Algorytm Hopcrofta minimalizacji automatu

Przedstawimy teraz algorytm minimalizacji skończonego automatu deterministycznego, zaproponowany przez Johna E. Hopcrofta[1].

Niech $A = (\Sigma, Q, q_I, \delta, F)$ będzie automatem deterministycznym. Przypomnijmy relację „sklejania stanów", $\sim \subseteq Q \times Q$

$$q \sim q' \Leftrightarrow L(A, q) = L(A, q'),$$

gdzie $L(A, q) = \{w \, : \, (\exists p \in F) \, q \xrightarrow{w} p\}$.

Relację $\sim$ znajdziemy poprzez obliczenie podziału, jaki ta relacja równoważności wyznacza na zbiorze Q. Podział ten będzie znaleziony "metodą kolejnych przybliżeń", przy czym każdy kolejny podział będzie rozdrobnieniem poprzedniego.

Niech $X, Y \subseteq Q$, $a \in \Sigma$. Określamy $X' = \{q \in X \, : \, \delta(q, a) \in Y\}$, $X'' = \{q \in X \, : \, \delta(q, a) \notin Y\}$. Oczywiście, $X' \cup X'' = X$, $X' \cap X'' = \emptyset$. Jeżeli oba zbiory X', X'' są niepuste, to powiemy, że para (Y, a) *łamie* zbiór X, a parę $\{X', X''\}$ nazwiemy *złamaniem* zbioru X przez (Y, a). Jeżeli P jest podziałem zbioru Q, to *złamaniem* P przez (Y, a) nazywamy podział otrzymany z podziału P przez zastąpienie każdego $X \in P$, który jest łamany przez (Y, a), przez jego złamanie X', X''.

W algorytmie będziemy używać dwóch typów struktur: z jednej strony, podziału (zmienna P), a z drugiej, listy par (Y, a) przeznaczonych do łamania (zmienna $\mathcal{L}$). Przyjmujemy oznaczenie $\bar{X} = Q - X$.

procedura **Hopcroft**
$P := \{F, \bar{F}\}$;
jeśli $|F| < |\bar{F}|$ **to** $\mathcal{L} := ((F, \sigma) : \sigma \in \Sigma)$
w przeciwnym przypadku $\mathcal{L} := ((\bar{F}, \sigma) : \sigma \in \Sigma)$
dopóki $\mathcal{L} \neq \emptyset$ **wykonuj**

1. zdejmij parę (Y, a) z listy $\mathcal{L}$;
2. oblicz zbiór $\mathcal{B}$ tych $X \in P$, że para (Y, a) łamie X;
3. **dla** $X \in \mathcal{B}$, $b \in \Sigma$ **wykonuj**
   (a) oblicz złamanie $\{X', X''\}$ przez (Y, a);
   (b) **jeśli** $(X, b) \in \mathcal{L}$ **to**
   zastąp (X, b) w $\mathcal{L}$ przez $(X', b), (X'', b)$;
   (c) **w przeciwnym przypadku**
   jeśli $|X'| < |X''|$ **to** dołącz (X', b) do $\mathcal{L}$
   w przeciwnym przypadku dołącz (X'', b) do $\mathcal{L}$.

W poniższej analizie zostanie wykazane, że algorytm działa poprawnie, tzn. zawsze zatrzymuje się i końcową wartością zmiennej P jest podział wyznaczony przez relację $\sim$. Ponadto, jego czas działania jest $\mathcal{O}(|\Sigma| n \log n)$.

[1] Patrz także Notatki, Odcinek 2, str. 12.

Analiza algorytmu Hopcrofta minimalizacji deterministycznego automatu skończonego

Paweł Parys (parys@mimuw.edu.pl)

24 listopada 2005

Algorytm się zatrzyma, bo każda para (X, a) może pojawić się na liście $\mathcal{L}$ tylko raz — wtedy, gdy zbiór X pojawia się w podziale P. A to dzieje się tylko raz, bo podział P cały czas się rozdrabnia.

Oznaczmy podziały uzyskiwane w kolejnych przebiegach pętli przez $P_0, P_1, \dots, P_s$ (przy czym $P_0 = \{F, \bar{F}\}$), natomiast odpowiadające im relacje równoważności przez $\sim_0, \sim_1, \dots, \sim_s$. Najpierw udowodnię poprawność otrzymywanego podziału, czyli że dla dowolnych $q, q' \in Q$ mamy $q \sim_s q' \Leftrightarrow L(A, q) = L(A, q')$. Wynika ona bezpośrednio z lematów 1 i 2

Lemat 1 *Jeśli dla pewnych $q, q' \in Q, k \in \{0, \dots, s\}$ mamy $q \not\sim_k q'$, to $L(A, q) \neq L(A, q')$.*

Dowód Udowodnię to przez indukcję ze względu na k. Dla $k = 0$ jeśli $q \not\sim_k q'$, to $q \in F$, $q' \notin F$ (lub odwrotnie), czyli $\epsilon \in L(A, q)$, ale $\epsilon \notin L(A, q')$ (lub odwrotnie). Załóżmy teraz, że teza indukcyjna jest prawdziwa dla pewnego k oraz że $q \not\sim_{k+1} q'$. Jeśli ponadto $q \not\sim_k q'$, to z założenia indukcyjnego dostajemy od razu, że $L(A, q) \neq L(A, q')$. Załóżmy więc, że $q \sim_k q'$. Oznacza to, że stany q i q' należały do pewnego wspólnego $X \in P_k$ i zostały rozdzielone w wyniku złamania tego X przy $k+1$-szym przebiegu pętli. Istnieją więc $a \in \Sigma$, $Y \in P_k$ takie, że $\delta(q, a) \in Y$ i $\delta(q', a) \notin Y$ (lub odwrotnie), czyli że $\delta(q, a) \not\sim_k \delta(q', a)$ (trzeba tu zauważyć, że jeśli para (Y, a) jest na liście $\mathcal{L}$, to na pewno Y należy do aktualnego podziału P). Z założenia indukcyjnego dostajemy, że $L(A, \delta(q, a)) \neq L(A, \delta(q', a))$. Istnieje więc słowo w, które należy tylko do jednego z tych dwóch języków. Wówczas słowo aw należy tylko do jednego z języków $L(A, q)$, $L(A, q')$, czyli $L(A, q) \neq L(A, q')$.

Lemat 2 *Jeśli dla pewnych $q, q' \in Q$ mamy $q \sim_s q'$, to $L(A, q) = L(A, q')$.*

Dowód Inaczej mówiąc chcemy dowieść, że dla dowolnego słowa $w \in \Sigma^*$ i dowolnych stanów $q, q' \in Q$ jeśli $w \in L(A, q)$ i $w \notin L(A, q')$, to $q \not\sim_s q'$. Udowodnię to przez indukcję ze względu na długość słowa w. Dla $w = \epsilon$ z $w \in L(A, q)$ wynika, że $q \in F$, natomiast z $w \notin L(A, q')$ wynika, że $q' \notin F$. Zatem na pewno $q \not\sim_0 q'$, czyli tym bardziej $q \not\sim_s q'$. Załóżmy, że dla wszystkich słów pewnej długości teza jest spełniona. Niech aw ($a \in \Sigma$, $w \in \Sigma^*$) będzie dowolnym słowem długości o jeden większej. Załóżmy, że dla pewnych

$q, q' \in Q$ mamy $aw \in L(A, q)$ i $aw \notin L(A, q')$. Wynika z tego, że $w \in L(A, \delta(q, a))$ i $w \notin L(A, \delta(q', a))$. Z założenia indukcyjnego dostajemy, że $\delta(q, a) \not\sim_s \delta(q', a)$. Zauważmy, że w pewnym momencie musieliśmy wstawić na listę $\mathcal{L}$ parę (X, a), taką, że dokładnie jeden ze stanów $\delta(q, a)$ i $\delta(q', a)$ należy do X (albo jest to F lub $\bar{F}$, albo stany te zostały rozdzielone podczas pewnego łamania, wówczas X' lub X'' jest takim zbiorem). Podczas pobytu na liście $\mathcal{L}$ zbiór X być może został złamany na dwa podzbiory X', X''. Jednak któryś z X', X'' ma nadal tą własność, że należy do niego dokładnie jeden ze stanów $\delta(q, a)$ i $\delta(q', a)$. Wyjmiemy więc w końcu parę (X, a), która ma tą własność. Jeśli w tym momencie q, q' należą jeszcze do tego samego elementu podziału, to element ten zostanie złamany przez (X, a), przy czym q i q' znajdą się w różnych częściach (wynika to wprost z definicji łamania). Zatem $q \not\sim_s q'$, co kończy dowód indukcyjny.

Sprecyzuję teraz jakie struktury danych są używane w programie i jak on dokładnie działa — jest to konieczne dla obliczenia jego złożoności. Utożsamiamy stany z liczbami naturalnymi (od 0 do $n-1$). Dla każdego stanu $q \in Q$ i każdej litery $a \in \Sigma$ pamiętamy listę stanów $p \in Q$ takich, że $\delta(p, a) = q$. Elementy podziału P będziemy numerować liczbami naturalnymi, od 0 do $|P| - 1$. Będą one pamiętane jako listy dwukierunkowe (powiedzmy, że będą to listy cykliczne). Konkretnie, dla każdego stanu pamiętamy numery poprzedniego i następnego stanu na liście odpowiadającej jego elementowi podziału, a także numer elementu podziału. Natomiast dla każdego numeru elementu podziału pamiętamy reprezentanta (czyli numer dowolnego ze stanów w tym elemencie) oraz liczbę stanów w tym elemencie. Na liście $\mathcal{L}$ będziemy trzymać pary (numer elementu, litera). Inicjalizację można przeprowadzić w czasie $O(|\Sigma|n)$.

Lemat 3 *Kroki 1-3 można zaimplementować tak, aby dla pary (Y, a) działały w czasie $O(|\{p \in Q \colon \delta(p, a) \in Y\}| + |Y| + \Delta\mathcal{L})$, gdzie przez $\Delta\mathcal{L}$ rozumiem liczbę par wstawionych na listę $\mathcal{L}$.*

Dowód Aby to uzyskać nie możemy przeglądać wszystkich elementów łamanego zbioru X. Postępujemy w następujący sposób (dla każdego X takiego, że $X' = \{p \in X \colon \delta(p, a) \in Y\} \neq \phi$). Na początku zliczamy ile stanów należy do X'. Jeśli $|X'| = |X|$, to nic nie trzeba już robić. Jeśli $|X'| < \frac{1}{2}|X|$, to usuwamy ze zbioru X elementy X' (trzeba je usunąć z listy dwukierunkowej i ewentualnie zmienić reprezentanta X) i tworzymy zbiór X' pod nowym numerem. Wstawiamy też (X', b) na listę $\mathcal{L}$ dla wszystkich $b \in \Sigma$. Nie trzeba odnajdować zbioru X na liście $\mathcal{L}$ — jeśli gdzieś tam występuje, to ma pozostać (ze zmienioną zawartością). Robiąc to przeglądamy tylko $O(|X'|)$ stanów. W przeciwnym wypadku ($\frac{1}{2}|X| \leqslant |X'| < |X|$) dla X' pozostawiamy dotychczasowy numer zbioru X, natomiast dla $X \setminus X'$ nadajemy nowy numer. Dodajemy $(X \setminus X', b)$ na listę $\mathcal{L}$ (dla każdego $b \in \Sigma$). Robiąc to przeglądamy $O(|X|) = O(|X'|)$ stanów. Umiemy więc dla konkretnego X wykonać $O(|X'| + \Delta\mathcal{L})$ operacji. Tyle że dla danego Y mamy dużo różnych X, które trzeba rozważać. Nie jest to jednak problemem. Wszystko to wykonujemy jednocześnie dla wszystkich X (takich, że $X' \neq \phi$) przechodząc listę Y kilka (stałą liczbę) razy — przy każdym przejściu przez listę wykonujemy mały krok powyższego algorytmu. Tymczasowe wyniki możemy trzymać w tablicach indeksowanych od 0 do $n-1$, ale z których używamy jedynie pól odpowiadającym stanom należącym do Y lub elementom podziału X, dla których $X' \neq \phi$. Będzie więc to wszystko (tzn. łamanie całego podziału parą (Y, a)) działało w czasie $O(|\{p \in Q \colon \delta(p, a) \in Y\}| + |Y| + \Delta\mathcal{L})$.

Przez $\widetilde{P}$ oznaczmy zbiór wszystkich podzbiorów Q, które kiedykolwiek pojawiły się w podziale P. Dla elementów $\widetilde{P}$ zdefiniuję rząd (oznaczany $r(Y)$). Definicja będzie indukcyjna (określamy rząd dla elementów powstających w kolejnych krokach algorytmu):

- $r(F) = r(\bar{F}) = 0$
- Jeśli w wyniku łamania pewnego X powstały niepuste zbiory X', X'', to jeśli $|X'| < \frac{1}{2}|X|$, to $r(X') = r(X) + 1$, $r(X'') = r(X)$, jeśli zaś $|X'| \geqslant \frac{1}{2}|X|$, to $r(X') = r(X)$, $r(X'') = r(X) + 1$.

Lemat 4 *Rząd każdegu elementu z $\widetilde{P}$ jest nie większy niż $\log_2 n$.*

Dowód Zauważmy, że aby zbiór miał pewien rząd $k+1 > 0$, to musi istnieć w $\widetilde{P}$ zbiór o rzędzie k i co najmniej dwa razy większej mocy. Ponieważ w $\widetilde{P}$ mamy zbiory o mocy co najwyżej n, to rząd każdego z nich musi być nie większy niż $\log_2 n$.

Ustalmy pewną literę $a \in \Sigma$. Niech $\mathcal{L}_a$ będzie zbiorem takich $Y \in \widetilde{P}$, że para (Y, a) została wyjęta z listy $\mathcal{L}$.

Lemat 5 *Jeśli $X, Y \in \mathcal{L}_a$ i $r(X) = r(Y)$, to $X \cap Y = \phi$ lub $X = Y$.*

Dowód Zauważmy najpierw, że dla dowolnych $X, Y \in \widetilde{P}$ jeśli $X \cap Y \neq \phi$, to $X \subset Y$ lub $Y \subset X$. Załóżmy więc, że $X, Y \in \mathcal{L}_a$, $X \neq Y$, $X \subset Y$, $r(X) = r(Y)$. Istnieje wtedy ciąg zbiorów $Y = X_0 \supset X_1 \supset \ldots \supset X_k = X$, przy czym zbiór X_i powstał w wyniku łamania zbioru X_{i-1}. Ponadto rząd każdego z tych zbiorów jest taki sam. Teraz udowodnię indukcyjnie po i, że w momencie łamiania zbioru X_i para (X_i, a) nie znajduje się na liście $\mathcal{L}$. Zauważmy, że w momencie łamania zbioru $X_0 = Y$ para (Y, a) nie znajduje się już na liście $\mathcal{L}$ (bo inaczej ta para zostanie zastąpiona innymi, czyli $Y \notin \mathcal{L}_a$). Załóżmy, że teza indukcyjna jest prawdziwa dla pewnego i. Wówczas podczas łamania zbioru X_i dla literki a wykona się podpunkt (c) algorytmu, tzn. tylko jeden z powstałych podzbiorów zostanie wstawiony na listę z literką a. Będzie to jednak ten, którego rząd jest równy $r(X_i)+1$, czyli na liście $\mathcal{L}$ nie znajdzie się para (X_{i+1}, a). To daje tezę indukcyjną dla $i+1$. Dla $i+1 = k$ dostajemy sprzeczność, bo $X_k = X \in \mathcal{L}_a$.

Lemat 6 *Algorytm działa w czasie $O(|\Sigma| n \log n)$*

Dowód Jak już powiedziałem wcześniej, inicjalizacja działa w czasie $O(|\Sigma|n)$. Trzeba udowodnić, że suma po wszystkich parach (Y, a) wyjmowanych z listy $\mathcal{L}$ czasu działania kroków 1-3 algorytmu wynosi $O(|\Sigma| n \log n)$. Jak wiemy z lematu 3 ten czas dla konkretnej pary (Y, a) wynosi $O(|\{p \in Q\colon \delta(p,a) \in Y\}| + |Y| + \Delta\mathcal{L})$, pozostaje zsumować to po wszystkich parach.

Zauważmy najpierw, że suma $|Y|$ po wszystkich $Y \in \mathcal{L}_a$ (dla konkretnego $a \in \Sigma$) wynosi $O(n \log n)$, ponieważ mamy najwyżej $\log_2 n$ rzędów, a zbiory danego rzędu są rozłączne. Zatem razem (tzn. dla wszystkich $a \in \Sigma$) będzie to $O(|\Sigma| n \log n)$.

Przyjrzyjmy się teraz sładnikom $\Delta\mathcal{L}$. Jest to liczba elementów, które przez cały czas wstawiliśmy na listę $\mathcal{L}$. Oczywiście elementów, które wstawiliśmy i które wyjęliśmy jest tyle samo. Jest to więc suma $|\mathcal{L}_a|$ po wszystkich $a \in \Sigma$. Natomiast w $|\mathcal{L}_a|$ elementy każdego rzędu są niepuste i rozłączne (jest ich więc najwyżej n), a rzędów jest tylko $\log_2 n$. Razem jest to więc $O(|\Sigma| n \log n)$.

Pozostaje zsumować po wszystkich (Y, a) wyjętych z listy $\mathcal{L}$ wartość $|\{p \in Q\colon \delta(p,a) \in Y\}|$. Znowu postępujemy tak samo. Patrzymy na konkretne $a \in \Sigma$ oraz na $Y \in \mathcal{L}_a$ ustalonego rzędu. Ponieważ Y są rozłączne, to także $\{p \in Q\colon \delta(p,a) \in Y\}$ są rozłączne. Zatem suma ich mocy wynosi $O(n)$, co łącznie daje $O(|\Sigma| n \log n)$. To kończy dowód.