### 6.2.4 Vom Endzustand zum leeren Stack

Lassen Sie uns nun in die entgegengesetzte Richtung gehen: Wir nehmen einen PDA $P_F$, der eine Sprache $L$ durch Endzustand akzeptiert, und konstruieren einen PDA $P_N$, der die Sprache $L$ durch Leeren seines Stacks akzeptiert. Die Konstruktion ist einfach und in Abbildung 6.7 dargestellt. Wir fügen zu jedem akzeptierenden Zustand von $P_F$ einen Übergang für die Eingabe $\varepsilon$ zu einem neuen Zustand $p$ hinzu. Wenn sich $P_N$ im Zustand $p$ befindet, dann leert er seinen Stack und liest kein weiteres Eingabesymbol ein. Daher wird $P_N$ seinen Stack nach dem Einlesen von $w$ leeren, wenn $P_F$ nach dem Einlesen der Eingabe $w$ in einen akzeptierenden Zustand übergeht.

Um zu verhindern, dass eine Situation simuliert wird, in der $P_F$ zufällig seinen Stack leert, ohne zu akzeptieren, muss $P_N$ eine Stackanfangsmarkierung $X_0$ verwenden. Diese Markierung ist das Startsymbol von $P_N$ und wie bei der Konstruktion von Satz 6.1 muss $P_N$ in einem neuen Zustand $p_0$ beginnen, dessen einzige Funktion darin besteht, das Startsymbol von $P_F$ auf dem Stack abzulegen und zum Startzustand von $P_F$ zu wechseln. Die Konstruktion ist in Abbildung 6.7 skizziert und wird im nächsten Satz formalisiert.

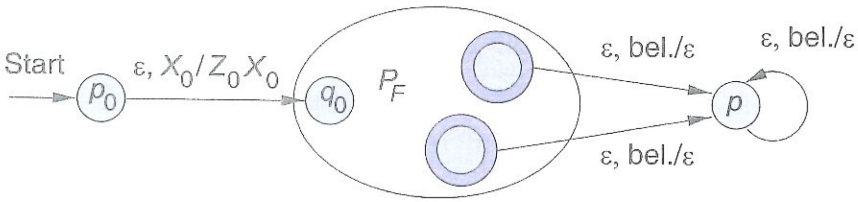


**Abbildung 6.7:** $P_N$ simuliert $P_F$ und leert seinen Stack genau dann, wenn $P_F$ in einen akzeptierenden Zustand übergeht

**Satz 6.4** Sei $L$ die Sprache $L(P_F)$ für einen PDA $P_F$ $(Q, \Sigma, \Gamma, \delta_F, p_0, Z_0, F)$. Dann gibt es einen PDA $P_N$, derart dass $L = N(P_N)$.

**Beweis** Die Konstruktion entspricht der Darstellung in Abbildung 6.7. Sei

$$P_N = (Q \cup \{p_0, p\}, \Sigma, \Gamma \cup \{X_0\}, \delta_N, p_0, X_0),$$

wobei $\delta_N$ definiert wird durch:

1. $\delta_N(p_0, \varepsilon, X_0) = \{(q_0, Z_0X_0)\}$. Wir beginnen, indem wir das Startsymbol von $P_F$ auf dem Stack ablegen und in den Startzustand von $P_F$ wechseln.
2. Für alle Zustände $q$ aus $Q$, Eingabesymbole $a$ aus $\Sigma$ oder $a = \varepsilon$ und $Y$ aus $\Gamma$ gilt, dass $\delta_N(q, a, Y)$ jedes Paar enthält, das in $\delta_F(q, a, Y)$ enthalten ist. Das heißt, $P_N$ simuliert $P_F$.
3. Für alle akzeptierenden Zustände $q$ aus $F$ und Stack-Symbole $Y$ aus $\Gamma$ oder $Y = X_0$ gilt, dass $\delta_N(q, \varepsilon, Y)$ das Paar $(p, \varepsilon)$ enthält. Nach dieser Regel kann, sobald $P_F$

akzeptiert, $P_N$ damit beginnen, seinen Stack zu leeren, ohne weitere Eingabesymbole einzulesen.

4. Für alle Stack-Symbole $Y$ aus $\Gamma$ oder $Y = X_0$ ist $\delta_N(p, \varepsilon, Y) = \{(p, \varepsilon)\}$. Wenn $P_N$ den Zustand $p$ erreicht hat, was nur nach dem Akzeptieren von $P_F$ der Fall sein kann, dann nimmt $P_N$ so lange Symbole von seinem Stack, bis der Stack leer ist, und liest keine weiteren Eingabesymbole ein.

Wir müssen nun beweisen, dass $w$ in $N(P_N)$ genau dann enthalten ist, wenn $w$ in $L(P_F)$ enthalten ist. Der Gedankengang ist ähnlich wie beim Beweis von Satz 6.1. Der »Wenn-Teil« ist eine direkte Simulation und der »Nur-wenn-Teil« erfordert, das wir die beschränkte Zahl der Dinge untersuchen, die der konstruierte PDA $P_N$ ausführen kann.

(Wenn-Teil) Angenommen, für einen akzeptierenden Zustand $q$ und eine Stackzeichenreihe $\alpha$ gelte $(q_0, w, Z_0) \overset{*}{\underset{P_F}{\vdash}} (q, \varepsilon, \alpha)$. Auf Grund der Tatsache, dass jeder Zustandsübergang von $P_F$ eine Bewegung von $P_N$ ist, und durch Anwendung von Satz 6.5, wonach $X_0$ am unteren Ende auf dem Stack gehalten werden kann, wissen wir, dass $(q_0, w, Z_0X_0) \overset{*}{\underset{P_N}{\vdash}} (q, \varepsilon, \alpha X_0)$. Somit kann $P_N$ Folgendes ausführen:

$$(p_0, w, X_0) \underset{P_N}{\vdash} (q_0, w, Z_0X_0) \overset{*}{\underset{P_N}{\vdash}} (q, \varepsilon, \alpha X_0) \overset{*}{\underset{P_N}{\vdash}} (p, \varepsilon, \varepsilon)$$

Die erste Bewegung folgt aus Regel (1) in der Konstruktion von $P_N$, während die letzte Bewegungsfolge den Regeln (3) und (4) entspricht. Somit wird $w$ von $P_N$ durch Leeren des Stacks akzeptiert.

(Nur-wenn-Teil) Der PDA $P_N$ kann den Stack nur leeren, indem er in den Zustand $p$ übergeht, da sich $X_0$ am unteren Ende des Stacks befindet und $P_F$ für $X_0$ keine Bewegungen definiert. In den Zustand $p$ kann $P_N$ wiederum nur wechseln, wenn der simulierte PDA $P_F$ einen akzeptierenden Zustand annimmt. Die erste Bewegung von $P_N$ entspricht mit Sicherheit der nach Regel (1) festgelegten Bewegung. Folglich sieht jede akzeptierende Berechnung von $P_N$ wie folgt aus:

$$(p_0, w, X_0) \underset{P_N}{\vdash} (q_0, w, Z_0X_0) \overset{*}{\underset{P_N}{\vdash}} (q, \varepsilon, \alpha X_0) \overset{*}{\underset{P_N}{\vdash}} (p, \varepsilon, \varepsilon),$$

wobei $q$ ein akzeptierender Zustand von $P_F$ ist.

Überdies sind alle Bewegungen zwischen den Konfigurationen $(q_0, w, Z_0X_0)$ und $(q, \varepsilon, \alpha X_0)$ Bewegungen von $P_F$. Insbesondere war $X_0$ vor dem Erreichen der Konfiguration $(q, \varepsilon, \alpha X_0)$ nie das oberste Stack-Symbol.[4] Somit schließen wir, dass dieselbe Berechnung in $P_F$ vorkommen kann, ohne dass sich $X_0$ auf dem Stack befindet; d. h. $(q_0, w, Z_0) \overset{*}{\underset{P_F}{\vdash}} (q, \varepsilon, \alpha)$. Daraus folgt, dass $P_F$ $w$ durch Endzustand akzeptiert, und daher ist $w$ in $L(P_F)$ enthalten. ■

4 Obwohl $\alpha$ auch $\varepsilon$ sein könnte; in diesem Fall hat $P_F$ seinen Stack geleert zu dem Zeitpunkt, in dem er akzeptiert.