

Diskrete Mathematik Übungsstunde

Zusammenfassung

Leon Kolmanić

04.12.2023

1 Besprechung Bonusaufgabe

Häufige Fehler waren:

- $a^{|G|}$ berechnet
- Vergessen zu erwähnen, dass Polynom bei a) Grad 2 hat
- Faktoren/Rechnungen vergessen

Die Aufgabe ist insgesamt sehr gut ausgefallen, fast alle Abgaben sind genau dem Schema der Musterlösung gefolgt. Es gab nur häufig kleinere Fehler: Zum Einen wurde sehr oft $a^{|G|}$ berechnet, also im Kontext der b) $(x+3)^{24}$. Diese Rechnung ist aber redundant, weil in jeder endlichen Gruppe $a^{|G|} = e$ gilt, also ein Element zur Potenz der Gruppenordnung ergibt das neutrale Element, in unserem Fall 1. Wenn wir zeigen wollen, dass ein Element ein Generator ist, zeigen wir, dass die Ordnung des Elements nicht strikt kleiner sein kann als die Gruppenordnung. Es ist dann klar, dass die Ordnung des Elements die Gruppenordnung selbst ist, weil die Ordnung eines Elements immer die Gruppenordnung teilt. Es ist natürlich nicht falsch, diese Rechnung durchzuführen. Ich wollte hier nur darauf hinweisen, dass man sie auslassen kann, um in der Prüfung Zeit zu sparen.

Es ist wichtig zu beachten, dass ein Polynom ohne Nullstellen reduzibel sein kann. Ein Beispiel ist $x^4 + x^2 + 1$ über \mathbb{Z}_2 . Dieses Polynom hat keine Nullstellen, aber man kann es faktorisieren mit $(x^2 + x + 1)^2$. Deshalb sollte man bei Polynomen vom Grad 2 oder 3 explizit erwähnen, dass es reicht zu prüfen, ob diese Nullstellen haben, weil sie Grad 2/3 haben. Als Begründung kann man Korollar 5.30 verwenden.

Es ist auch manchmal vorgekommen, dass Faktoren der Gruppenordnung vergessen wurden. Somit wurden dann auch bei der b) einige Potenzen nicht berechnet. Um das zu vermeiden kann ich nur empfehlen, am Ende noch mal ganz genau zu schauen, ob man alle positiven Faktoren der Gruppenordnung gefunden hat. Das sind sonst verschenkte Punkte.

2 Logische Kalküle

2.1 Einführung in logische Kalküle

Logische Kalküle sind endliche Mengen von Schlussregeln. Eine Schlussregel gibt eine Formel an, die man aus einer Menge von Formeln, den Preconditions, herleiten kann. Beweise mittels logischen Kalkülen sind die formalste Art von Beweis, die wir in dieser Vorlesung betrachten. Im Gegensatz zu anderen Beweisen kann man in einem logischen Kalkül nur die Schlussregeln verwenden und das gesamte Format des Beweises ist fest vorgegeben.

In einer typischen Aufgabe zu logischen Kalkülen ist ein logisches Kalkül und eine Menge von Formeln gegeben. Weiter wird eine Zielformel bestimmt, die man aus der Menge von Formeln herleiten soll. Der häufigste Fehler bei diesen Aufgaben ist, dass Lemmas/Theoreme/... aus dem Skript verwendet werden. Man könnte ja zum Beispiel denken, dass wenn man die Zielformel $\neg\neg A$ herleiten soll, alternativ die Zielformel A herleiten kann, weil diese nach Lemma 6.1 äquivalent sind. Das ist aber in so einem Beweis nicht erlaubt, weil Lemma 6.1 keine Schlussregel in dem logischen Kalkül ist und man in dem Beweis nur diese verwenden darf. Ein weiterer häufiger Fehler ist die Annahme, dass die Schlussregeln in einem Kalkül immer sound sind. Betrachte zum Beispiel das Kalkül $K = \{R\}$, mit einer einzigen Schlussregel R :

$$\emptyset \vdash_R F$$

Die Schlussregel R hat keine Preconditions, das heißt wir können diese immer anwenden. Sie enthält ausserdem einen Platzhalter F . Da die Preconditions immer erfüllt sind, dürfen wir für F beliebige Formeln einsetzen. Wir können also zum Beispiel aus der Formelmengung $\{A\}$ die Formel $\neg A$ herleiten, indem wir die Schlussregel R einmal verwenden und $F = \neg A$ setzen. Es ist also korrekt $\{A\} \vdash_K \neg A$ zu schreiben, weil es in dem Kalkül K eine Herleitung von $\neg A$ aus der Menge $\{A\}$ gibt. Natürlich widerspricht dies unserer Intuition, die uns sagt, dass $\neg A$ natürlich keine logische Konsequenz von A ist. Aber das ist genau der Punkt: Der Begriff der logischen Konsequenz \models kann zu dem Begriff der Herleitbarkeit \vdash_K in einem Missverhältnis stehen. Für \vdash_K ist einzig und allein entscheidend, ob die Schlussregeln im Kalkül K eine Herleitung erlauben oder nicht. Kalküle, in denen wir aus einer Menge M nur logische Konsequenzen von M ableiten können, nennen wir sound. Ausserdem nennen wir Kalküle komplett, wenn man jede logische Konsequenz von M auch tatsächlich in dem Kalkül herleiten kann. Auch diese Eigenschaft muss nicht immer erfüllt sein. Betrachte zum Beispiel das Kalkül \emptyset . Dieses Kalkül enthält keine Herleitungsregeln. Natürlich gilt $A \wedge B \models A$, das kann man mittels der Wahrheitstabellen überprüfen. Aber

$$\{A \wedge B\} \not\vdash_{\emptyset} A$$

weil man mangels Herleitungsregeln A in dem Kalkül \emptyset nicht aus der Menge $\{A \wedge B\}$ herleiten kann.

Hier sind noch mal die wichtigsten Erkenntnisse zusammengefasst:

- In einem Beweis mit einem logischen Kalkül darf man ausschliesslich auf die Schlussregeln dieses Kalküls zurückgreifen, es ist insbesondere **nicht** erlaubt, Lemmas/Theoreme/... aus dem Skript zu verwenden oder den Beweis durch Argumentationen in Worten zu stützen
- Beweise mittels Kalkülen folgen einem fest vorgegebenen Format
- In einem Kalkül kann man unter Umständen “unsinnige Konsequenzen” beweisen, davon soll man sich nicht beirren lassen. Alles, was durch die Schlussregeln erlaubt ist, darf man herleiten.

2.2 Beweis mit einem logischen Kalkül

Im Folgendem betrachten wir an einem Beispiel das erwartete Format bei Kalkülbeweisen und wie die Anwendung einer Schlussregel genau abläuft. Zunächst ist das Kalkül K_1 mit den folgenden Schlussregeln gegeben:

$$\begin{array}{l} \emptyset \vdash_{R_1} F \rightarrow F \\ \{F\} \vdash_{R_2} G \rightarrow F \\ \{F \rightarrow G, F \rightarrow (H \rightarrow \neg G)\} \vdash_{R_3} F \rightarrow \neg H \end{array}$$

Wir sollen in diesem Kalkül die Formel $A \rightarrow \neg\neg A$ aus \emptyset herleiten. Es ist häufig vorteilhaft bei diesen Aufgaben rückwärts zu arbeiten, also sich zuerst zu überlegen, welche Regel man im letzten Schritt anwenden wird. Wir können im letzten Schritt nicht die Regel R_1 anwenden, weil wir mit R_1 nur Implikationen herleiten können, bei denen die linke Seite der Implikation gleich der rechten ist. Aber bei $A \rightarrow \neg\neg A$ ist nicht die linke Seite gleich der rechten. Hier muss man vorsichtig sein: Mit “gleich” meine ich hier $=$, also die Formeln sind exakt gleich geschrieben. Ich meine **nicht** \equiv , also dass die Formeln die gleichen Wahrheitstabellen haben. Wenn wir hingegen R_2 im letzten Schritt verwenden wollen, müssen wir im letzten Schritt $F = \neg\neg A$ und $G = A$ setzen. Wenn wir F und G so setzen, bedeutet das, dass die Menge der Preconditions $\{F\} = \{\neg\neg A\}$ ist. Wir müssen also um R_2 so anzuwenden zunächst $\neg\neg A$ mit dem Kalkül herleiten. Aber jede der Schlussregeln im Kalkül erlaubt uns nur Implikationen herzuleiten. Wir können also die Preconditions nicht erfüllen, also können wir auch R_2 nicht im letzten Schritt verwenden. Nun ist klar, dass wir im letzten Schritt R_3 verwenden müssen. Damit $F \rightarrow \neg H = A \rightarrow \neg\neg A$ gilt, müssen wir $F = A$ und $H = \neg A$ wählen. Hier gilt auch wieder: $F \rightarrow \neg H \equiv A \rightarrow \neg\neg A$ ist **nicht** ausreichend. Wir müssen in einem Kalkül die geforderte Formel syntaktisch exakt herleiten. Bei R_3 gibt es aber noch einen dritten Platzhalter G . Je nachdem wie wir diesen wählen, ändert sich die Menge der Preconditions. Wir wählen $G = A$ und probieren, ob die Herleitung klappt. Falls nicht, müssen wir uns für G etwas anderes einfallen lassen. Das passiert bei solchen Aufgaben häufig: Man muss manchmal die Wahl der Platzhalter solange raten, bis man die Herleitung schafft. Wir erhalten für die Menge der Preconditions

$$\{F \rightarrow G, F \rightarrow (H \rightarrow \neg G)\} = \{A \rightarrow A, A \rightarrow (\neg A \rightarrow \neg A)\}$$

Gelingt es uns, alle Formeln in dieser Menge im Kalkül herzuleiten, können wir den finalen Schritt mit R_3 durchführen und sind fertig. Versuchen wir zunächst $A \rightarrow A$ herzuleiten. Dafür können wir R_1 verwenden. Wir wählen $F = A$, denn dann $F \rightarrow F = A \rightarrow A$. Die Menge der Preconditions ist leer, also können wir diesen Schritt durchführen. Formal notieren wir das so:

$$\emptyset \vdash_{R_3} A \rightarrow A \tag{1}$$

Jeder Beweisschritt gibt an, was unsere Preconditions sind, welche Regel wir verwenden, welche Formel wir erhalten und welche Nummer wir dieser Formel geben, damit wir sie später wiederverwenden können. Die Belegung der Platzhalter geben wir nicht an, weil diese aus den Preconditions und der erhaltenen Formel klar ist. Nun müssen wir noch $A \rightarrow (\neg A \rightarrow \neg A)$ herleiten. Diese Formel können wir nur mit Regel R_2 herleiten, weil die linke und rechte Seite der Implikation verschieden sind und die rechte Seite der Implikation $\neg A \rightarrow \neg A$ nicht negiert ist. Damit

$$G \rightarrow F = A \rightarrow (\neg A \rightarrow \neg A)$$

gilt, müssen wir $G = A$ und $F = \neg A \rightarrow \neg A$ wählen. Um R_2 so anzuwenden brauchen wir die Precondition

$$\{F\} = \{\neg A \rightarrow \neg A\}$$

Diese können wir mit R_1 herleiten. Jetzt haben wir uns den gesamten Beweis rückwärts überlegt. Nun müssen wir von vorne anfangen und Schritt für Schritt die Herleitungen durchführen. Wir müssen aufpassen, dass wir eine Herleitung nur dann machen, wenn wir alle nötigen Preconditions gesammelt haben. Formal schreiben wir den Beweis so auf:

Beweis.

$$\emptyset \vdash_{R_1} A \rightarrow A \tag{1}$$

$$\emptyset \vdash_{R_1} \neg A \rightarrow \neg A \tag{2}$$

$$\{(2)\} \vdash_{R_2} A \rightarrow (\neg A \rightarrow \neg A) \tag{3}$$

$$\{(1), (3)\} \vdash_{R_3} A \rightarrow \neg\neg A$$

□

Noch einmal kurz zusammengefasst, wie ein Herleitungsschritt abläuft: Wir wählen für die Platzhalter in dem Herleitungsschritt beliebige Formeln. Wir können für die Platzhalter auch Formeln wählen, die wir noch nicht hergeleitet haben. Wir haben im Beweis vorhin zum Beispiel $F = A$ im letzten Schritt gewählt, obwohl wir A selbst nie hergeleitet haben. Wichtig ist nur, dass die Menge der Preconditions nach Einsetzen der Platzhalter nur Formeln enthält, die man schon hergeleitet hat. Falls alles passt, notieren wir den Herleitungsschritt formal wie oben beschrieben.

3 Beweissysteme

3.1 Einführung in die Beweissysteme

Beweissysteme sind ein sehr abstraktes Konzept. Sie sind definiert als ein Tupel $(\mathcal{S}, \mathcal{P}, \tau, \phi)$, das eine Menge von Aussagen \mathcal{S} , eine Menge von Beweisen \mathcal{P} , eine Wahrheitsfunktion τ und eine Beweisverifikationsfunktion ϕ enthält. Obwohl wir von Aussagen und Beweisen sprechen, können die Elemente der Aussagenmenge/Beweismenge beliebige mathematische Objekte sein. In einem Beweissystem kann zum Beispiel \mathbb{N} eine Aussage sein oder 4 ein Beweis sein. Beweissysteme müssen ausserdem nicht sinnvoll sein. Man könnte zum Beispiel ein Beweissystem definieren, in dem die Aussage "3 ist keine Primzahl" wahr ist, weil das so durch die Wahrheitsfunktion definiert wird. Ob eine Aussage in einem Beweissystem wahr ist oder ob ein Beweis ein gültiger Beweis für eine Aussage ist, wird einzig und allein durch die Wahrheitsfunktion und Beweisverifikationsfunktion bestimmt. Es gibt zwei Eigenschaften, die wir bei einem Beweissystem betrachten. Wir nennen ein Beweissystem sound, falls es keine falsche Aussage gibt, die einen gültigen Beweis hat. Ausserdem nennen wir ein Beweissystem komplett, falls es für jede wahre Aussage einen Beweis gibt. Was bedeutet dies ausgedrückt durch die Wahrheitsfunktion τ und die Beweisverifikationsfunktion ϕ ? Eine Aussage s ist falsch, wenn $\tau(s) = 0$. Wir sagen, dass p ein gültiger Beweis für s ist, falls $\phi(s, p) = 1$. Also bedeutet die Eigenschaft sound, dass es kein $s \in S$ gibt, sodass $\tau(s) = 0$ und es ein p gibt mit $\phi(s, p) = 1$. Analog bedeutet die Eigenschaft komplett, dass es für jedes $s \in S$ mit $\tau(s) = 1$ ein p gibt mit $\phi(s, p) = 1$.

Weil Beweissysteme ein so abstraktes Konzept sind, darf man in einem Beweis von Eigenschaften eines Beweissystems nur die Definitionen von τ und ϕ und die beiden Begriffe sound und komplett verwenden. Lemmas oder Theoreme gibt es im Skript für diese nicht. Argumente, die sich nicht auf diese Eigenschaften stützen, kommen in einem korrekten Beweis nicht vor.

3.2 Beweis durch Gegenbeispiel

Wenn wir eine Aussage über Beweissysteme widerlegen sollen, machen wir das genauso wie bei allen anderen Aussagentypen. Wir geben ein Gegenbeispiel an. Wir müssen also eine Menge von Aussagen \mathcal{S} , eine Menge von Beweisen \mathcal{P} , eine Wahrheitsfunktion τ und eine Beweisverifikationsfunktion ϕ für alle Beweissysteme angeben, die in unserem Gegenbeispiel vorkommen. Betrachten wir folgende falsche Behauptung, die wir widerlegen sollen:

Behauptung. Seien $\prod_1 = (\mathcal{S}_1, \mathcal{P}_1, \tau_1, \phi_1)$ und $\prod_2 = (\mathcal{S}_2, \mathcal{P}_2, \tau_2, \phi_2)$ zwei Beweissysteme. Wir definieren ein weiteres Beweissystem $\prod_3 = (\mathcal{S}_1 \times \mathcal{S}_2, \mathcal{P}_1 \times \mathcal{P}_2, \tau_3, \phi_3)$ mit

$$\tau_3(s_1, s_2) = 1 \stackrel{def}{\iff} \tau_1(s_1) \neq \tau_2(s_2)$$

und

$$\phi_3((s_1, s_2), (p_1, p_2)) = 1 \stackrel{def}{\iff} \phi_1(s_1, p_1) \neq \phi_2(s_2, p_2)$$

Wenn \prod_3 sound ist, dann ist \prod_1 oder \prod_2 sound.

Die Aussagen im neuen Beweissystem 3 sind Paare von Aussagen, wobei die linke Aussage aus dem Beweissystem 1 und die rechte Aussage aus dem Beweissystem 2 kommt. Eine Aussage in Beweissystem 3 ist wahr, wenn die linke Aussage einen anderen Wahrheitswert als die rechte Aussage hat. Analoges gilt für die Beweisverifikationsfunktion. Nun ist ein Gegenbeispiel gefragt. Wir müssen Beweissysteme \prod_1 und \prod_2 finden, sodass \prod_3 sound ist, aber weder \prod_1 noch \prod_2 sound ist. Um ein Gegenbeispiel bei so einer Aufgabe zu finden, muss man etwas herumprobieren, bis man alle Anforderungen erfüllt. Wir starten mit folgendem einfachem Beweissystem:

$$\begin{aligned} \prod_1 = \prod_2 &= (\{\heartsuit\}, \{\square\}, \tau, \phi) \\ \tau(\heartsuit) &= 0 \\ \phi(\heartsuit, \square) &= 1 \end{aligned}$$

In diesem System gibt es nur eine Aussage \heartsuit und nur einen Beweis \square . Die Aussage \heartsuit ist falsch, weil $\tau(\heartsuit) = 0$. \square ist aber trotzdem ein gültiger Beweis für \heartsuit , weil $\phi(\heartsuit, \square) = 1$. Das heisst es gibt in dem Beweissystem eine falsche Aussage, die einen gültigen Beweis hat. Also ist dieses Beweissystem nicht sound. Wir haben also die erste Bedingung für ein Gegenbeispiel erfüllt: Sowohl \prod_1 als auch \prod_2 ist nicht sound. Schauen wir uns nun \prod_3 wie in der Behauptung definiert an. Die Menge der Aussagen ist $\{(\heartsuit, \heartsuit)\}$, weil das das kartesische Produkt $\{\heartsuit\} \times \{\heartsuit\}$ ist. Analog ist die Menge der Beweise in \prod_3 $\{(\square, \square)\}$. Es gilt $\tau_3(\heartsuit, \heartsuit) = 0$, weil $\tau(\heartsuit) = \tau(\heartsuit)$. Um das zu prüfen, haben wir lediglich die Definition von τ_3 benutzt, wobei in unserem Gegenbeispiel $\tau_1 = \tau_2 = \tau$ gilt. Also ist die Aussage (\heartsuit, \heartsuit) in \prod_3 falsch. Es gilt ausserdem $\phi_3((\heartsuit, \heartsuit), (\square, \square)) = 0$, weil $\phi(\heartsuit, \square) = \phi(\heartsuit, \square)$. Auch hier haben wir nur die Definition von ϕ_3 eingesetzt, wobei in unserem Gegenbeispiel $\phi_1 = \phi_2 = \phi$ gilt. Also ist (\square, \square) kein gültiger Beweis für (\heartsuit, \heartsuit) . Wir halten abschliessend fest, dass es für die einzige falsche Aussage (\heartsuit, \heartsuit) in \prod_3 keinen gültigen Beweis gibt, weil der einzige Beweis in \prod_3 (\square, \square) nicht gültig ist. Das bedeutet, dass \prod_3 sound ist und somit ist unser Gegenbeispiel gültig. Wir müssen dieses Argument nur noch formal aufschreiben.

Beweis. Die Behauptung ist falsch. Betrachte

$$\begin{aligned} \prod_1 = \prod_2 &= (\{\heartsuit\}, \{\square\}, \tau, \phi) \\ \tau(\heartsuit) &= 0 \\ \phi(\heartsuit, \square) &= 1 \end{aligned}$$

\prod_1 und \prod_2 sind beide nicht sound, weil es in beiden eine falsche Aussage \heartsuit gibt mit $\phi(\heartsuit, \square) = 1$. Es gilt nach Definition von \prod_3

$$\prod_3 = (\{(\heartsuit, \heartsuit), (\square, \square)\}, \tau_3, \phi_3)$$

Wegen $\phi(\heartsuit, \square) = \phi(\heartsuit, \heartsuit)$ und der Definition von ϕ_3 folgt

$$\phi_3((\heartsuit, \heartsuit), (\square, \square)) = 0$$

Das heisst es gibt in \prod_3 keine Aussage, sodass es für diese Aussage einen gültigen Beweis gibt. Also kann es auch keine falsche Aussage mit einem gültigen Beweis geben. Es folgt, dass \prod_3 sound ist. \square

3.3 Beweis einer wahren Eigenschaft

Wenn wir eine wahre Eigenschaft über Beweissysteme zeigen wollen, dürfen wir uns nicht wie beim Widerlegen ein Beweissystem herauspicken. Wir müssen unseren Beweis über alle beliebigen Beweissysteme führen. Wir betrachten wieder eine Aussage über die Kombination von zwei Beweissystemen.

Behauptung. Seien $\prod_1 = (\mathcal{S}_1, \mathcal{P}_1, \tau_1, \phi_1)$ und $\prod_2 = (\mathcal{S}_2, \mathcal{P}_2, \tau_2, \phi_2)$ zwei Beweissysteme. Wir definieren ein weiteres Beweissystem $\prod_3 = (\mathcal{S}_1 \times \mathcal{S}_2, \mathcal{P}_1 \times \mathcal{P}_2, \tau_3, \phi_3)$ mit

$$\tau_3(s_1, s_2) = 1 \stackrel{def}{\iff} \tau_1(s_1) = 1 \text{ oder } \tau_2(s_2) = 1$$

und

$$\phi_3((s_1, s_2), (p_1, p_2)) = 1 \stackrel{def}{\iff} \phi_1(s_1, p_1) = 1 \text{ oder } \phi_2(s_2, p_2) = 1$$

Wenn \prod_3 komplett ist, dann ist \prod_1 oder \prod_2 komplett.

Wenn man eine Implikation mit einer Veroderung auf der rechten Seite zeigen muss, empfiehlt sich ein indirekter Beweis. Denn es ist meist einfacher, aus einer Veroderung etwas herzuleiten als eine Veroderung zu zeigen, weil das meistens auf eine Fallunterscheidung hinausläuft.

Wir dürfen im Beweis also annehmen, dass \prod_1 und \prod_2 beide nicht vollständig sind. Dann müssen wir zeigen, dass \prod_3 nicht vollständig ist, wir müssen also eine wahre Aussage in \prod_3 finden, für die es keinen gültigen Beweis gibt.

Beweis. Seien $\prod_1 = (\mathcal{S}_1, \mathcal{P}_1, \tau_1, \phi_1)$ und $\prod_2 = (\mathcal{S}_2, \mathcal{P}_2, \tau_2, \phi_2)$ zwei Beweissysteme, die beide nicht vollständig sind. Weil \prod_1 nicht vollständig ist, gibt es eine Aussage $s_1 \in \mathcal{S}_1$ mit $\tau_1(s_1) = 1$ sodass für alle $p_1 \in \mathcal{P}_1$ $\phi_1(s_1, p_1) = 0$ gilt. Weil \prod_2 nicht vollständig ist, gibt es eine Aussage $s_2 \in \mathcal{S}_2$ mit $\tau_2(s_2) = 1$ sodass für alle $p_2 \in \mathcal{P}_2$ $\phi_2(s_2, p_2) = 0$ gilt. Sei nun \prod_3 wie in der Behauptung definiert. Aus der Definition von \prod_3 und $\tau_1(s_1) = 1$ folgt $\tau_3(s_1, s_2) = 1$. Sei nun $(p_1, p_2) \in \mathcal{P}_1 \times \mathcal{P}_2$ beliebig. Wir haben schon gezeigt, dass $\phi_1(s_1, p_1) = 0$ und $\phi_2(s_2, p_2) = 0$. Also gilt nach der Definition von ϕ_3 auch

$$\phi_3((s_1, s_2), (p_1, p_2)) = 0$$

Insgesamt haben wir gezeigt, dass für alle $(p_1, p_2) \in \mathcal{P}_1 \times \mathcal{P}_2$

$$\phi_3((s_1, s_2), (p_1, p_2)) = 0$$

Weil $\tau_3(s_1, s_2) = 1$ ist \prod_3 nicht vollständig. □

Die Idee des Beweises noch mal informell zusammengefasst: Wir haben zwei Beweissysteme gegeben, die beide unvollständig sind, also wahre Aussagen ohne gültigen Beweis enthalten. Nennen wir diese beiden wahren Aussagen ohne Beweis s_1 und s_2 . Nun betrachten wir die Kombination dieser zwei Beweissysteme, in der Aussagen und Beweise Paare von Aussagen und Beweisen der ursprünglichen Beweissysteme sind. Eine Aussage in dem neuen System, also ein Aussagenpaar, ist wahr, wenn eine der beiden Aussagen im Paar wahr ist. Das Paar (s_1, s_2) ist deshalb eine wahre Aussage. Aber diese Aussage hat keinen Beweis, weil jedes Beweispaar (p_1, p_2) kein gültiger Beweis ist. Das liegt daran, dass p_1 nie ein gültiger Beweis für s_1 ist und p_2 nie ein gültiger Beweis für s_2 ist, weil s_1 und s_2 ja nach Annahme keinen gültigen Beweis haben.