

§6. WPNC-BERECHENBARKEIT, ZÄHLERAUTOMATEN

Im vorangegangenen Paragraphen haben wir u.a. die Auswirkungen von inhibitor-Kanten auf dimensionsbezogene Semilinearitätsgrenzen studiert – mit dem etwas überraschenden Ergebnis, daß kontrollierende Zustände in dieser Hinsicht wirkungsvoller sind. Dennoch sind die inhibitor-Kanten ein Mittel, das qualitativ mehr ermöglicht, als nur das Basismodell. Das belegt z.B. die Unentscheidbarkeit des Erreichbarkeitsproblems, kann aber auch im Zusammenhang mit der Simulation von Berechnungsvorgängen beobachtet werden. Wir beginnen mit einer partiellen Vorstellung des durch Rabin angedachten Konzeptes der WPNC-Berechenbarkeit.

Definition 6.1. Sei \mathcal{P} ein Petri-Netz mit einer Stellenmenge, die aus folgenden, paarweise disjunkten Teilmengen besteht:

$$S = \{o\} \cup \{s\} \cup X \cup R \text{ mit } X = \{x_1, \dots, x_n\}, R = \{r_1, \dots, r_k\}.$$

Ferner gelte für alle Transitionen $t \in T$: $\xi(o, t) = 0$ und $\xi(t, x_j) = 0$ für alle $x_j \in X$, sowie $\xi(t, s) \geq \xi(s, t)$. \mathcal{P} mit einer Initialmarkierung $m_I(R) = (m_I(r_1), \dots, m_I(r_k))$ ist ein schwacher Petri-Netz-Computer für eine Funktion $f: D \rightarrow \mathbb{N}$ mit $D \subseteq \mathbb{N}^n$ (kurz: ein $m_I(R)$ -WPNC für f), wenn für alle Anfangsmarkierungen $m_o(S) \in \mathbb{N}^{n+k+2}$ mit $m_o(R) = m_I(R)$, $m_o(s) = 1$, $m_o(o) = 0$ und $m_o(X) \in D$ gilt:

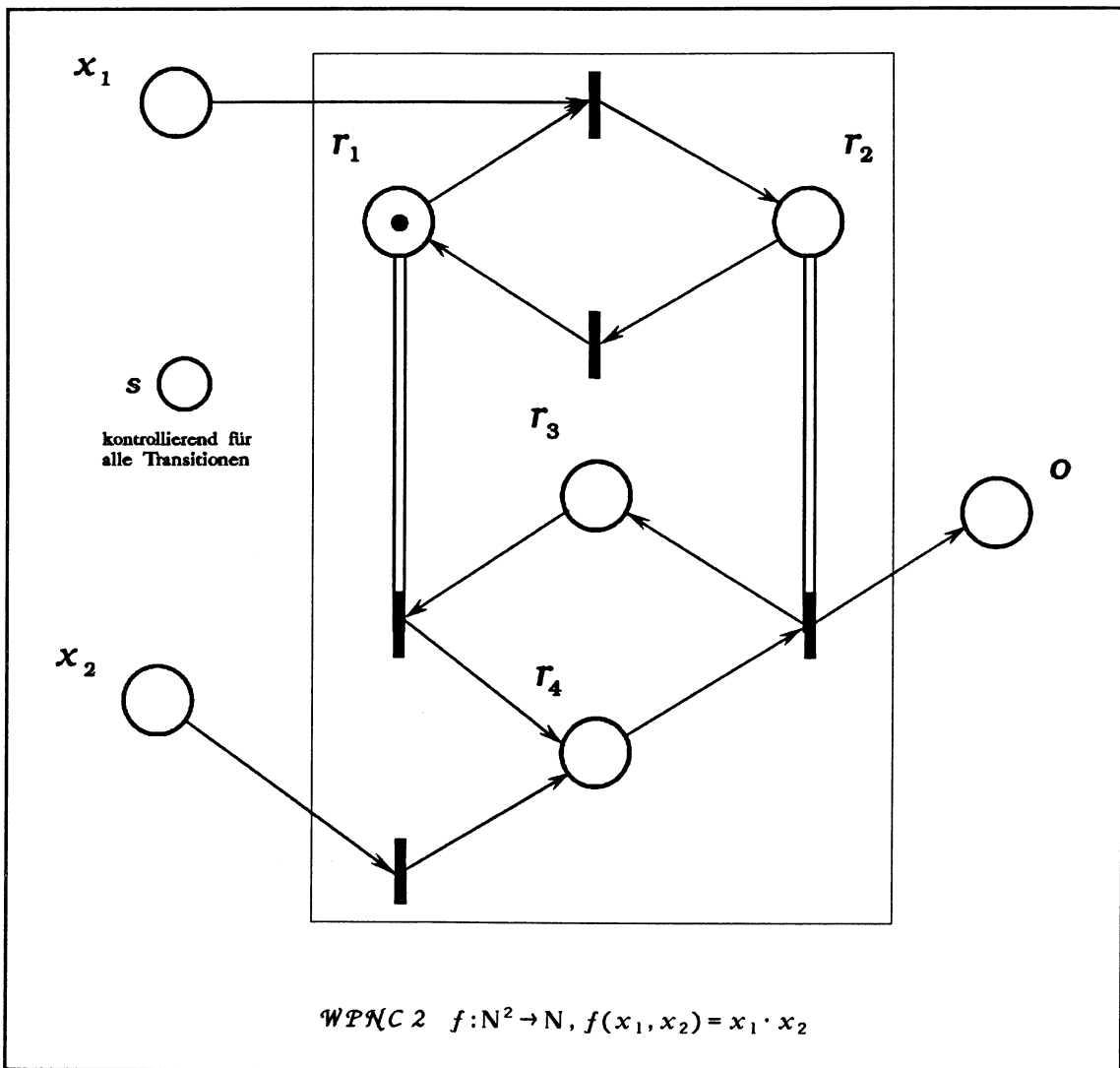
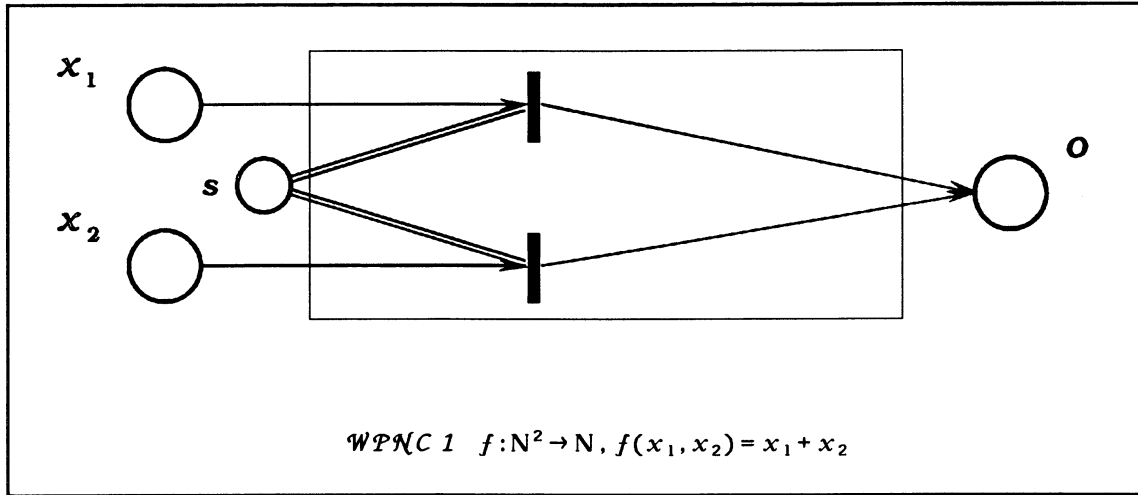
$$\{[\mathfrak{R}_{\mathcal{P}}(m_o(S))] \downarrow_{m(X)-o}\} \downarrow_o = \{0, 1, \dots, f(m_o(X))\}.$$

Existiert ein $m_I(R)$ -WPNC für eine Funktion f , dann ist diese Funktion WPNC-berechenbar.

Die Stelle s ist eine Art „Handbremse“ und kontrolliert meistens alle Transitionen. Wird sie mit einem Token belegt, so kann der Berechnungsvorgang beginnen. Es muß natürlich für die entsprechende Initialmarkierung der inneren Stellen, sowie eine Input-Markierung gesorgt sein. Die Ausgabestelle o darf nicht größer werden als $f(x)$. Andererseits muß $m(o) = f(x)$ unbedingt erreichbar sein.

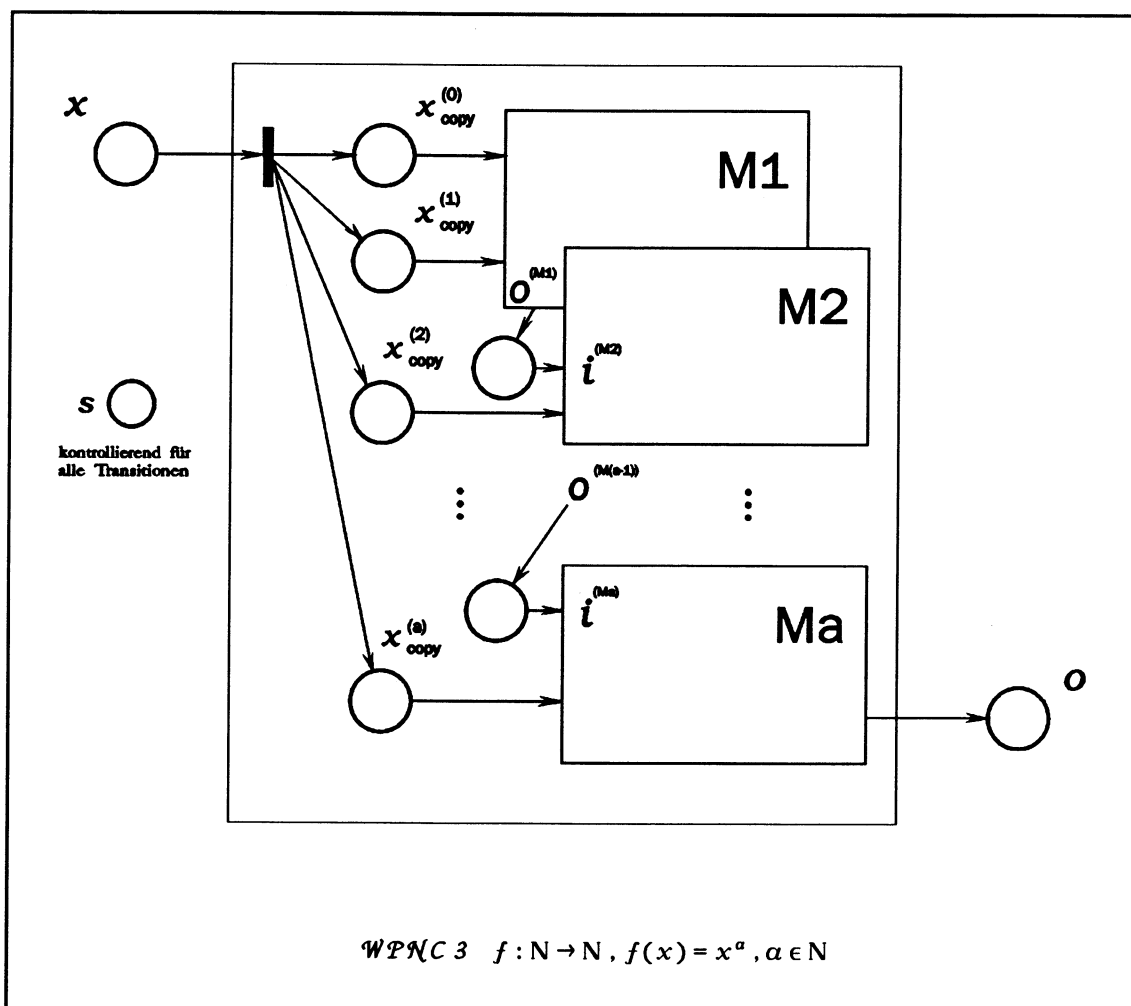
Wir wenden uns den ersten Beispielen zu. Besonders einfach ist der Addierer (WPNC 1), der keine inneren Stellen besitzt. Der Multiplizierer (WPNC 2) dagegen benötigt vier solche Stellen und unbedingt eine bestimmte Initialmarkierung (hier: $m_I(R) = (1, 0, 0, 0)$). Die Korrektheitsbeweise überlassen wir dem Leser mit dem Verweis auf [18],[27].

§6. WPNC-BERECHENBARKEIT. ZÄHLERAUTOMATEN.



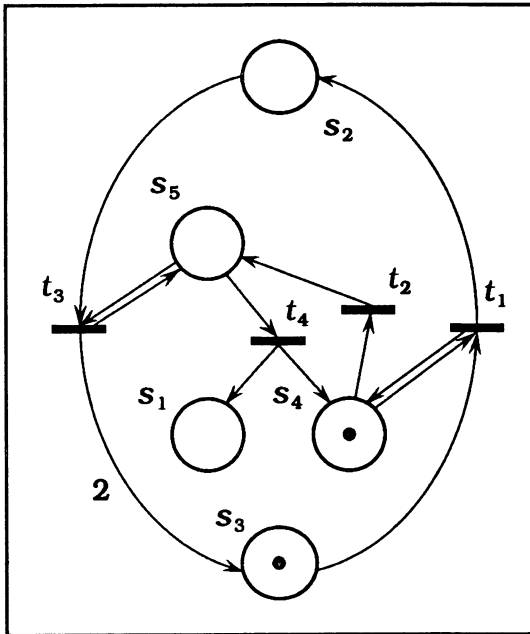
96. WPNC-BERECHENBARKEIT. ZÄHLERAUTOMATEN.

Die WPNC-Berechenbarkeit der Addition und Multiplikation liefert sofort einen Petri-Netz-Computer für Polynome, d.h. $f: \mathbb{N} \rightarrow \mathbb{N}$, $f(x) = x^\alpha$ für ein $\alpha \in \mathbb{N}$. Eine mögliche (wenn auch nicht die kompakteste) Konstruktion ist die α -fache Kaskade von Multiplizierern (WPNC 3). Die Eingabe wird auf Stellen kopiert, die jeweils den zweiten Operanden für entsprechende Sektionen liefern. Die erste Sektion multipliziert die Eingabe mit sich selbst und liefert den ersten Operanden an die zweite Sektion. Diese wiederum multipliziert (höchstens) x^2 mit x , usw.



Durch geeignete Kombination der bisher diskutierten Gebilde können sogar Polynome berechnet werden. Eine kompaktere Konstruktion findet der Leser in [18].

Um den Zusammenhang zwischen nicht-semilinearen Erreichbarkeitsmengen und der WPNC-Berechenbarkeit hinreichend schwieriger Funktionen zu beleuchten, berechnen wir die Exponentialfunktion $f: \mathbb{N} \rightarrow \mathbb{N}$, $f(x) = \alpha^x$ für ein $\alpha \in \mathbb{N}$ (im Beispiel zur Basis 2),



mit Hilfe des Petri-Netzes aus Abb. 2. Laut Lemma 4.4, zusammen mit den Erkenntnissen aus dem §3 (gegenseitige Simulierbarkeit der Petri-Netze aus den Beispielen Abb. 2 und Abb. 4), sind alle Markierungen mit

$$m(s_2) + \frac{m(s_3)}{2} \leq 2^{m(s_1)}$$

erreichbar und nur solche Markierungen. Folglich können die Stellen s_2 und s_3 nur so groß werden, wie es die s_1 erlaubt. Schließen wir die Input-Stelle an die Transition t_4 an, so wird stets gelten:

$$m(s_2) + \frac{m(s_3)}{2} \leq 2^x.$$

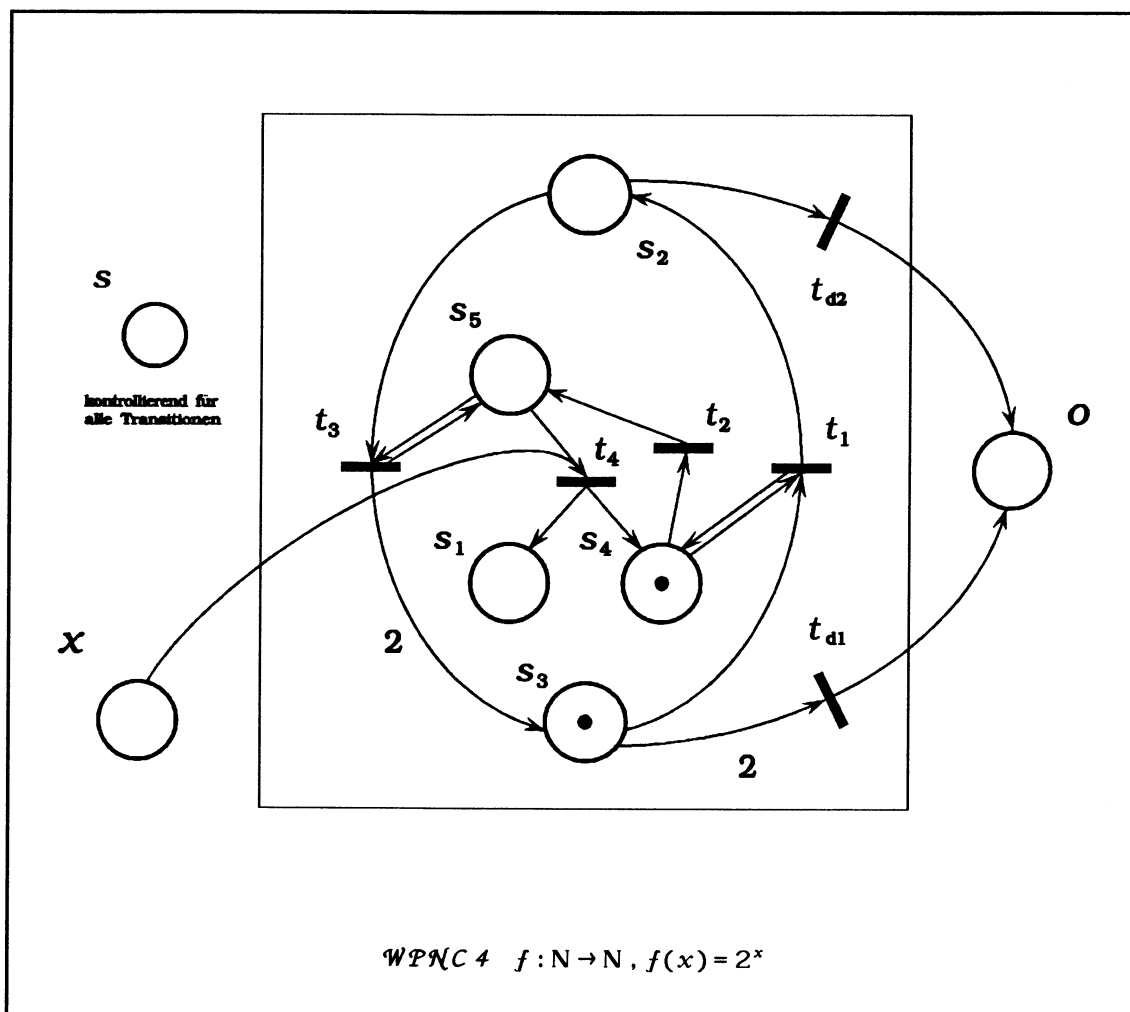
Das Ergebnis ist dann den Stellen s_2 und s_3 , unter Berücksichtigung der notwendigen Halbierung von $m(s_3)$ zu entnehmen. Diese Halbierung, wie überhaupt Division durch Konstanten, ist durch geeignete Kantenvielfachheiten problemlos realisierbar. Damit haben wir bewiesen, daß WPNC 4 die Funktion $f(x) = 2^x$ korrekt berechnet.

Immer wieder machen wir die Erfahrung, daß sogar ein starkes Wachstum von Funktionen keine unlösbaren Probleme bei der Berechnung im Sinne der Definition 6.1 bereitet. Sogar so stark wachsende Funktionen, wie die Ackermann-Funktion, sind WPNC-berechenbar.¹

All diese Funktionen haben allerdings eines gemeinsam: sie sind nämlich monoton wachsend. Diese Eigenschaft begünstigt natürlich die WPNC-Berechenbarkeit, weil ja unsere Definition das „Herantasten an $f(x)$ von unten“ ausdrücklich gestattet. Aber gerade dieser Umstand läßt Schwierigkeiten bei der Berechnung von anderen Funktionen vermuten. Nehmen wir z.B. das boolesche „NOT“. Diese sehr einfache, aber eben nicht monoton wachsende Funktion ist nicht WPNC-berechenbar. Das liegt einfach an der Natur der gewöhnlichen Petri-Netze. Zwar haben wir im §3 festgestellt, daß das „Testen auf \leq “ u.U. simulierbar ist (z.B. durch komplementäre Stellen), aber das würde die Initialmarkierung von der Eingabe abhängig machen. Bei einer festen Initialmarkierung, wie in der Definition 6.1 gefordert, erreichen wir jeden Punkt aus $\mathfrak{X}(m_0)$ erst recht bei einer Anfangsmarkierung, die größer ist als m_0 . Folglich ist „NOT“ nicht WPNC-berechenbar.

¹ Realisierung von Rekursionen - siehe [18].

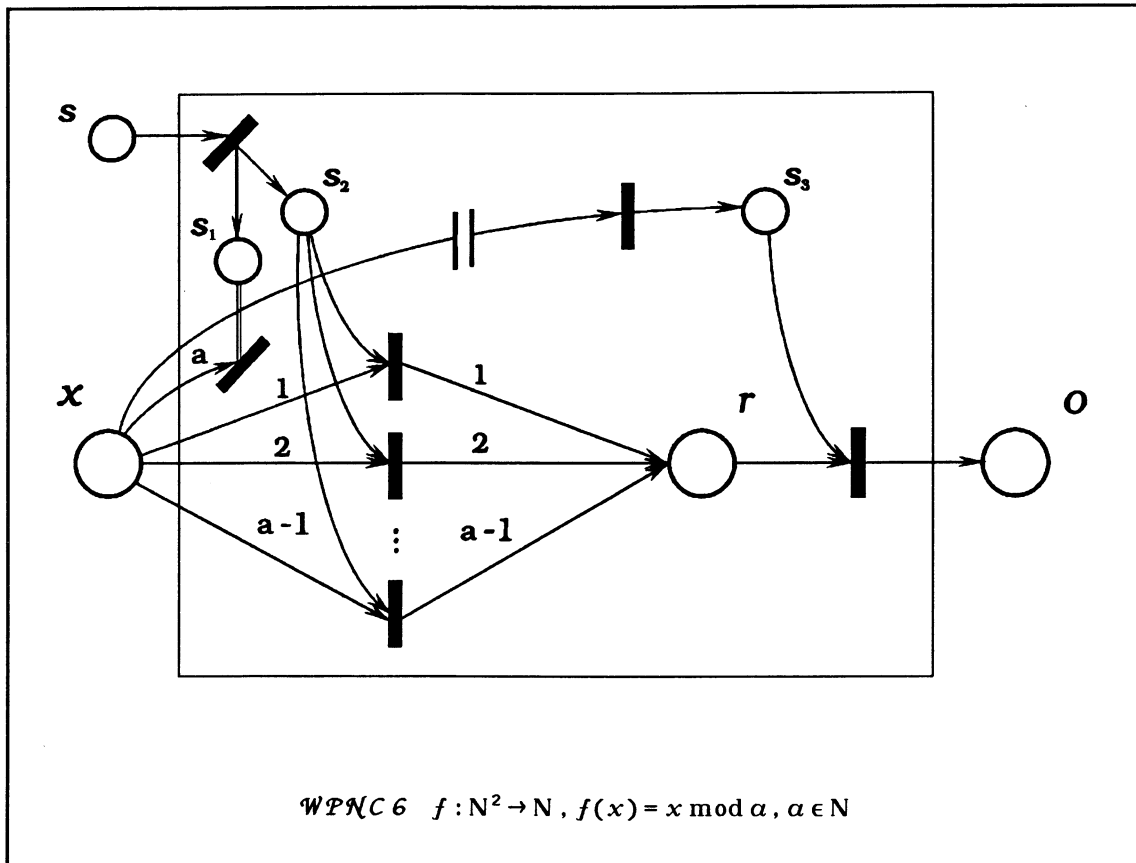
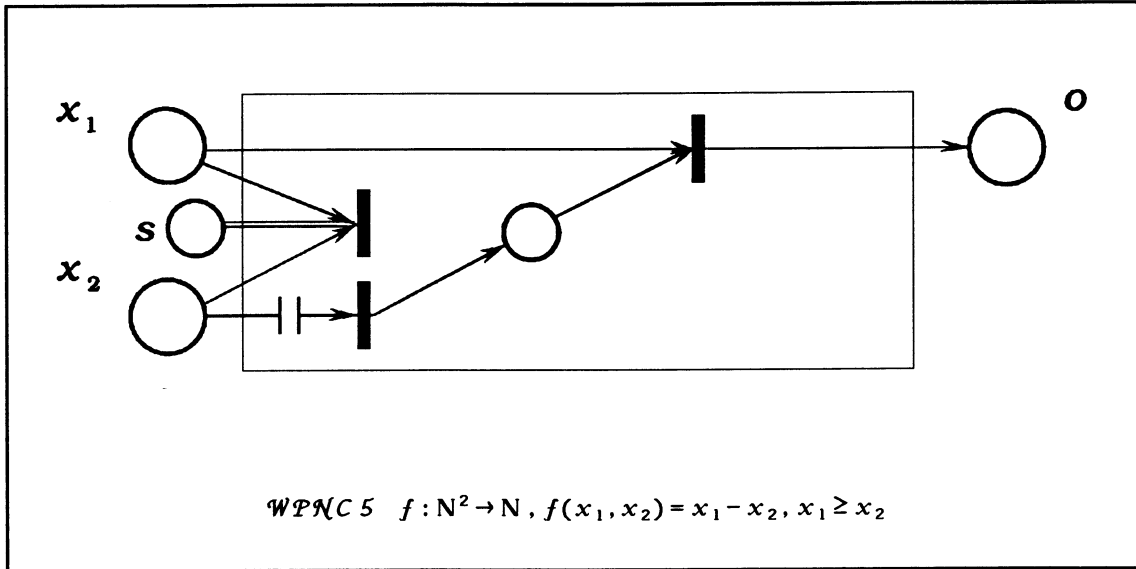
S6. WPNC-BERECHENBARKEIT. ZÄHLERAUTOMATEN.



Diese Überlegung lassen wir als Beweis für die nicht-Berechenbarkeit (auf der Basis der gewöhnlichen Petri-Netze) verschiedener Funktionen gelten. So sind z.B. Subtraktion, Division und Modulo-Division aus dem gleichen Grund nicht WPNC-berechenbar. Diese Funktionen sind mindestens in einem ihrer Argumente nicht wachsend. Zwei inhibitor-Kanten dürften da Abhilfe schaffen (wo sie doch sogar Turing-Maschinen simulieren können). Eine andere Frage ist, ob bereits eine einzelne inhibitor-Kante die ansonsten nicht gegebene Berechenbarkeit impliziert. Sicherlich ist das beim „NOT“ der Fall. Aber auch die Subtraktion realisieren wir problemlos mit Hilfe von WPNC 5.

Problematisch ist die Modulo-Division, die hinsichtlich beider Argumente die notwendige Monotonie vermissen läßt. Höchstwahrscheinlich sind hier zwei inhibitor-Kanten erforderlich, weil jede solche Kante nur eine Stelle auf 0 testen kann. Auf solche Funktionen wollen wir hier nicht näher eingehen. Stattdessen untersuchen wir die Modulo-Division zu einer konstanten Basis ($f: \mathbb{N} \rightarrow \mathbb{N}, f(x) = x \bmod a$ für ein $a \in \mathbb{N}$), die offensichtlich nicht

§6. WPNC-BERECHENBARKEIT. ZÄHLERAUTOMATEN.



56. WPNC-BERECHENBARKEIT. ZÄHLERAUTOMATEN.

WPNC-berechenbar ist. Auch diese Funktion können wir unter Zuhilfenahme einer inhibitor-Kante berechnen. In WPNC 6 kann die Input-Stelle nur dann leer werden, wenn die innere Stelle r exakt $f(x) = x \bmod a$ Token enthält, die nur in diesem Falle auf o transferiert werden können.

Funktionen, die in nur einem ihrer Argumente nicht wachsend sind, konnten wir bislang mit Hilfe genau einer inhibitor-Kante berechnen. Trifft das aber auf alle solche Funktionen zu, d.h. sind sie alle - sagen wir - WPNC¹¹- berechenbar?² Die ganzzahlige Division $f: \mathbb{N}^2 \rightarrow \mathbb{N}$, $f(x_1, x_2) = \lfloor x_1 / x_2 \rfloor$ scheint dem zu widersprechen. Hier können wir aus verständlichem Grund nicht mehr so vorgehen, wie bei der Modulo-Division. Auf diese und andere offene Fragen werden wir in der abschließenden Diskussion ausführlich eingehen. Die nachfolgenden Überlegungen, die sich mehr auf die Anzahl der Stellen hinsichtlich oberer Schranken beziehen, hängen ebenfalls mit diesen Fragen zusammen, wie wir gleich sehen werden. Zuvor wollen wir aber unbedingt festhalten, was die Subtraktion und Modulo-Division belegen, nämlich die folgende doppelte Inklusion zwischen Klassen von Funktionen, die durch WPNC's mit oder ohne inhibitor-Kanten berechenbar sind:

$$\text{WPNC} \subset \text{WPNC}^{11} \subset \text{WPNC}^{21}.$$

Auf den Einsatz von kontrollierenden Zuständen in WPNC's haben wir bislang verzichtet - aus Gründen der Übersichtlichkeit. Interessant wäre nun zu überlegen, inwieweit die bislang vorgestellten WPNC's kompaktiert werden könnten, falls man auf kontrollierende Zustände zurückgreifen würde. Diese können nichts an der Berechenbarkeit einer gegebenen Funktion ändern - egal, ob im Sinne einer der Klassen aus der obigen Relation oder im Sinne der WPNC-Berechenbarkeit überhaupt (Lemma 3.2, §3). Die Frage, ob alle Funktionen WPNC-berechenbar sind, werden wir im folgenden ebenfalls diskutieren, wobei noch näher zu präzisieren bleibt, was hier unter „allen“ Funktionen zu verstehen ist. Wir beschränken uns auf einargumentige und berechenbare Funktionen. Sei also:

$$\mathcal{F} = \{f: \mathbb{N} \rightarrow \mathbb{N} \mid f \text{ algorithmisch berechenbar}\}$$

Resultate, die für unsere Überlegungen wertvoll sind, finden wir vor allem in der Arbeit von Barsdin [30], die sich auf ein anderes, mit Petri-Netzen jedoch verwandtes Maschinenmodell bezieht. Es handelt sich um sog. Zählerautomaten (auch Minsky-Maschinen). Das in [30] diskutierte Spezialmodell (o.B.d.A.) dieser Maschinen wollen wir nun vorstellen.

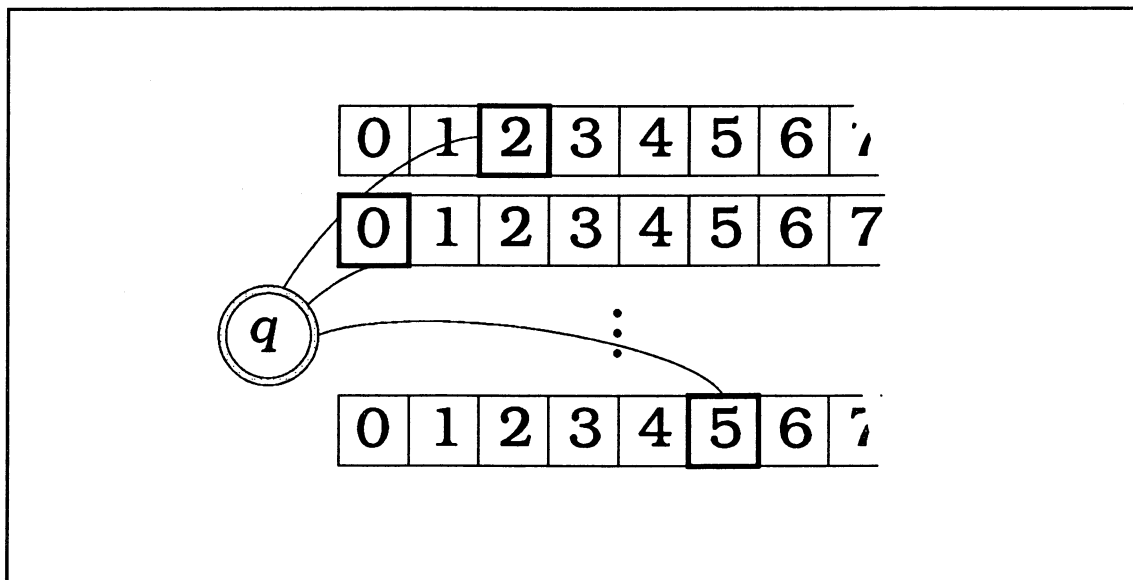
Definition 6.2. Ein n -Band Zählerautomat \mathcal{CM}^n ist ein Tupel $\mathcal{CM}^n = (Q, \chi)$ mit einer endlichen Menge von Zuständen $Q = \{q_1, \dots, q_m\}$ und einer Abbildung $\chi: (Q \times \mathbb{N}^n) \rightarrow (Q \times \mathbb{Z}^n)$ mit der Eigenschaft:

² Selbstverständlich können hier nur Funktionen gemeint sein, die überhaupt (algorithmisch) berechenbar sind.

$$\forall x, x' \in \mathbb{N}^n, q_j \in Q: \chi(q_j, x) = \chi(q_j, x')$$

$$\text{falls } \{i \in \{1, \dots, n\} \mid x_i = 0\} = \{i \in \{1, \dots, n\} \mid x'_i = 0\}.$$

Ob die Äquivalenz von Zählerautomaten zu einer bestimmten Klasse von Petri-Netzen besteht (zu Vektor-Additionssystemen mit inhibitor-Vektoren und kontrollierenden Zuständen etwa), werden wir später untersuchen. Zunächst aber wollen wir uns die Frage stellen, ob dieses Gebilde als eine Art Turing-Maschine aufgefaßt werden kann. Die Theorie von Mehrband-Turing-Maschinen wird als bekannt vorausgesetzt. Die Abbildung χ können wir sicherlich als eine Bewegungs- und gleichzeitig Zustandsüberföhrungsfunktion ansehen. Zwar sieht das Grundkonzept von Minsky ein beliebiges Bandalphabet vor - die Berechnungsschritte sind aber ohnehin nur von der Position der Leseköpfe (deshalb auch Zählerköpfe genannt) und von dem vorliegenden Zustand abhängig. Deswegen folgen wir hier der Idee von Barsdin [30] und gehen davon aus, daß die (unveränderbaren) Inhalte der einseitig begrenzten Bänder als die Position der entsprechenden Sektoren interpretiert werden. Folglich nehmen wir o.B.d.A. an, daß die Sektoren auf jedem Band mit Null beginnend durchnummeriert sind, wie unten dargestellt. Die andere Eigenart einer solchen Turing-Maschine besteht darin, daß der vorliegende Zustand zu jedem Zeitpunkt für alle Zählerköpfe gilt. Weiter einigen wir uns darauf, daß ein Bandüberlauf nicht zulässig ist, was der Natur von Petri-Netzen entspricht.



Ein Zählerautomat ist also eine Turing-Maschine, sicherlich eine deterministische, was an der Abbildung χ liegt. Dieser Determinismus wirft allerdings ein anderes Problem auf. Wir können zwar jeden Zählerautomaten durch ein Petri-Netz mit inhibitor-Kanten

simulieren, aber nicht umgekehrt. Die bei Petri-Netzen vorstellbare Situation, wo zwei und mehr Transitionen gleichzeitig aktiviert sind, würde bei den Zählerautomaten bedeuten, daß für eine Konfiguration (x, q) die Nachfolgekonfiguration nicht eindeutig bestimmt wäre, was auf Nichtdeterminismus hinauslaufen würde. Halten wir aber fest, daß Zählerautomaten zu sog. single-path Petri-Netzen (mit inhibitor-Kanten und kontrollierenden Zuständen) äquivalent sind, d.h. solchen, bei denen höchstens eine Transition aktiviert sein kann.³

Definition 6.3. Eine Funktion $f \in F$, $f: N \supseteq D \rightarrow N$ ist $CM^{\mathbb{N}}$ -berechenbar, wenn es einen Zählerautomaten $CM^{\mathbb{N}}$ gibt, der für alle Anfangskonfigurationen $(x, 0, \dots, 0, q_0)$, $x \in D$, $q_0 \in Q$ immer in endlicher Zeit und in der Endkonfiguration $(0, \dots, 0, f(x), q)$ hält, für ein $q \in Q$.

Theorem 6.1. Jede Funktion $f \in F$ ist $CM^{\mathbb{Z}}$ -berechenbar. (Minsky), [30]⁴ \square

Die ohnehin abgeschwächte (da lediglich nicht-reflexive) Relation zwischen Petri-Netzen und Zählerautomaten ist auf Zählerautomaten und WPNC's, und damit auch auf WPNC^k- und $CM^{\mathbb{N}}$ -Berechenbarkeit, nicht übertragbar. Wir müßten noch zusätzlich sicherstellen, daß ein ausgewählter Zählerkopf den zu berechnenden Funktionswert nicht überschreitet. Deshalb sind wir kaum in der Lage, bekannte Resultate auf das jeweils andere Gebilde (oder die andere Art der Berechenbarkeit) zu verallgemeinern.

Auf noch größere Probleme stoßen wir beim Versuch, die Semilinearität der Petri-Netz-Erreichbarkeitsmengen auf Zählerautomaten zu übertragen. Letztere halten ja in endlicher Zeit, wenn sie denn eine Funktion aus F berechnen sollten. Deshalb wollen wir die Linearität aus dem nachfolgenden Theorem nicht mit der Semilinearität gleichsetzen. Trotzdem darf anhand der bislang erzielten Resultate (§5) zumindest vermutet werden, daß eine noch effizientere Berechenbarkeit von Funktionen aus F eher nicht zu erwarten ist - allenfalls für Funktionen, die eben eine gewisse „Linearität“ aufweisen. Barsdin gelang es, diese Eigenschaft genauer anzugeben:

Theorem 6.2. Für jede $CM^{\mathbb{Z}}$ -berechenbare Funktion $f \in F$ existieren Konstanten $\phi, \omega \in \mathbb{N}$ mit der Eigenschaft: \rightarrow

³ Die Abbildung χ läßt zwar kontrollierende Stellen nicht zu - wohl aber rückgekoppelte inhibitor-Transitionen. Man könnte folglich die Zählerautomaten als Vektor-Additionssysteme mit rückgekoppelten inhibitor-Transitionen (und kontrollierenden Zuständen) bezeichnen. Diese "Rückkopplungen" fallen jedoch wegen Lemma 3.1 überhaupt nicht ins Gewicht.

⁴ Bereits aus dem Resultat von Lambek [a] (erzielt in einem anderen Zusammenhang), kann die $CM^{\mathbb{N}}$ -Berechenbarkeit von allen Funktionen $f \in F$ gefolgert werden. Der Bedarf an Bändern ist allerdings hier nicht herleitbar. Diese Folgerung ermöglicht erst das Resultat von Minsky [b]. Siehe dazu [30].

[a] Lambek, Joachim, "How to program an infinite abacus", Canad. Math. Bull., 1964, 4, Nr. 3.

[b] Minsky, Marvin, "Recursive unsolvability of Post's problem of Tag and other topics in theory of Turing machines", Annals of Mathem., 1961, 74, Nr. 3.

$$f(x + m\omega) = f(x) + m\phi, \quad m = 0, 1, 2, \dots \quad (\text{Barsdin}), [30] \quad \square$$

Die Minsky-Barsdin Theoreme sind sicherlich sehr wertvolle Resultate, deren Konsequenzen für Petri-Netze (und vor allem für WPNC's) interessant sind. Es sei allerdings ausdrücklich betont, daß es sich hier lediglich um wettbewerbsfreie Petri-Netze handelt und um eine Eigenschaft, die nicht gleichbedeutend ist mit der Semilinearität der Erreichbarkeitsmengen. Deshalb sind die Theoreme 5.4 und 5.5 allemal stärker, als jede direkte Folgerung aus 6.1 und 6.2.

Im nachfolgenden Korollar beschäftigen wir uns mit den härtesten Funktionen $f \in \mathcal{F}$ hinsichtlich der CM^n -Berechenbarkeit. Aufgrund der fehlenden Äquivalenz zwischen WPNC's und Zählerautomaten können wir kaum brauchbare Aussagen über WPNC-Komplexität von \mathcal{F} machen. Deshalb:

Korollar 6.3. Es gibt Funktionen aus \mathcal{F} , die mit Hilfe von zweidimensionalen WPNC's (mit inhibitor-Kanten und kontrollierenden Zuständen) nicht berechenbar sind. □

Damit hätten wir geklärt, wie die Kompaktierung von WPNC's durch kontrollierende Zustände aussehen und vor allem, wie wirksam sie sein kann. Der Schluß auf die Dimension von WPNC's ohne kontrollierende Zustände ist wegen Lemma 3.3 trivial. Allerdings ist damit noch nichts über die Anzahl der Transitionen, insbesondere der inhibitor-Kanten gesagt. Ferner resultiert aus der CM^n -Berechenbarkeit einer gegebenen Funktion lediglich die untere Schranke für den kompaktesten WPNC (mit inhibitor-Kanten und kontrollierenden Zuständen), der sie berechnet. Auf diese Fragen kommen wir im nächsten und zugleich letzten Kapitel dieser Arbeit zurück.
