(r) \in S(r)$ sein muss, folgt $S_f(r)_l > s_l$, da $s_l \neq t_l$ für alle $s,t \in L_l$ gilt.
Daraus folgt: $r_{l,min} \leq s_l < S_f(r)_l \leq^{KT} r_{l,min}$ ↯
Analog folgt, dass $s_{i,min} > r_{i,max}$ unmöglich ist. □

Das nächste Lemma zeigt, dass zwei Empfangsereignisse im selben Log, deren Sendeereignisse ebenfalls im gleichen Log liegen, unterschiedliche frühest- und spätestmögliche Sendeereignisse besitzen:

Lemma B.8. *Für zwei Empfangsereignisse $q,r \in L_l$ mit einem festen m mit $S(q) \subseteq L_m$ und $S(r) \subseteq L_m$, gilt $q_l < r_l \Rightarrow q_{m,min} < r_{m,min} \wedge q_{m,max} < r_{m,max}$.*

Beweis. Sei $q_l < r_l$

Fall 1: *min*-Einträge:
Angenommen es gilt $q_{m,min} \geq r_{m,min}$, so ist $S_f(q) \geq S_f(r)$, da $S_f(q)_m = q_{m,min} \geq r_{m,min} = S_f(r)_m$ ist, also ist $S_f(r)_{l,max} \leq S_f(q)_{l,max} \leq^{KT} q_l < r_l$. Daraus folgt $S_f(r)$ erfüllt nicht $S_f(r)_{i,max} \geq r_{i,min}$ für alle i, also ist $S_f(r) \notin S(r)$ ↯

Fall 2: *max*-Einträge:
Angenommen es gilt $q_{m,max} \geq r_{m,max}$, so ist $S_l(q) \geq S_l(r)$, also ist $S_l(q)_{l,min} \geq$

$S_l(r)_{l,min} \geq^{KT} r_l > q_l$
Daraus folgt $S_l(q)$ erfüllt nicht $S_l(q)_{i,min} \leq q_{i,max}$ für alle i, also ist $S_l(q) \notin S(q)$ ↯

□

Lemma B.9. *Für zwei Empfangsereignisse $q, r \in L_l$ mit $q_l \neq r_l$ gilt $S_f(q) \neq S_f(r)$ und $S_l(q) \neq S_l(r)$.*

Beweis. Ist $P(q) \neq P(r)$ ist es trivial.
Gilt $P(q) = P(r)$, so gibt es ein L_m mit $S(q) \subseteq L_m$ und $S(r) \subseteq L_m$. Somit gilt nach dem vorherigen Lemma B.8, dass gilt $q_{m,min} \neq r_{m,min}$ und $q_{m,max} \neq r_{m,max}$.
Daraus folgt $S_f(q)_m = q_{m,min} \neq r_{m,min} = S_f(r)_m$ und $S_l(q)_m = q_{m,max} \neq r_{m,max} = S_l(r)_m$.
Also gilt die Behauptung dieses Lemmas. □

Eine interessante Eigenschaft der Vektoruhren ist es, dass es möglich ist, den Minimumanteil der Intervallvektoruhr eines einzelnen Sendeereignisses ohne vollständige Ausführung des Algorithmus zu bestimmen. Es gilt nämlich:

Lemma B.10. *Für ein Sendeereignis $s \in S_i$ ist*

$$s_{j,min} = \begin{cases} s_{i,max} \text{ (=eindeutiger Index)} & \text{für } i = j \\ r_j & \text{falls es ein } r \in L_j \cap R \text{ mit } S_l(r) = s \text{ gibt} \\ max(\{e_{j,min}\} \cup \{r_{j,min} | s \in S_l(r)\}) & \text{sonst, mit } e \in L_i \text{ das größte } e \text{ mit } e_i < s_i \end{cases}$$

Beweis. Betrachte die einzelnen Fälle:

Fall 1 i = j
Dies ist trivial und folgt aus der Eingabe

Fall 2 Es gibt ein $r \in L_j$ mit $S_l(r) = s$
Dann ist $s_{j,min} \leq^S r_{j,max}$ und $r_{j,min} \leq^{KT} s_{j,min}$ und aus $r_{j,min} = r_{j,max}$ folgt:
$s_{j,min} \leq r_{j,max} = r_{j,min} \leq s_{j,min}$
$\Rightarrow s_{j,min} = r_j$

Fall 3 Offensichtlich wird der Wert $s_{j,min}$ vom Algorithmus nur auf zwei Weisen geändert. Entweder beim Wischen, oder beim Kreuztauschen.
Beim Wischen wird nun $s_{j,min} = \max(s_{j,min}, e_{j,min})$ gesetzt, mit $e \in L_i$ das größte e mit $e_i < s_i$
Beim Kreuztauschen wird entsprechend $s_{j,min} = \max(s_{j,min}, r_{j,min})$ gesetzt, für ein r mit $s = S(r)$.
Also ist $s_{j,min} \geq max(\{e_{j,min}\} \cup \{r_{j,min} | s \in S_l(r)\})$
Auch $\leq$ gilt, wovon man sich überzeugen kann, indem man beobachtet, dass jeder Wert der $s_{j,min}$ zugewiesen werden könnte, kleiner als die obige Maximumsbildung ist.
Dazu wird mit e^T der Wert der Vektoruhren des Ereignis des e zum Ausführungszeitpunkt T des Algorithmus notiert. (beispielsweise gezählt als Schleifendurchlauf). Zu Beginn ist $s^0_{j,min}$ offensichtlich 0.
Jede *min*-Grenze wird nur vergrößert, also ist $e^T_{j,min} \leq e^{T'}_{j,min} \leq e_{j,min}$ für $T \leq T'$
Wird s_{min} im Verlauf durch ein Empfangsereignis r, für das auch im Endergebnis $S_l(r) = s$ gilt, gesetzt, so gilt offenbar $r^T_{min} \leq r^{T'}_{min}$ für alle $T \leq T'$. Also ist in dem Fall die Betrachtung des neusten r ausreichend.
Gab es dagegen einen Zeitpunkt T mit $S^T_l(q) = s$ mit einem $q \in R$ mit $S_l(q) \neq s$, so gilt abermals $q^T_{min} \leq q^{T'}_{min}$ für jeden späteren Zeitpunkt $T' \geq T$. Direkt nach einem Kreuztausch für $S_l(q) \subseteq L_i$ gilt zu allen Zeiten $q^T_{i,max} \leq S^T_l(q)_{i,max} = S^T_l(q)_{i,min} \leq q^T_{i,max}$. Da die *max*-Grenze nur verkleinert wird, gilt offenbar $S^T_l(q)_i = q^T_{i,max} \geq q_{i,max} = S_l(q)_i$.
Also folgt aus $s = S^T_l(q) \in L_i$, dass $s_i \geq S_l(q)_i$ gilt, also $s_{min} \geq S_l(q)_{min} \geq q_{min} \geq q^T_{min}$, wobei das erste $\geq$ allein durch Wischen gilt!
Der Anteil des Wischen wurde aber bereits zu $s_{min} \leq e_{min}$ gezeigt, also ist das oben angegebene Maximum tatsächlich der Wert von s_{min}.

□

Im Abschnitt B.2 wurde allgemein gezeigt, dass man zwei gültige Lösungen zu einer kombinieren kann, wenn die Einträge disjunkt sind, alle Ereignisse im zweiten Log eine spätere Vektoruhr als die im ersten besitzen, und die Vektoruhren eine Reihenfolge im eindeutigen Index erhalten.

Im Spezialfall eines vom Algorithmus verarbeiteten Log kann man dies vereinfachen, indem man verlangt, dass in jeder partiellen gültigen Lösung nur Vektoruhren vergeben werden, die auch vom Algorithmus als möglich erkannt wurden. Dann ist es nämlich nicht mehr nötig, dass jede Vektoruhr im zweiten Log später ist als die im ersten, sondern es reicht, wenn dies für die Vektoruhren der Empfangsereignisse gilt.

Dazu benutzt man einfach das Lemma, dass eine gültige Lösung zu $R \cup S(R)$ zu einer allgemeinen Lösungen mit diesen Vektoruhren fortgesetzt werden kann, und die Tatsache, dass jedes Sendeereignis die gleiche Vektoruhr wie das zugeordnete Empfangsereignis besitzt.

Lemma B.11. *Gibt es gültige Lösungen (V^1, F^1) und (V^2, F^2) für die Logs L_i^1 und L_i^2 mit $L^1 \cap L^2 = \emptyset$, so gibt es auch eine gültige Lösung (V, F) für $L_i = L_i^1 + L_i^2$ mit $V(r^1) = V^1(r^1)$ für alle $r^1 \in L^1 \cap R$ und $V(r^2) = V^2(r^2)$ für alle $r^2 \in L^2 \cap R$, falls gilt $V(r^1) \leq V(r^2)$ für alle $r^1 \in L^1 \cap R$ und $r^2 \in L^2 \cap R$ sowie $r_{min} \leq V(r) \leq r_{max}$ für alle $r \in L \cap R$ und $s_{min} \leq V(s) \leq s_{max}$ für alle $s \in L \cap S(R)$*

Beweis. Sei $L_i' = L_i \cap (R \cup S(R))$, $L_i'^1 = L_i' \cap L^1$ und $L_i'^2 = L_i' \cap L^2$.
Aus $V(r^1) \leq V(r^2)$ für alle $r^1 \in L_i^1 \cap R$ und $r^2 \in L_i^2 \cap R$, folgt $V(s^1) \leq V(s^2)$ für alle $s^1 \in L_i^1 \cap S(R)$ und $s^2 \in L_i^2 \cap S(R)$, da bei einer gültigen Lösung die Sende- und zugeordneten Empfangsereignisse dieselbe Vektoruhr haben und beide vorhanden sein müssen.
Also ist $V(e^1) \leq V(e^2)$ für alle $e^1 \in L'^1$ und $e^2 \in L'^2$.
Außerdem wurde im Lemma verlangt, dass $e_{min} \leq V(e) \leq e_{max}$ für alle $e \in L_i'$ gilt, woraus folgt $V(e)_i = e_i$.
Da dieser Index eindeutig ist, ist $V(e)_i \neq V(f)_i$ für $e, f \in L_i$ mit $e \neq f$.
Daraus folgt mit $V(e^1) \leq V(e^2)$, dass gilt $V(e^1)_i < V(e^2)_i$ für $e^1 \in L_i'^1$ und $e^2 \in L_i'^2$.
Also existiert nach Lemma B.5 eine gültige Lösung für L_i', welche die Werte von $V'(e)$ für $e \in L'$ erhält.
Mit der induktiven Anwendung von Lemma B.3 folgt aus der Existenz einer gültigen Lösung für $L_i' = L_i \cap (R \cup S(R))$, dass es eine gültige Lösung für L_i gibt, die ebenfalls alle Vektoruhren beibehält. □

Nun muss man diese Erkenntnisse benutzen, um damit zu untersuchen, ob jede vom Algorithmus gefundene Lösung, tatsächlich eine gültige Lösung ist.

Die Grundidee ist, dass man eine beliebige Zuordnung von Empfangsereignis und Sendeereignis wählt, die nach dem Algorithmus möglich, und aus den restlichen, nicht zugeordneten Ereignissen eine gültige Lösung konstruiert.

B.6. Spezialfall: $n_R = 1$

In diesem Abschnitt wird der Spezialfall behandelt, dass es nur einen Empfänger gibt und sich in allen anderen Logs bloß Sendeereignisse befinden.
Dies ist besonders einfach, da dann die bekannte lokale Reihenfolge automatisch eine globale Reihenfolge aller Empfangsereignisse darstellt, und oben gezeigt wurde, dass eine solche Reihenfolge, mit einigen leicht zu zeigenden Bedingungen, ausreicht, um eine gültige Lösung zu induzieren.

Lemma B.12. *Ist $L'_i \subseteq L_i$ und $L' = \bigcup_i L'_i$ eine Untermenge der Ereignisse mit $L'_i \cap R = \emptyset$ für alle $i \neq l$ für ein festes l und gilt für jedes $r \in L'_l \cap R$, dass $S_f(r) \in L'$ ist, so gibt es eine gültige Lösung (V', e) für L'_i mit $V'(r) = r_{min}$ für alle $r \in L' \cap R$.*

Beweis. Sei $V'(r) = r_{min}$ für alle $r \in R' = L' \cap R$.
Offenbar existiert eine Sortierfunktion $e : \{1, 2, \ldots, |R'|\} \to R'$ mit $V'(e_i) \leq V'(e_j)$ für $i < j$, da nach dem Wischen innerhalb eines Logs die Minima sortiert sind und alle r in L'_l liegen.
Offensichtlich gilt auch $q_{min} \leq V'(q) \leq q_{max}$ für alle $q \in R$.
Oben wurde gezeigt, dass für $S_f(q)$ gilt $S_f(q)_{min} \leq^{KT} q_{min} \leq^{S} S_f(q)_{max}$, also ist $|S(q)| \geq 1$ und nach der Eindeutigkeit mindestens einer Koordinate in jedem Log gilt somit $|S(q)| = 1$.
Für zwei Empfangsereignisse p, q mit $p \neq q$ im selben Log mit Sendeereignissen im gleichen Log (vom gleichen Typ) gilt nach Lemma B.8 $p_{min} \neq q_{min}$ also gilt $S'(p) \neq S'(q)$.
Somit folgt aus $S'(q) = S'(p)$, dass $p = q$ also $V'(p) = V'(q)$ gilt.

Somit existiert nach Lemma B.4 eine gültige Lösung mit diesen Werten von $V'(r)$.

□

Auf Grund der Symmetrie kann man dieses Lemma auch umkehren, so dass sich analog zeigen lässt, dass wenn alle $S_l(r) \in L'$ sind, eine gültige Lösung mit $V'(r) = r_{max}$ existiert.

Damit folgt dann, dass jedes Sendeereignis im Intervall (vom gleichen Typ) zu einer gültigen Lösung führt:

Lemma B.13. *In diesem Spezialfall gibt es für jedes Empfangsereignis r zu jeder Wahl der Vektoruhr* $V'(r)$ *mit* $r_{min} \leq V'(r) \leq r_{max}$ *und* $|S(r)| = 1$ *eine gültige Lösung.*

Beweis. Sei $V'(r)$ fest.
Offenbar ist $V'(r)$ mit $e : 1 \mapsto r$ eine gültige Lösung für $B = \{r\} \cup S(r)$.
Sei $A = \{q \in R | q \leq r, q \neq r\} \cup \{S_f(q) | q \in R, q \leq r, q \neq r\}$.
Dann gibt es nach dem vorherigen Lemma B.12 eine gültige Lösung für A, so dass $V'(q) = q_{min}$ für alle $q \in A$ ist.
Sei $C = \{q \in R | q \geq r, q \neq r\} \cup \{S_l(q) | q \in R, q \geq r, q \neq r\}$.
Dann gibt es nach der Umkehrung des vorherigen Lemmas B.12 eine gültige Lösung für C, so dass $V'(q) = q_{max}$ für alle $q \in C$ ist.
Offenbar ist $q_{min} \leq V'(r)$ für alle $q \in A$ und $q_{max} \geq V'(r)$ für alle $q \in A$, da nach dem Wischen alle Empfangsereignisse in L_l sortiert sind.
Daraus folgt $V'(a) \leq V'(r) \leq V'(c)$ für alle $a \in A, c \in C$ und somit gibt es eine gültige Lösung mit diesen Vektoruhren für $A \cup B \cup C$ nach Lemma B.11.
Da $R \subseteq A \cup B \cup C$ ist, und somit $S(R) \subseteq A \cup B \cup C$ ist, lässt sich nach Lemma B.3 jedes weitere Sendeereignis iterativ zu dieser gültigen Lösung hinzufügen, so dass es eine gültige Lösung mit diesen Vektoruhren für alle Ereignisse gibt.

□

Also führt in diesem Spezialfall jeder Sender innerhalb des vom Algorithmus bestimmten Intervalles zu einer gültigen Lösung und er wird im Gegensatz zum allgemeinen Fall optimal gelöst.

B.7. Spezialfall: $n_S = 1$

In diesem Spezialfall arbeitet die Heuristik exakt, wovon man sich anhand folgender Beobachtung leicht überzeugen kann:

Sämtliche Kreuztäusche erfolgen jeweils nur zwischen dem eindeutigen Sendelog und einem weiteren Empfangslog, woraus letztendlich folgt, dass die Intervalle der Vektoruhr eines Empfangsereignisses immer größer sind, als die Intervalle in der Vektoruhr eines Sendeereignisses, es sei denn bei dem Intervall handelt es sich um den eindeutigen Index. Formal: Ist $r \in R_i$, so gilt für $s \in S(r)$ anfänglich $r_{k,min} \leq s_{k,min} \leq s_{k,max} \leq r_{k,max}$ für alle $k \neq i$.
Geht man alle möglichen Änderungen der Intervallvektoruhr bei einem Kreuztausch durch, sieht man, dass diese Eigenschaft invariant unter Kreuztäuschen ist.

Wischen in einem Empfangslog kann diese Eigenschaft zwar kurzzeitig aufheben, sobald das Wischen aber ebenfalls im Sendelog durchgeführt wurde, ist sie aber wiederhergestellt. Die Eigenschaft wird nämlich beim Wischen für das Minimum genau dann zerstört, wenn es $p, q \in R_i$ mit $p \leq q$ und zwei $s \in S(p)$, $t \in S(q)$ gibt, für die gilt $V(p)_{k,min} > V(t)_{k,min}$. Nach dem Wischen im Empfängerlog gilt dann $V'(q)_{k,min} \geq^W V(p)_{k,min} > V(t)_{k,min}$. Nach dem Wischen im Senderlog gilt jedoch auch $V'(t)_{k,min} \geq^W V'(s)_{k,min} \geq V(p)_{k,min}$ wodurch die Invariante wieder gilt (analog für das Maximum).

Das bedeutet, dass Zuordnungen nur auf Grund der Reihenfolge der Ereignisse im Sendelog ausgeschlossen werden und keine Zuordnungen in einem Empfangslog von der Reihenfolge der Ereignisse in einem anderen Empfangslog abhängig sind. Also erhält man durch das gleichzeitige Ausführen des Algorithmus auf allen Logs exakt dieselbe Zuordnungen, wie wenn man den Algorithmus einzeln für jeweils jedes Empfangslog und das Sendelog durchführt. Dass aber die gefundene Lösung bei einem Sendelog und einem Empfangslog gültig ist, wurde schon im vorherigen Abschnitt B.6 bewiesen.

B.8. Spezialfall: $n_l = 2$

In diesem Abschnitt wird der Spezialfall behandelt, dass exakt zwei Logs existieren.

Um zu die Eigenschaften einer gültigen Ausgabe des Algorithmus besser zu verstehen, kann man überlegen, wie man vorgehen muss, um aus dieser Ausgabe nun eine einzige gültige Lösung zu berechnen. Angenommen also der Algorithmus hätte eine Menge von möglichen Intervallvektoruhren berechnet, wie findet man daraus eine globale Reihenfolge für alle Empfangsereignisse?

Ein naheliegendes Vorgehen ist es, einfach für ein Ereignis eine Vektoruhr innerhalb der berechneten Intervalle zu wählen, und dann die Menge der Ereignisse zu betrachten, die später aufgetreten sein könnten. Dann muss man nun für jedes dieser Ereignisse eine passende Vektoruhr finden.

Es ist leicht eine neue Vektoruhr zu finden, die zu einer globalen Sortierung passt, da man immer das Maximum von der letzten Vektoruhr und einer anderen, innerhalb der Intervallvektoruhr des neuen Ereignis liegenden wählen kann. Diese kommt dann immer direkt nach der letzten Vektoruhr in der Sortierung. Das Problem dabei ist nun, weder ein Ereignis noch eine Vektoruhr zu wählen, die später zu Widersprüchen führen könnten.

Die naive Idee, die Minimumvektoruhr für die Maximumbildung zu wählen, funktioniert nicht, da dann möglicherweise kein passendes Sendeereignis gefunden werden kann. Nimmt dagegen man das Maximum von dieser Vektoruhr mit dem Minimum eines Sendeereignis, entsteht das Problem, dass eventuell ein Sendeereignis doppelt zugeordnet wird.

Glücklicherweise lässt sich zeigen, dass man immer entweder das frühestmögliche oder das spätestmögliche Sendeereignis wählen kann, ohne Widersprüche zu erhalten, also so, dass man später wieder ein Sendeereigniss wählen kann.

Dies zeigt das nächsten Lemma, dessen Variablen die folgende intuitive Bedeutung haben: v ist die zuletzt berechnete, globale Vektoruhr. Es werden nur die Ereignisse betrachtet, die nach dem durch diese Vektoruhr festgelegtem Zeitpunkt geschehen sein können (welche sich im $n_L = 2$-Fall eindeutig bestimmen lassen). Man kann diese Vektoruhr auch als Zuordnung eines Sendeereignis s zu einem Empfangsereignis r betrachten, welche bei zwei Logs ebenfalls eindeutig ist. s muss dabei entweder das nach den Intervallvektoruhren spätestmögliche oder das frühestmögliche Sendeereignis für r sein. K_i sind die Logs, beschränkt auf die Ereignisse, denen noch keine Vektoruhr zugewiesen worden ist.

Lemma B.14. *Sei $r \in R_\rho$ ein Empfangsereignis und $s \in L_\varsigma \cap S(r)$ ein Sendeereignis mit $s = S_f(r)$ oder $s = S_l(r)$, sei v eine Vektoruhr mit $v_\rho = r_\rho$ und $v_\varsigma = s_\varsigma$.*
Sei $K_i = \{q \in R_i | q_i > v_i\} \neq \emptyset$.
Dann gibt es ein l, ein $r' \in K_l$ mit $r'_{max} \geq v$ und $r' \leq q$ für alle $q \in K_l$ und ein $s' \in \{S_f(r), S_l(r)\}$ mit $s' \neq s$, $s'_{max} \geq v$ und $s' \leq q$ für alle $q \in K_m$ für $m \neq l$.

Beweis. O.B.d.A. sei $r \in L_1$ und $s \in L_2$. Dann lässt sich v als (r_1, s_2) schreiben.

Nach dem Wischen sind die Ereignisse in den L_i wohlgeordnet, es existieren also n_L feste $e_i \in K_i \subseteq L_i$, so dass $e_i \leq e$ für alle $e \in K_i$ für $i \leq n_L$ gilt.
Für e_1 gilt offensichtlich $e_{1_1} \geq^W r_1 = v_1$ und $e_{1_{2,max}} \geq^W r_{2,max} \geq^S S_l(r)_{2,min} \geq^W s_{2,min} = s_2 = v_2$ also ist $e_{1_{max}} \geq v$.
Ebenso ist $e_{2_2} \geq^W s_2 = v_2$ und $e_{2_{1,max}} \geq^W s_{1,max} \geq^S r_{1,min} = r_1 = v_1$, also ist $e_{2_{max}} \geq v$.

Fall 1: $s = S_f(r)$

Da $e_1, r \in L_1$ sind und $e_{1_1} > v_1 = r_1$ gilt, ist nach der lokalen Ordnung $e_{1_{2,min}} \geq^W r_{2,min}$. Also ist $S_f(e_1)_2 = e_{1_{2,min}} \geq r_{2,min} = S_f(r)_2 = s_2 = v_2$.
Da auch $S_f(e_1)_{1,max} \geq^S e_{1_1} > v_1$ gilt, ist $S_f(e_1)_{max} \geq v$.

Offensichtlich gilt $L_1 \ni S_l(e_2) \neq s \in L_2$.
Außerdem wurde in Lemma B.9 gezeigt, dass $s = S_f(r) \neq S_f(e_1)$ aus $r \neq e_1$ folgt.

Der einfache Fall $S_f(e_1) \leq e_2$ wird weiter unten behandelt, angenommen also es gilt $S_f(e_1) \geq e_2$ (da $S_f(e_1), e_2 \in L_2$ sind, gibt es tatsächlich nur diese beiden Fälle).
So folgt $e_2 \leq S_f(e_1) \leq^{KT} e_1$ und somit $S_l(e_2)_1 = e_{2_{1,max}} \leq e_{1_1}$.
Also gilt $S_l(e_2) \leq^W e_1$.

Fall 2: $s = S_l(r)$

Da $e_2, s \in L_2$ sind und $e_{2_2} > v_2 = s_2$ gilt, ist $e_{2_{1,min}} \geq^W s_{1,min} = S_l(r)_{1,min} \geq^{KT} r_{1,min} = r_1$. Daraus folgt $S_f(e_2)_1 = e_{2_{1,min}} \geq r_1 = v_1$.
Da auch gilt $S_f(e_2)_{2,max} \geq^S e_{2_2} > v_2$ ist $S_f(e_2)_{max} \geq v$.

Offensichtlich gilt $L_1 \ni S_f(e_2) \neq s \in L_2$.
Außerdem wurde in Lemma B.9 gezeigt, dass $S_l(e_1) \neq S_l(r) = s$ aus $r \neq e_1$ folgt.

Der einfache Fall $S_f(e_2) \leq e_1$ wird weiter unten behandelt, angenommen also es gilt $S_f(e_2) \geq e_1$ (wiederum sind diese beiden Möglichkeiten ausreichend).
So folgt $e_1 \leq S_f(e_2) \leq^{KT} e_2$ und somit $S_l(e_1)_2 = e_{1_{2,max}} \leq e_{2_2}$.
Also gilt $S_l(e_1) \leq^W e_2$.

In beiden Fällen gibt es also ein l, so dass für alle $m \neq l$ entweder $S_l(e_l) \leq e_m$ oder $S_f(e_l) \leq e_m$ mit $S_f(e_l)_{max} \geq v$ gilt.
Offensichtlich ist $S_l(e_i)_{max} \geq e_{i_{max}} \geq v$, man kann also das Ergebnis der beiden Fälle schreiben als: Es gibt ein l und ein $s' \in \{S_f(e_l), S_l(e_l)\}$ mit $s'_{max} \geq v$ und $s' \leq e_m$ für alle $m \neq l$
Die Untersuchung der beiden Fälle hat außerdem gezeigt, dass $s \neq s'$ gilt.
Also gilt die Aussage dieses Lemmas. □

Nice to have und auch interessant, aber für die weitere Argumentation völlig nutzlos:

Lemma B.15. *Sei v eine Vektoruhr und $K_i = \{r \in R_i | r_i > v_i\} \neq \emptyset$ und gibt es ein l, ein $r \in K_l \cap R$ mit $r_{max} \geq v$ und $r \leq e$ für alle $e \in K_l$ und ein $s \in S(r)$ mit $s_{max} \geq v$ und $s \leq e$ für alle $e \in K_m$ für $m \neq l$:*

Dann gibt es eine neue Vektoruhr $v' = (v'_i)$, und $K'_i = \{r \in R_i | r_i > v'_i\}$, so dass folgende Eigenschaften erfüllt sind:

1. $v' \geq v$
2. $r_{min} \leq v' \leq r_{max}$
3. $s_{min} \leq v' \leq s_{max}$

4. $v_l = r_l$

5. $v_m = s_m$ *falls* $s \in L_m$

6. $K'_i = K_i$ *falls* $i \neq l$

7. $K'_l = K_l \setminus \{r\}$

8. $v' \leq e_{max}$ *für alle* $e \in K'_i$

Beweis. Nach dem Wischen sind die Ereignisse in den L_i wohlgeordnet, es existieren also n_L feste $e_i \in K_i \subseteq L_i$, so dass $e_i \leq e$ für alle $e \in K_i$ für $i \leq n_L$ gilt.

Sei $v'_i = \max\{v_i, r_{i,min}, s_{i,min}\}$ für alle i.

Dies erfüllt folgende Bedingungen:

$$v'_i \geq v_i \forall i \Rightarrow v' \geq v$$

$$v'_i \geq r_{i,min} \forall i \Rightarrow r_{min} \leq v'$$

$$v \leq r_{max} \wedge r_{min} \leq r_{max} \wedge s_{min} \leq^S r_{max} \Rightarrow v' \leq r_{max}$$

$$v'_i \geq s_{i,min} \forall i \Rightarrow s_{min} \leq v'$$

$$v \leq s_{max} \wedge r_{i,min} \leq^S s_{i,max} \wedge s_{i,min} \leq s_{i,max} \forall i \Rightarrow v' \leq s_{max}$$

Also gilt $r_{min} \leq v' \leq r_{max}$ und somit $v'_l = r_l$, da $r_{l,min} = r_{l,max}$ gilt.
Für $s \in L_m$ gilt entsprechend $s_{min} \leq v' \leq s_{max}$ und somit $v'_m = s_m$, da $s_{m,min} = s_{m,max}$ gilt.

Sei $K'_i = \{r \in R_i | r_i > v'_i\}$. Da $v' \geq v$ gilt, ist $K'_i \subseteq K_i$

Nach den Anforderungen von s gilt für $m \neq l$, dass $s \in L_m$ und $s \leq e_m \in K_m$ ist.
Da e_m kein Sendeereignis ist, muss $s_m < e_{m_m}$ gelten und da $e_m \leq q$ für alle $q \in K_m$ gilt, folgt $q_m \geq e_{m_m} > s_m = v_m$, also ist $q \in K'_m$ und $K'_m \supseteq K_m$.
Also ist $K'_m = K_m$ für $m \neq l$.
Da $v_l = r_l$ ist, ist $r \notin K'_l$. Für jedes andere $q \in K_l$ wurde $q \geq r$ verlangt und somit folgt

aus der Eindeutigkeit des lokalen Index $q_l > r_l = v_l$, also $q \in K'_l$ und $K'_l \supseteq K_l \setminus \{r\}$.
Also ist $K'_l = K_l \setminus \{r\}$.

Oben wurde gezeigt, dass $v' \leq r_{max}$ und $v' \leq s_{max}$ gilt.
Für alle $q \in K_l$ gilt $q \geq r$ und folglich $q_{max} \geq r_{max} \geq v'$.
Für alle $q \in K_m$ mit $m \neq l$ gilt entsprechend $q_{max} \geq s_{max} \geq v'$.

Damit erfüllt die obere Wahl von v' und K' die im Lemma geforderten Bedingungen. □

Das nächste Lemma stellt eine hinreichende Bedingung für die Existenz einer gültigen Lösung für eine Teilmenge aller Logs auf.

Diese Bedingung ist dabei in zwei Teile geteilt, der erste Teil, ist der Spezialfall, dass von den beiden Logs nur ein Log tatsächlich Empfangsereignisse enthält, und es sich bei dem $n_L = 2$-Fall in Wirklichkeit um einen $n_R = 1$-Fall handelt.

Der zweite Teil ist der allgemeinere Fall, dass in beiden Logs Empfangsereignisse existieren. Dann gibt es in jedem Log ein frühestes Empfangsereignis mit einem zugeordneten frühesten und spätesten Sendeereignis. Die Bedingung für die Existenz einer gültigen Lösung lautet nun einfach, dass bei einem diesen beiden Empfangsereignis eines der beiden erwähnte Sendeereignissen vor jedem Empfangsereignis im anderen Log passiert sein muss.

Lemma B.16. *Sei p ein Empfangsereignis, $t \in S(r)$ ein ihm zugeordnetes Sendeereignis und v eine Vektoruhr mit $p_{min} \leq v \leq p_{max}$ und $t_{min} \leq v \leq t_{max}$.*
Erfüllen $H_i = \{e \in L_i | e_i > v_i\}$ und $K_i = H_i \cap R$ eine der folgenden Eigenschaften:

1. *Es gibt ein l, so dass $K_m = \emptyset$ für $l \neq m$ ist.*

2. *Es gibt ein l und $r \in K_l$ mit $r_{max} \geq v$ und $r \leq e$ für alle $e \in K_l$, so dass für ein $s \in \{S_f(r), S_l(r)\} \subseteq L_m$ gilt $s \geq v$ und $s \leq e$ für alle $e \in K_m$.*

So existiert eine gültige Lösung (V', f) für diese H_i mit $V'(r) \geq v$ für alle $r \in K_i$.

Beweis. Sei $H = \bigcup_i H_i$ und $K = \bigcup_i K_i$

Fall 1: Die erste Bedingung ist erfüllt:

O.B.d.A. sei $p \in L_1$ und $t \in L_2$.

Für $q \in K_1$ gilt $q \geq^W p$ und mit Lemma B.8 folgt $q_{2,max} > p_{2,max} \geq^S S_l(p)_{2,min} = t_2 = v_2$, also ist $S_l(q) \in H_2$.
Für $q \in K_2$ ist entsprechend $S_l(q)_1 =^{KT} q_{1,max} \geq^W t_{1,max} \geq^S p_{1,min} = p_1$, woraus $S_l(q)_1 > p_1 = v_1$ folgt, da p kein Sendeereignis ist, also ist $S_l(q) \in H_1$.
Für jedes $q \in K$ ist also $S_l(q) \in H$.

Erfüllt H die erste oben genannte Bedingung, so folgt mit dem Lemma B.12 aus dem Fall $n_R = 1$ sofort, dass es eine gültige Lösung (V', e) für H gibt, mit $V'(q) = q_{max}$ für alle $q \in K$.

Fall 2: Die zweite Bedingung ist erfüllt:
Der Beweis erfolgt mittels Induktion über die Zahl der Empfangsereignisse in allen K_i:

Induktionsanfang: $|K| = 1$
Offensichtlich ist $K_m = \emptyset$ für $m \neq l$, da es nur ein einiges Empfangsereignis $r \in L_l$ gibt.
Damit entspricht der Induktionsanfang der ersten Bedingung und ist durch sie bereits gezeigt.

Induktionsschritt: $|K| = n \rightarrow |K| = n+1$
Man kann O.B.d.A. davon ausgehen, dass $K_i \neq \emptyset$ für alle i ist (sonst ist alles bereits gezeigt), O.B.d.A. $r \in L_1$.
Sei $v' = (r_1, s_2)$, $H_i' = \{e \in L_i | e_i > v_i'\}$ und $K_i' = H_i' \cap R$.
Da $r \leq q$ für alle $q \in K_1$ und $s \leq q$ für alle $q \in K_2$ ist, gibt es offensichtlich nur ein Empfangsereignis in $H_i \setminus H_i'$, und somit erhält man eine gültige Lösung für $H_i \setminus H_i'$, wenn man die Vektoruhr von r und s auf v' setzt.
Man sieht, dass offensichtlich $|K'| < |K|$ ist, es scheint also logisch nun die Induktionsannahme anzuwenden.
Ist ein $K_i' = \emptyset$, so ist offensichtlich die erste Bedingung dieses Lemmas erfüllt, ansonsten sieht man: Da $s \in \{S_f(r), S_l(r)\}$ ist und $v' = (r_1, s_2)$ gilt, folgt mit dem Lemma B.14, dass es ein l', ein $r' \in K_{l'}'$ mit $r_{max}' \geq v'$ und $r' \leq q$ für alle $q \in K_{l'}'$ und $s' \in \{S_f(r'), S_l(r')\}$ mit $s' \neq s$, $s_{max}' \geq v'$ und $s' \leq q$ für alle $q \in K_m'$ mit $m \neq l'$

gibt.
Damit erfüllen die r, s, r', s', l' die zweite Bedingung dieses Lemmas und somit gibt es nach der Induktionsannahme eine gültige Lösung für K' mit $V'(q) \geq v'$ für alle $q \in K'$.
v' ist offensichtlich die Vektoruhr die in einer gültigen Lösung für r und s diesen beiden Ereignissen zugeordnet ist, also gilt $V'(r) \leq V'(q)$ für alle $q \in K'$ und es gibt nach Lemma B.11 eine gültige Lösung für $H' \cup (H \setminus H') = H$. Außerdem ist $V'(q) \geq v$, da $V'(q) \geq V'(r) \geq v$ für alle $q \in K$.

$\square$

Dieses Lemma ist auch in seiner umgekehrten Form gültig, wenn man jeweils die Bedeutung von früher und später vertauscht. Der Beweis erfolgt analog.

Dieses Lemma (mit seinem analogen Zwilling) wird nun benutzt, um eine hinreichende Bedingung für eine mögliche Zuordnung aufzustellen.
Offensichtlich teilt im Fall $n_L = 2$ jede Zuordnung die Logs in eine spätere und eine frühere Hälfte, also lautet diese Bedingung nun einfach, dass beide Hälften nach der oben definierten, hinreichenden Bedingung eine gültige Lösung besitzen müssen:

Lemma B.17. *Sei r ein Emfangsereignis, $s \in S(r)$ ein ihm zugeordnetes Sendeereignis und v eine Vektoruhr mit $r_{min} \leq v \leq r_{max}$ und $s_{min} \leq v \leq s_{max}$.*
Sei $A_i = \{e \in L_i | e_i < v_i\}$ und $B_i = \{e \in L_i | e_i > v_i\}$.
Erfüllen die B_i eine der folgenden Eigenschaften:

1. *Es gibt ein l, so dass $B_m \cap R = \emptyset$ für $l \neq m$ ist.*

2. *Es gibt ein l und $q \in B_l \cap R$ mit $q_{max} \geq v$ und $q \leq e$ für alle $e \in B_l$, so dass für ein $t \in \{S_f(q), S_l(q)\} \subseteq L_m$ gilt $t \geq v$ und $t \leq e$ für alle $e \in B_m$.*

Und erfüllen die A_i gleichfalls eine der folgenden Eigenschaften:

1. *Es gibt ein l, so dass $A_m \cap R = \emptyset$ für $l \neq m$ ist.*

2. *Es gibt ein l und $r \in A_l \cap R$ mit $r_{min} \leq v$ und $r \geq e$ für alle $e \in A_l$, so dass für ein $s \in \{S_f(r), S_l(r)\} \subseteq L_m$ gilt $s \leq v$ und $s \geq e$ für alle $e \in A_m$.*

So existiert eine gültige Lösung, die r und s einander zuordnet, also beiden die Vektoruhr v zuweist.

Beweis. O.B.d.A. sei $r \in L_1$ und $s \in L_2$.

Sei $A = \bigcup_i A_i$ und $B = \bigcup_i B_i$.

Offensichtlich ist $L_i = \{e \in L_i | e_i < v_i\} + \{e \in L_i | e_i = v_i\} + \{e \in L_i | e_i > v_i\}$.
Man kann die Logs also aufteilen in:

$$L_1 = A_1 + \{r\} + B_1$$

$$L_2 = A_2 + \{s\} + B_2$$

Dann gibt es eine gültige Lösung für $\{r\} \cup \{s\}$ mit $V'(r) = V'(s) = v$.

Nach dem zuvor gezeigten Lemma B.16 gibt es eine gültige Lösung für B_i mit $V'(q) \geq v$ für alle $q \in B_i \cap R_i$ und auf Grund der Symmetrie von S_f und S_l muss es auch eine gültige Lösung für A_i mit $V'(q) \leq v$ für alle $q \in A_i \cap R$ geben.

Somit existiert nach Lemma B.11 eine gemeinsame gültige Lösung (V', e) für L_i mit $V'(r) = v$. □

Es lohnt sich einen Spezialfall des obigen Lemmas zu betrachten, nämlich, dass die anfängliche Zuordnung entweder zum frühesten oder spätesten vom Algorithmus als möglich erklärten Sendeereignis erfolgt. Dann sind die beiden Bedingungen nämlich automatisch erfüllt:

Lemma B.18. *Wählt man eine Vektoruhr $V'(r)$ mit $r_{min} \leq V'(r) \leq r_{max}$ so, dass gilt $S'(r) = \{S_f(r)\}$ oder $S'(r) = \{S_l(r)\}$, so gibt es eine gültige Lösung (V', f) mit dieser Wahl von $V'(r)$.*

Beweis. O.B.d.A. sei $r \in L_1$ und $s \in L_2$.

Offensichtlich folgt aus $r_{min} \leq V'(r) \leq r_{max}$ und $s_{min} \leq V'(r) \leq s_{max}$, dass gilt $V'(r) = (r_1, s_2)$.

Sei $B_i = \{r \in R_i | r_i > v_i\} \neq \emptyset$.
Gibt es ein $B_i = \emptyset$ ist die erste Bedingungen im Lemma B.17 erfüllt.
Ist $B_i \neq \emptyset$ für alle i, existiert nach dem Lemma B.14 ein l, ein $r' \in K_l$ mit $r'_{max} \geq v$ und $r' \leq q$ für alle $q \in K_l$ und ein $s' \in \{S_f(r), S_l(r)\}$ mit $s' \neq s$, $s'_{max} \geq v$ und $s' \leq q$ für alle $q \in K_m$ für $m \neq l$, was die zweite Bedingung erfüllt.
Also erfüllt B_i immer mindestens eine der beiden Bedingungen.

Analoges lässt sich auf Grund der Symmetrie für A_i zeigen.

Also existiert nach dem vorherigen Lemma B.17 eine gültige Lösung mit diesem Wert von $V'(r)$.

□

Damit hat man nun gezeigt, dass jede vom Algorithmus als möglich deklarierte minimale oder maximale Vektoruhr eines Empfangsereignisses tatsächlich im Spezialfall $n_L = 2$ zu einer gültigen Lösung führt.

Anhang C.

Notation

Hier werden die wichtigsten in dieser Arbeit verwendete Symbole in alphabetischer Reihenfolge aufgezählt:

$\blacktriangleleft, \blacktriangle, \blacktriangleright$		Trennsymbole bei der Reduktion von X3C auf uSRC
$\prec_l$	$\prec_l \subseteq L_l \times L_l$	Eine für jedes Log lokale Ordnungsrelation
$\preceq_l$	$\prec_l \vee =$	Reflexive Version von $\prec_l$
$\prec$	$\prec \subseteq S \times S$	GTO, Globale Ordnungsrelation über alle Sendeereignisse
$\square, \boxdot$		Symbolisieren irrelevante Ereignisse
A	$A : L \to S$	GA, Für jedes Ereignis ein zugeordnetes Sendeereignis
B	$B : L \to \mathscr{P}$	Basistyp eines Ereignisses bei der X3C->SRC Reduktion
$A, B, \ldots, X, Y$		Sendeereignis; die Buchstabenart bestimmt den Typ
$a, b, \ldots, x, y$		Empfangsereignisse; die Buchstabenart bestimmt den Typ
α, β, γ	$\alpha, \beta, \gamma \in M_i$	Alle Elemente in der X3C-Menge M_i
C	$C : M \to \mathbb{N}$	Zahl der ein bestimmtes Element enthaltenden X3C-Mengen
D	$D : L \to \{\tilde{s}, \tilde{r}\}$	Unterscheidet Sende- und Empfangsereignisse
E_{kl}^m	$E_{kl}^m \in L_k$	Spezielles Sendeereignis bei der X3C->SRC-Reduktion
e_{kl}	$E_{kl} \in L_l$	Spezielles Empfangsereignis bei der X3C->SRC-Reduktion
e, f		Beliebige Ereignisse
e, f	$e, f : \mathbb{N} \to L$	Bei der Heuristik eine Sortierfunktion
i, j, k, l, m		Indizes
I		Bei der Heuristik Indexmenge der Ereignisse
I	$I \subset \mathbb{N}$	Nummern der X3C-Mengen einer Überdeckung

J	$J \subset \mathbb{N}$	Nummern der Mengen nicht in einer Überdeckung
L	$L = \bigcup_i L_i$	Sämtliche Ereignisse
L_0		Spezielles Log mit Sendeereignissen (nur Komplexität)
H_l, K_l, L_l		Log l, eine Menge von Ereignissen
M	$M = \{1, \ldots, m_T\}$	Die Elemente der bei X3C zu überdeckenden Menge
M_i		Eine der bei X3C zur Auswahl stehenden Mengen
m_T		Anzahl der Elemente bei X3C
N	$N : L \to \mathbb{N}$	Für jedes Ereignis die Nummer des enthaltenden Logs
n_L		Zahl der Logs (falls $L_0 \neq \emptyset$ ist, gibt es $n_L + 1$ Logs)
P	$P : L \to \mathscr{P}$	Der Pakettyp eines Ereignisses
p, q, r		Empfangsereignisse
$\mathscr{P}$		Die Menge aller Pakettypen
r, s, t		Sendeereignisse
R		Die Menge aller Empfangsereignisse
S		Die Menge aller Sendeereignisse
$S(r)$		Alle für r gefundenen Sendeereignisse (nur Heuristik)
$S_f(r)$	$S_f(r) \in S(r)$	Frühestes Ereignis in $S(r)$
$S_l(r)$	$S_l(r) \in S(r)$	Spätestes Ereignis in $S(r)$
T_i	$\{T_i\} \subseteq \{M_i\}$	Eine zur Lösung von X3C gehörende Menge
$V(e)$	$V : L \to I^{n_L} \times I^{n_L}$	(Intervall-)vektoruhr für e (nur Heuristik)
$V(e)_{min}$		Minimumspalte einer Intervallvektoruhr
$V(e)_{max}$		Maximumspalte einer Intervallvektoruhr
$V(e)_{i,m}$		Eintrag einer Intervallvektoruhr
X	$X > m_T$	Ein zusätzlicher Typ bei der Reduktion von X3C
x		Ein Ereignis mit Typ X bei X3C

Diese Liste ist nicht vollständig, es fehlen zum Beispiel alle gestrichenen Variablen wie i', welche dieselbe Grundbedeutung wie die ungestrichenen haben. In manchen Beweisen, vor allem bei denen bezüglich der Heuristik, werden zudem kurzzeitig neue Bedeutungen eingeführt, diese sind aber jeweils aus dem Kontext ersichtlich.

Literaturverzeichnis

[Che08] CHESHIRE, S.: *IPv4 Address Conflict Detection*. RFC 5227 (Proposed Standard). http://www.ietf.org/rfc/rfc5227.txt. Version: Juli 2008 (Request for Comments).

[Coo71] COOK, Stephen A.: The complexity of theorem-proving procedures. In: *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, Shaker Heights, Ohio, United States, 1971, S. 151–158.

[GJ79] GAREY, Michael R.; JOHNSON, David S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979

[Lam78] LAMPORT, Leslie: Time, clocks, and the ordering of events in a distributed system. In: *Communication ACM* 21 (1978), July, Nr. 7, S. 558– –565.

[Las61] LASSER, Daniel J.: Topological ordering of a list of randomly-numbered elements of a network. In: *Commun. ACM* 4 (1961), Nr. 4, S. 167–168. ISSN 0001–0782.

[Mar08] MAROLD, Alexander: *Rekonstruktion der globalen Ereignisreihenfolge aus lokalen Logdateien*, Heinrich-Heine-Universität Düsseldorf, Bachelorarbeit, 2008

[Mat89] MATTERN, Friedemann: Virtual time and global states of distributed systems. In: *Parallel and Distributed Algorithms* (1989).

[SK09] SCHEUERMANN, Björn; KIESS, Wolfgang: Who Said That? - The Send-Receive Correlation Problem in Network Log Analysis. In: *ACM Performance Evaluation Review* (2009).

[SKR+09] SCHEUERMANN, Björn; KIESS, Wolfgang; ROOS, Magnus; JARRE, Florian; MAUVE, Martin: On the time synchronization of distributed log files in networks with local broadcast media. In: *IEEE/ACM Trans. Netw.* 17 (2009), Nr. 2, S. 431–444. ISSN 1063–6692.

[Wan09] WANKE, Egon: *Ohne Titel*. 2009. (Eine Reduktion von X3C auf das allgemeine Zuordnungsproblem)

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Düsseldorf, 24.August 2009 Benito van der Zander