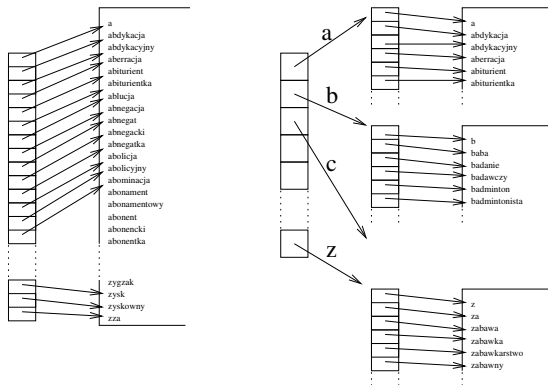


Słownik jako struktura danych



Pamięć: tyle co tekst + wektor

Czas: $\mathcal{O}(|w| \cdot \log n)$

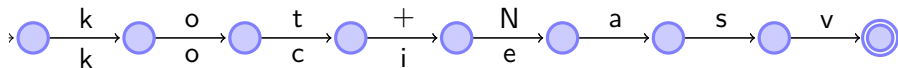
Pamięć: n bajtów mniej

Czas: $\mathcal{O}(|w| \cdot \log(n/|\Sigma|))$

Powtarzanie powoduje coraz mniejszą oszczędność, a w końcu straty pamięci, natomiast czas dostępu staje się proporcjonalny tylko do długości słowa. Wynik: drzewo literowe (ang. *trie*) – automat!

Analiza i synteza morfologiczna z użyciem automatów Mealy'ego

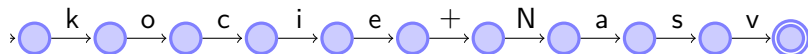
Automat Mealy'ego może służyć zarówno do analizy, jak i syntezy morfologicznej. Ścieżka w automacie zapewnia właściwą analizę/syntezę.



Chociaż automat jest odwracalny (analiza lub synteza), to jest dostosowany do użycia w jedną stronę za pomocą determinizacji.

Analiza i synteza morfologiczna z użyciem automatów bez wyjścia

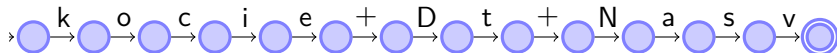
Dodatkowe informacje umieszczamy za formą odmienioną:



A co z formą podstawową? Nie można jej umieścić tak sobie, bo rozděłaby automat. Można za to:

kocie+Dt+Nasv

gdzie +Dt koduje formę podstawową. D oznacza „usuń trzy znaki z końca formy odmienionej” (A – zero, B – jeden,...).



Chyba mamy za mały słownik...

- **biegaj+Bć+V+imp+imper+sg+p2**
- **ślizgaj+Bć+V+imp+imper+sg+p2**
- **czołgaj+Bć+V+imp+imper+sg+p2**
- **łgaj+Bć+V+imp+imper+sg+p2**
- **błagaj+Bć+V+imp+imper+sg+p2**
- ...
- **pelgaj?**

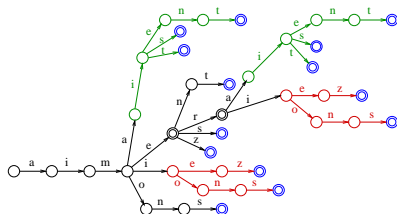
Chyba mamy za mały słownik...

- **biegaj**+Bć+V+imp+imper+sg+p2
- **ślizgaj**+Bć+V+imp+imper+sg+p2
- **czołgaj**+Bć+V+imp+imper+sg+p2
- **łgaj**+Bć+V+imp+imper+sg+p2
- **błagaj**+Bć+V+imp+imper+sg+p2
- ...
- **pelgaj**?

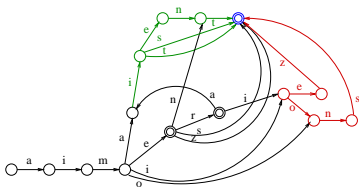
Chyba mamy za mały słownik...

- jag**eib**+Bć+V+imp+imper+sg+p2
- jag**zilś**+Bć+V+imp+imper+sg+p2
- jag**tozc**+Bć+V+imp+imper+sg+p2
- jag**ł**+Bć+V+imp+imper+sg+p2
- jag**ałb**+Bć+V+imp+imper+sg+p2
- ...
- jag**lep**?

Tradycyjne tworzenie minimalnego słownika



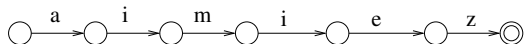
- Utworzenie drzewa



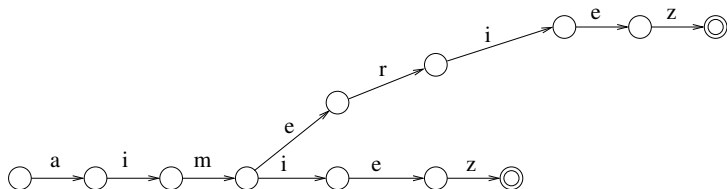
- Minimalizacja

Konieczność przechowywania ogromnego drzewa. Można lepiej!

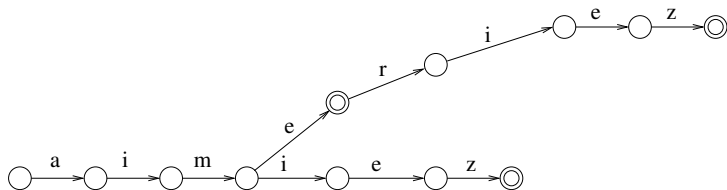
Tworzenie drzewa



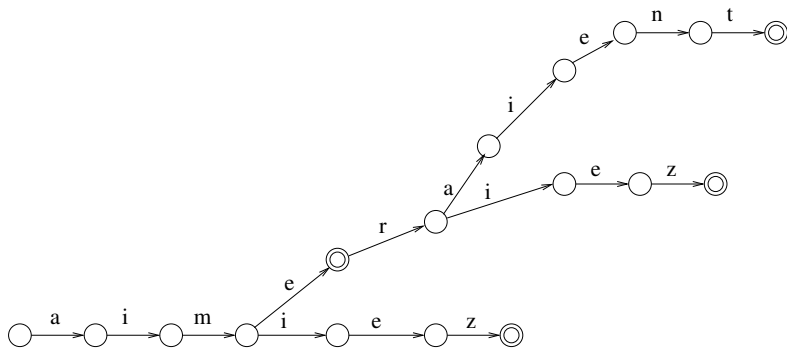
Tworzenie drzewa



Tworzenie drzewa



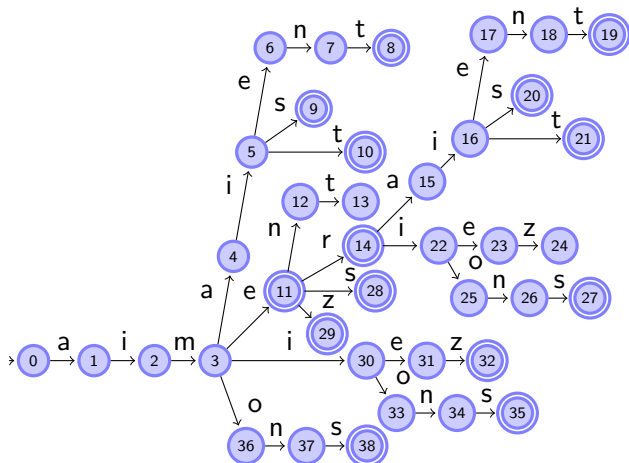
Tworzenie drzewa



Co to jest właściwie minimalizacja?

- $M = (Q, \Sigma, \delta, q_0, F)$, $|M| = |Q|$ – rozmiar automatu
 - Q – skończony zbiór stanów, $|Q|$ – liczba stanów
 - Σ – skończony zbiór symboli = alfabet
 - $\delta : Q \times \Sigma \rightarrow Q$ – funkcja przejścia
 - $q_0 \in Q$ – stan początkowy
 - $F \subseteq Q$ – zbiór stanów końcowych automatu
- M jest minimalny wtw. $\forall M' : \mathcal{L}(M') = \mathcal{L}(M) \quad |M| \leq |M'|$
- $\vec{\mathcal{L}}(q) = \{w : \delta^*(q, w) \in F\}$, $\mathcal{L}(M) = \vec{\mathcal{L}}(q_0)$
 - $\vec{\mathcal{L}}(q)$ – prawostronny język stanu q , $\mathcal{L}(M)$ – język automatu
 - $\delta^* : Q \times \Sigma^* \rightarrow Q$ – rozszerzona funkcja przejścia
- $p \equiv q$ wtw. $\vec{\mathcal{L}}(p) = \vec{\mathcal{L}}(q)$.
- M jest minimalny wtw. $\forall p, q \in Q \quad p \equiv q \Leftrightarrow p = q$
- $\vec{\mathcal{L}}(q) = \bigcup_{a: \delta(q, a) \neq \perp} a \vec{\mathcal{L}}(\delta(q, a)) \cup \begin{cases} \emptyset & q \notin F \\ \{\epsilon\} & q \in F \end{cases}$

Prawostronny język stanu



$$\begin{aligned} \vec{\mathcal{L}}(14) &= \{\epsilon\} \cup a\vec{\mathcal{L}}(15) \cup i\vec{\mathcal{L}}(22) = \{\epsilon\} \cup ai\vec{\mathcal{L}}(16) \cup i(e\vec{\mathcal{L}}(23) \cup o\vec{\mathcal{L}}(25)) = \dots \\ \vec{\mathcal{L}}(14) &= \{\epsilon, aient, ais, ait, iez, ions\} \end{aligned}$$