Wir sprachen über lexikalische Analyse wollten eine Bedingung entwickeln, die das Funktionieren des einfachen Zerteilungsalgorithmus garantiert.

Sei $T = (T_1 \cup T_2 \cup \ldots \cup T_k)$. Für eine Sprache L bezeichnen wir die Menge der Präfixe der Worte in L mit $P(L)$ und mit $M(L) = L \setminus P(L)$ die Menge der Worte in L, die man nicht echt verlängern kann zu einem Wort in L. Falls L regulär ist, dann ist das auch $P(L)$ und $M(L)$.

An einer Eingabe u tut der Algorithmus das folgende. Er bestimmt einen maximalen Präfix t von u mit $t \in P(T)$, d.h. $t \in P(T)$ und entweder $u = t$ oder $u = tav$ und $ta \notin P(T)$. Falls $t \notin T$, dann behauptet der Algorithmus, dass u nicht zerlegbar ist. Andernfalls liefert er t als ersten Token ab und verfährt im zweiten Fall analog mit dem Restwort av.

Betrachte folgende Bedingungen (A) und (B):

Bedingung (A): falls $u \in T^* \cap P(T)$ dann $u \in T$

Bedingung (B): falls $u \in T^* \setminus P(T)$ und $u = tbv$ mit $t \in P(T)$ und $tb \notin P(T)$ dann ist $t \in T$ und $bv \in T^*$.

Lemma 1 *Wenn (A) und (B) gilt, dann findet der Algorithmus für jedes $u \in T^*$ eine Zerlegung.*

Lemma 2 *Man kann entscheiden, ob die Bedingungen gelten*

Wir beweisen zunächst, dass der Algorithmus funktioniert. Nehmen wir an, das sei nicht der Fall. Dann gibt es ein kürzestes Wort $u \in T^*$, für das unser Algorithmus keine Zerlegung findet. Wir haben entweder $u \in P(T)$ oder $u \notin P(T)$. Im ersten Fall ist $u \in T$ nach der ersten Bedingung. Der Algorithmus liest das Wort bis zum Ende und liefert einen einzigen Token ab, nämlich das u.

Im zweiten Fall sei $u = tbv$ mit $t \in P(T)$ und $tb \notin P(T)$. Nach der zweiten Behauptung ist dann $t \in T$ und $bv \in T^*$. Da bv kürzer ist als u, findet der Algorithmus eine Zerlegung für bv. Damit sind wir auch im zweiten Fall fertig.

Wir zeigen nun, wie man die Bedingungen testen kann. Für (A) ist das ganz einfach. Die Behauptung ist $(T^* \cap P(T)) \subseteq T$. T ist regulär. Also auch T^* und $P(T)$ also auch $T^* \cap P(T)$. Schließlich kann man für reguläre Sprachen testen, ob eine in der anderen enthalten ist.

Nun zur Behauptung (B). Es ist einfacher die Verneinung zu testen. Es gibt Worte u, t, v und Buchstaben b mit $u \in T^* \setminus P(T)$, $t \in P(T)$, $tb \notin P(T)$, $u = tbv$ und entweder $t \notin T$ oder $bv \notin T^*$. Oder anders ausgedrückt, die Sprachen

$$L = T^* \setminus P(T)$$
$$L_b = \{tbv\,;\, t \in P(T), tb \notin P(T) \text{ und entweder } t \notin T \text{ oder } bv \notin T^*\}$$

haben nicht-leeren Schnitt für irgendeinen Buchstaben $b \in \Sigma$. Die erste Sprache ist regulär. Wenn wir das auch von der zweiten zeigen sind wir fertig. Wir beobachten zunächst, dass

$$Q(T) = \{t\,;\, t \in P(T) \text{ and } tb \notin P(T)\}$$

regulär ist. Dann haben wir

$$
\begin{aligned}
L_b &= L'_b \cup L''_b \quad \text{wobei} \\
L'_b &= \{tbv\,;\, t \in Q(T) \text{ und } t \notin T\} \\
L''_b &= \{tbv\,;\, t \in Q(T) \text{ und } bv \notin T^*\}
\end{aligned}
$$

Also gilt

$$
\begin{aligned}
L'_b &= (Q(T) \setminus T) \cdot b \cdot \Sigma^* \\
L''_b &= Q(T) \cdot ((b \cdot \Sigma^*) \setminus T^*)
\end{aligned}
$$

und auch diese Sprachen sind regulär.