Wir betrachten nun **Maschinen über** $\mathbb{N}^*$.

Wir repräsentieren $\mathbb{N}^*$ durch

$$B := \{\varrho : \mathbb{N} \longrightarrow \mathbb{N} \mid \sum_{i=1}^{\infty} \varrho(i) < \infty\}.$$

Die Elemente von B können wir als Rechenbänder anschauen, die in ∞-viele Felder unterteilt sind; jedes Feld enthält als Inschrift eine natürliche Zahl. Alle bis auf endlich viele Felder sind „leer“.

Wir setzen nun

$$K_1 := B \quad \text{und} \quad K_2 := I := L := \mathbb{N}.$$

Wir lassen also auch zu, daß die Maschine über ∞-viele „interne“ Zustände verfügt. Es seien

$$\Delta_1 : \mathbb{N}^2 \longrightarrow \mathbb{N}^2$$

und

$$\Delta_2 : \mathbb{N}^2 \longrightarrow \{0, 1, -1\}$$

berechenbare Funktionen.

Wir setzen

$$K \;:=\; \{(\varrho,\zeta,i,k) \mid \varrho \in B, \zeta \in \mathbb{N}, i \in \mathbb{N}, k \in \mathbb{N}, i \leq k\} \text{ und } \varrho(l) = 0 \text{ für } l > k$$

Nun definieren wir

$$\Delta : B \times K_2 \times I \times L \longrightarrow B \times K_2 \times I \times L,$$

$$\begin{array}{rrcl} & (\varrho',\zeta',i',k') & = & \Delta(\varrho,\zeta,i,k) \\ :\Longleftrightarrow & \varrho'(l) & := & \begin{cases} \varrho(l) \text{ für } l \neq i, \\ p_1 \circ \Delta_1(\varrho(i),\zeta); \end{cases} \\ & \zeta' & := & p_2 \circ \Delta_1(\varrho(i),\zeta); \\ & i' & := & i + \Delta_2(\varrho(i),\zeta); \\ & k' & := & \begin{cases} k & \text{falls } i' \leq k \\ k+1 & \text{falls } i' := k+1. \end{cases} \end{array}$$

Erläuterung: i bezeichnet die aktuelle Position der Maschine auf dem Band. Δ_1 berechnet in Abhängigkeit von der Inschrift im i-ten Feld und dem aktuellen Maschinenzustand die neue Inschrift $p_1 \circ \Delta_1(\varrho(i),\zeta)$ und den Folgezustand $p_2 \circ \Delta_1(\varrho(i),\zeta)$.

p_i ist hierin die Projektion auf die i-te Komponente. Δ_2 gibt an, ob und wie sich die Maschine auf dem Band weiterbewegt: eine Position nach rechts ($+1$), eine Position nach links (-1) oder die Maschine bleibt an ihrem Ort (0).

Die Funktion Δ ist im intuitiven Sinne berechenbar. Wir definieren als Menge der **Anfangskonfigurationen**

$$K_a := \{(\varrho, 1, i, k) \mid (\varrho, 1, i, k) \in K\}$$

und

$$K_e := \{(\varrho, 0, i, k) \mid (\varrho, 0, i, k) \in K\}$$

als Menge der **Endkonfigurationen**.

Definition: $\mathfrak{M} = (K, K_a, K_e, \Delta)$ heißt Maschine über $\mathbb{N}^*$. $\mathfrak{M}$ heißt $\mathbb{P}$-Maschine, falls Δ_1 und Δ_2 aus $\mathbb{P}$ sind.

Nun wollen wir zeigen, daß wir jede Berechnung auf $\mathfrak{M} = (K, K_a, K_e, \Delta)$ auf einer Maschine $\mathfrak{M}'$ mit $K_1' := \mathbb{N}^k$ simulieren können. Über k werden wir weiter unten verfügen.

Zunächst definieren wir, was wir unter einer **Simulation** verstehen. Das folgende Diagramm, in dem $\sigma : K \longrightarrow K'$ eine Abbildung ist, zeigt einen Maschinen-**Homomorphismus**.

$$\begin{array}{cccc} & K & \stackrel{\Delta}{\longrightarrow} & K & \\ \sigma & \downarrow & & \downarrow & \sigma \\ & K' & \stackrel{\Delta'}{\longrightarrow} & K' & \end{array}$$

Im Falle der **Simulation** lassen wir verallgemeinernd zu, daß $\mathfrak{M}'$ mehrere Rechenschritte ausführt, wenn $\mathfrak{M}$ einen einzigen ausführt.

Definition: Sind $\mathfrak{M}$ und $\mathfrak{M}'$ Maschinen und ist $\sigma : K \longrightarrow K'$ eine Abbildung mit $\sigma(K_a) \subset K'_a$ und $\sigma(K_e) \subset K'_e$, die das Diagramm

$$\begin{array}{cccc} & K & \stackrel{\Delta}{\longrightarrow} & K & \\ \sigma & \downarrow & & \downarrow & \sigma \\ & K' & \stackrel{\Delta'^{(j)}}{\longrightarrow} & K' & \end{array}$$

erfüllt, dann heißt σ **Simulation** von $\mathfrak{M}$ auf $\mathfrak{M}'$. Dabei mag j von dem Argument von Δ abhängen.

Definition: Die Simulation $\sigma : \mathfrak{M} \longrightarrow \tilde{\mathfrak{M}}$ heißt **treu** : $\Longleftrightarrow$ Die Einschränkungen $\sigma \mid K_a$ und $\sigma \mid K_e$ sind bijektiv.

Lemma: Ist $\sigma : \mathfrak{M} \longrightarrow \tilde{\mathfrak{M}}$ eine treue Simulation, und ist f $\mathfrak{M}$ berechenbar, dann läßt sich die Berechnung von f mittels σ auf $\tilde{\mathfrak{M}}$ übertragen.

Beweis: Sei, wie auf Seite 189, T bzw. $\tilde{T}$ die Laufzeitfunktion von $\mathfrak{M}$ bzw. $\tilde{\mathfrak{M}}$. Wir definieren wie dort

$$F(\xi, 0) := (\xi, 1),$$
$$F(\xi, t+1) := \Delta(F(\xi, t)).$$

Analog wird $\tilde{F}$ definiert. Es gilt also

$$F(\xi, T(\xi)) \in K_e, \quad \tilde{F}(\tilde{\xi}, T(\tilde{\xi})) \in \tilde{K}_a.$$

Da σ eine Simulation ist, gibt es $j = j(\xi, t)$, so daß

$$\sigma(F(\xi, t+1)) = \tilde{\Delta}^j(\sigma(F(\xi, t))).$$

Hieraus folgt wegen $\sigma(K_a) \subset \tilde{K}_a$, $\sigma(K_e) \subset \tilde{K}_e$:

$$\sigma(F(\xi, T(\xi))) = \tilde{F}(\sigma(\xi, 1), \tilde{T}(\sigma(\xi))).$$

Da σ bijektiv ist, gilt nun

$$F(\xi, T(\xi)) = \sigma^{-1}(\tilde{F}[\sigma(\xi, 1), \tilde{T}(\sigma(\xi))]).$$

Das heißt aber gerade, daß wir mittels σ die Berechnung von $\mathfrak{M}$ berechenbaren Funktionen auf Berechnungen in $\tilde{\mathfrak{M}}$ zurückführen können.

Satz: Jede $\mathbb{P}$-Maschine über $\mathbb{N}^*$ kann durch eine $\mathbb{P}$-Maschine über $\mathbb{N}^4$ treu simuliert werden.

Beweis: Wir wählen $k = 4$ und definieren

$$\sigma : K \longrightarrow \mathbb{N}^4$$

zunächst, indem wir

$$b_{k+1}(\varrho) := b_{k+1}(\varrho(0), \ldots, \varrho(k))$$

und

$$\sigma(\varrho, \zeta, i, k) := (b_{k+1}(\varrho), \zeta, i, k)$$

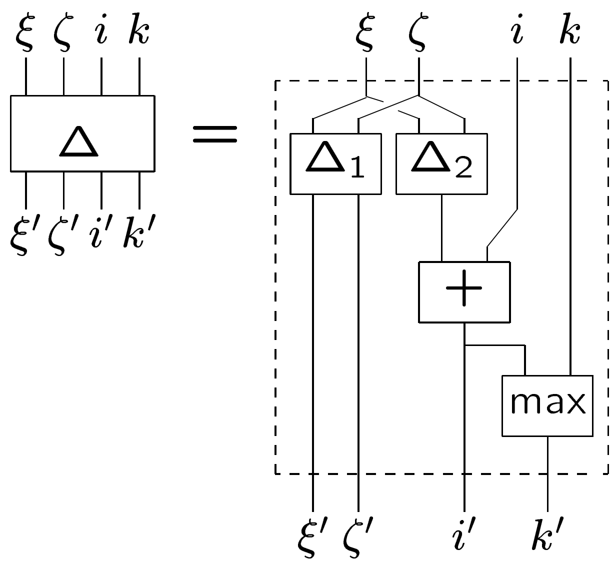
setzen. σ ist bijektiv, da k die Abbildung b_{k+1} eindeutig bestimmt und da b_k für $k = 1, 2, \ldots$ bijektiv ist.

Diese Definition von σ ist naheliegend aber vorläufig. Wir werden mit dieser Definition auf eine Schwierigkeit stoßen, die sich aber durch eine einfache Variation des Ansatzes beheben läßt.

Wir definieren $\tilde{\mathfrak{M}}$ passend zu σ. Sei also

$$(\tilde{\xi}, \zeta, i, k) \in \mathbb{N}^4.$$

Wir fassen die beiden Abbildungen Δ_1 und Δ_2 zu einer Abbildung Δ zusammen, wie es das folgende Netz definiert



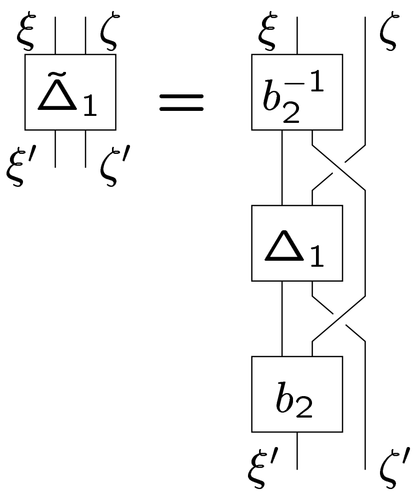
Nun definieren wir $\tilde{\Delta}$ für den Fall $i = 0$ und $k = k'$ durch das Diagramm

$$
\begin{tikzcd}
(\tilde{\xi}, \zeta, i, k) \arrow[r, dashed, "\tilde{\Delta}"] \arrow[d, "\sigma^{-1}"'] & (b_{k+1}(\varrho'), \zeta', i', k) \\
(b_{k+1}^{-1}(\xi), \zeta, i, k) \arrow[r, mapsto, "\Delta"] & (\varrho', \zeta', i', k') \arrow[u, "\sigma"']
\end{tikzcd}
$$

Da die Maschine auf Feld 0 steht, ergibt sich, wenn wir den ζ-Eingang von $\tilde{\Delta}_1$ und Δ_1 vernachlässigen

$$
\begin{aligned}
\tilde{\Delta}_1 &= b_2 \circ (1 \times b_2) \circ \ldots \circ (1_{k-1} \times b_2) \circ (\Delta_1 \times 1_k) \\
&\quad \circ (1_{k-1} \times b_2^{-1}) \circ \ldots \circ (1 \times b_2^{-1}) \circ b_2^{-1} \\
&= b_2 \circ (\Delta_1 \times 1) \circ b_2^{-1}.
\end{aligned}
$$

Wir übersetzen das Resultat nun mit Berücksichtigung des ζ-Eingangs in ein Schaltbild.



Offensichtlich liegt für diesen Fall $\tilde{\Delta}$ in $\mathbb{P}$.

Nun führen wir die Fälle $i < k$ auf diesen Sonderfall zurück, indem wir die Reduktionen

$$b_{k_j i} := b_2 \circ (b_i \times b_{k-i+1})$$

einführen. Offensichtlich gilt

$$b_{k,i} \in \mathbb{P} \quad \text{und} \quad b_{k,i}^{-1} \in \mathbb{P} \quad \text{für } i = 0, \ldots , k-1.$$

Für $i < k$ erhalten wir nun das Diagramm

$$\begin{array}{ccc} (\tilde{\xi}, \zeta, i, k) & \xrightarrow{\tilde{\Delta}} & (b_{k,i}(\varrho'), \zeta', i', k') \\ \sigma^{-1} \downarrow & & \uparrow \sigma \\ (b_{k,i}^{-1}(\tilde{\xi}), \zeta, i, k) & \xrightarrow{\Delta} & (\varrho', \xi', i', k'), \end{array}$$

das $\tilde{\Delta}$ und σ für $0 \leq i < k$ definiert. Setzen wir diese Transformationen hintereinander, dann ergibt sich, wenn wir vereinfachend den Eingang ζ vernachlässigen

$$\begin{aligned} \tilde{\Delta}_1 &= b_{k,i} \circ (1_i \times \Delta_1 \times 1_{k-i}) \circ b_{k,i}^{-1} \\ &= b_2 \circ (b_i \times b_{k-i+1}) \circ (1_i \times \Delta_1 \times 1_{k-i}) \\ &\quad \circ (b_i^{-1} \times b_{k-i+1}^{-1}) \circ b_2^{-1} \\ &= b_2 \circ (1 \times b_{k-i+1} \circ (\Delta_1 \times 1_{k-i}) \\ &\quad \circ (1 \times b_{k-i+1}) \circ b_2^{-1} \\ &= b_2 \circ (1 \times b_2 \circ (\Delta_1 \times 1) \circ b_2^{-1}) \circ b_2^{-1}. \end{aligned}$$

Die für den Fall $i < k$ definierte Abbildung $\tilde{\Delta}$ liegt offensichtlich in $\mathbb{P}$.

Wir betrachten nun den Fall $i = k$. Dieser Fall gliedert sich in zwei Unterfälle:

$$i = k,\ k' = k \text{ und } i = k,\ k' = k + 1.$$

Im zweiten Fall wird ein neues „Feld des Bandes“ in die Berechnung mit einbezogen.

Fall $i = k,\ k' = k.$

Wir setzen

$$b_{k,k} := b_2 \circ (b_k \times 1)$$

und erhalten in diesem Fall

$$\begin{aligned}\tilde{\Delta}_1 &= b_2 \circ (b_k \times 1) \circ (1_k \times \Delta_1) \circ (b_k^{-1} \times 1) \circ b_2^{-1} \\ &= b_2 \circ (1 \times \Delta_1) \circ b_2^{-1}.\end{aligned}$$

Also auch in diesem Fall liegt $\tilde{\Delta}$ in $\mathbb{P}$. Die Vervollständigung unter Einschluß des ζ-Eingangs ist symmetrisch zu dem Fall $i = 0$, der oben dargestellt wurde.

Fall $i = k, k' = k+1$.

Hier erhalten wir

$$\begin{aligned}
\tilde{\Delta}_1 &= b_2 \circ (b_{k+1} \times 1) \circ (1_k \times \Delta_1 \times c_0) \\
&\quad \circ (b_k^{-1} \times 1) \circ b_2^{-1} \\
&= b_2 \circ [b_2 \circ (b_k \times 1) \times 1] \\
&\quad \circ (b_k^{-1} \times \Delta_1 \times c_0) \circ b_2^{-1} \\
&= b_2 \circ [b_2 \circ (b_k \circ b_k^{-1} \times 1) \times 1] \\
&\quad \circ (1 \times \Delta_1 \times c_0) \circ b_2^{-1} \\
&= b_2 \circ [b_2 \times 1] \circ (1 \times \Delta_1 \times c_0) \circ b_2^{-1}.
\end{aligned}$$

Auch diese Funktion liegt in $\mathbb{P}$.

Nun sind die drei betrachteten Fälle primitiv-rekursive Prädikate, so daß unsere Definition von $\tilde{\Delta}$ durch Fallunterscheidung schließlich eine primitiv-rekursive Abbildung $\tilde{\Delta}$ liefert.

Da die definierten Abbildungen σ eine treue Simulation von $\mathfrak{M}$ auf $\tilde{\mathfrak{M}}$ liefern, läßt sich jede $\mathfrak{M}$-berechenbare Funktion auch auf $\tilde{\mathfrak{M}}$ berechnen.

Aus dem Simulationssatz folgt der

Satz: Jede durch eine $\mathbb{P}$-Maschine über $\mathbb{N}^*$ berechenbare Abbildung $f : D \subset \mathbb{N}^k \longrightarrow D' \subset \mathbb{N}^j$ läßt sich auch durch eine $\mathbb{P}$-Maschine über $\mathbb{N}^4$ berechnen.

Beweis: Es bleibt nur zu ergänzen, daß man die Eingabe von $\xi \in D$ in $\mathfrak{M}$ durch die Standardreduktionen vornehmen kann. Das Gleiche gilt für die Ausgabe von $\mathbb{N}^4$ in D'. Die Berechnungen über $\mathbb{N}^4$ lassen sich durch die Standardreduktion auf Berechnungen über $\mathbb{N}$ reduzieren.

Zusammenfassung:

Wir haben abstrakte Maschinen $\mathfrak{M}$ über Mengen $\mathbb{N}^k$ und über $\mathbb{N}$ definiert. Wir haben gezeigt, daß jede über $\mathbb{N}^*$ durch eine $\mathbb{P}$-Maschine berechenbare Funktion auch durch eine $\mathbb{P}$-Maschine über $\mathbb{N}$ berechnet werden kann. Weiter haben wir gezeigt, daß alle die durch diese Maschinen berechenbaren Funktionen μ-rekursiv sind und daß sich jede μ-rekursive Funktion auf diese Weise berechnen läßt.

Die hier betrachteten Maschinen sind Verallgemeinerungen der Turingmaschinen. Somit gilt natürlich auch, daß die turingberechenbaren Funktionen alle μ-rekursiv sind. Da Turingmaschinen zur Allgemeinbildung der Informatiker gehören, werden wir im folgenden Abschnitt die Turingmaschinen einführen und beweisen, daß die μ-rekursiven Funktionen auch bereits auf Turingmaschinen berechenbar sind.

Der hier geschilderte Aufbau der Theorie ist meines Wissens neu. Der Beweis unseres Simulationssatzes ist einfacher und kürzer als der übliche Beweis für die μ-Rekursivität der turingberechenbaren Funktionen.