

magisterka

michalfica125

August 2024

1 JAKI JEST PLAN?

1. Znaleźć przyczyny dlaczego mój kod przestał działać.
2. Rozszerzyć oprogramowanie tak bym mógł symulować uczenie się automatów z nowymi wiedzami.
3. Musze zrewidować mój początkowy pomysł - uczenie się częściowych maszyn mealy'go.

2 Wstęp

2.1 Uczenie się autmatów

Wykorzystuje w tej pracy klasyczny algorytm uczenia się automatów L^* . Niech M_A określa automat, którego algorytm ma za zadanie się nauczyć. Algorytm L^* można dostosować tak, aby służył do bezpośredniego uczenia się maszyn Mealego.

2.1.1 Zwiezły opis algorytmu L^*

Algorytm zaczyna prace od automatu pustego, w trakcie swojego działania rozbudowuje autmat, a na końcu zwraca automat równoważny docelowej maszynie M_A . Podczas swojego działania algorytm zadaje zapytania o należenie słów do języka docelowego, na które dostaje odpowiedź tak lub nie oraz formułuje hipotezy i sprawdza czy zbudowany do tej pory automat jest poprawny (równoważny M_A). Na zapytania drugiego rodzaju możliwa odpowiedź jest tak - wtedy algorytm kończy działanie. W przeciwnym przypadku algorytm dostaje kontrprzykład czyli słowo $w \in L(M_A)$, które nie zostało zaakceptowane przez sformułowaną hipotezę.

2.2 Uczenie się automatów z wiezami

Wiezy są to dodatkowe założenia o języku, dające pewne gwarancje algorytmowi. Algorytm może wykorzystać wiedzę wynikającą z wiezów w taki sposób, aby zmniejszyć liczbę zadawanych zapytań do wyroczni.

2.3 Przykłady wiezów

- System przepisywania słów
- Częściowa maszyna mealy'go

3 Uwagi, spostrzeżenia na temat L^*

Lemma 3.1. *Liczba wierszy w tabelce, po poprawnym zakończeniu się algorytmu, jest równa $n \cdot i + 1$.*

Proof. Wiersze w tabelce można podzielić na dwie grupy Q - wiersze reprezentujące stany oraz P - wiersze reprezentujące przejścia. Początkowo

$$Q = \{\epsilon\}, P = \{a | a \in |I|\}.$$

Aby dodać wiersz do Q wybieram 1 wiersz $q \in P$ i przenoszę go do Q . Następnie, zgodnie z tym jak działa algorytm, do grupy przejść trafia $|I|$ nowych słów. Żadne z nowo dodanych słów do P nie występowało wcześniej zarówno w Q jak i w P . W przeciwnym razie, z powodu domknietości prefiksowej wierszy, słowo q musiało by trafić już wcześniej do Q . Z kolei każdy wiersz może trafić do Q tylko 1 raz (dowód: weźmy najkrótszy, który trafił 2 razy i z domknietości prefiksowej musi istnieć krótszy - \perp).

W chwili zakończenia działania algorytmu

$$|Q| = n, |P| = (n - 1) \cdot (i - 1),$$

wiec liczba wierszy równa $|Q| + |P|$ wynosi dokładnie $n \cdot i + 1$. \square

Wiersze tabeli są domknięte prefiksowo, kolumny sufiksowo. Poniżej uwagi dlaczego ta domknietność jest przydatna (według mnie).

- Domknietość **prefiksowa** - wygodnie i łatwo daje się wtedy pokazać, że hipoteza jest zgodna z tabelką obserwacji.
- Domknietość **sufiksowa** - przy uzupełnianiu przejść, mogą wyodrębnić się kolejne nowe stany - dla nich sufiks słowa rozróżniającego będzie słowem rozróżniającym, w przeciwnym wypadku (gdyby dodawać tylko słowo rozróżniające a nie wszystkie jego sufiksy) hipoteza nadal może nie działać na tym samym kontrprzykładzie.

4 System przepisywania słów

4.1 Opis wiezów

Ograniczam się do uczenia się automatów *DFA*. Wiezy zadaje system przepisywania słów. Dane są reguły przepisywania słów

$$\Gamma : u_i \rightarrow v_i,$$

gdzie

$$u_i, v_i \in (\Sigma \cdot X)^*, |u_i| \geq |v_i|$$

Np. $XabY \rightarrow XbaY$. Zakładamy, że język DFA $L(A)$ spełnia warunek:

$$\forall w, v (w \rightarrow_{\Gamma}^* v) \Rightarrow (w \in L(A) \leftrightarrow v \in L(A))$$

Dzięki temu można sprawdzić czy dane słowo należy do języka wyprowadzając je ze słowa, o którym już wiemy, że jest w języku. Dodatkowo można wyprowadzić wiele kontrprzykładów z jednego kontrprzykładu (na tyle różnych na ile pozwalają reguły przepisywania).

5 Cześciowa maszyna mealy'go

5.1 Opis wiezów

Ograniczam się tylko do maszyn Mealy'go, które rozpoznają wzorce. Załóżmy, że docelowa maszyna ma rozpoznawać kilka wzorców. Pomysł jest taki aby najpierw nauczyć się maszyny, która rozpoznaje tylko jeden wzorec i następnie rozbudować ją aby rozpoznawała kolejny wzorec. Uczyć się "iteracyjne". Problem polegał na tym, że tak czy siak zadawałem zapytania do wyroczni o to samo słowo. To podejście muszę **zrewidować**.

5.2 Cel

Niech:

- q_w - liczba zapytań o różne słowa zawierające nowy wzorec,
- q_{all} - liczba wszystkich "nowych" zapytań.

Chciałbym wiedzieć ile zapytań jestem w stanie zaoszczędzić korzystając z wyroczni. Do tego potrzebuje znać:

- $q_{all} \geq ?$ - ograniczenie dolne,
- $? \geq q_w$ - ograniczenie górne.

5.3 Wnioski (do tej pory)

- Zapytania biorą się ze sklejania 2 słów (wiersz kolumna) takich, że wiersz kończy się prefiksem wzorca, a kolumna zaczyna odpowiednim sufiksem wzorca.
- $q_w \geq |E| + 1$ (tyle zapytań bierze się z wiersza przejść, który zawiera pełny wzorec).
- Nowo powstałe stany (każdy z nich) też jest odpowiedzialny za conajmniej jedno zapytanie o słowo z nowym wzorcem.

5.4 Spostrzeżenia (do zweryfikowania)

- **Odwzorowanie między tabelką a maszyną:**
 - wiersze na górze tabelki - stany,
 - wiersze na dole tabelki - przejścia wychodzące ze stanów (bez "najkrótszego" / "pierwszego" wejścia do każdego stanu),
 - **kolumny** - dla każdej pary najkrótsze słowo zawierające jedynekę?
- Krótkie słowa lepiej uczyć się na początku niż na końcu.
- Nie jest tak, że zapytania o nowy wzorec pojawiają się dopiero po otrzymaniu kontrprzykładu z tym wzorcem.

5.5 Pytania

- (DLACZEGO?) Zapytań o nowe słowa jest dużo a zapytań o nowe słowa, w których występuje nowy wzorzec jest względnie mało.
- Czy da się to powiązać z różnicą w przejściach między dwoma automatami - **wyrocznia i docelowym**?
- Przykłady **wymiany pod słów z opóźnieniem**.
- Przykłady gdy **wiecej wzorców - mniej stanów w maszynie**.

5.6 Wzorce 2 literowe

Podczas uczenia się wzorców 2-literowych nie potrzeba kontrprzykładów, pierwsza hipoteza, która algorytm formułuje jest już dobra.

5.7 Tabelka - wyszukiwanie wzorców (już nie aktualne)

Maszyna Mealy'ego			Wiezy	ilość zapytań bez wyroczni	ilość zapytań z wyrocznia	uwagi do wiezów
Opis	Q	Σ				
<i>a</i>	1	3	dfa, słowa z literka <i>a</i>	12, 1	6, 1	akcept tylko słowa, w których występuje <i>a</i>
<i>a, b</i>	1	3	dfa, słowa z literka <i>a</i> i bez <i>b</i>	12, 1	8, 1	akcept tylko słowa, w których występuje <i>a</i>
<i>a, b</i>	1	3	dfa, słowa z literka <i>a</i> i/lub <i>b</i>	12, 1	2, 1	akcept tylko słowa, w których jest <i>a</i> lub jest <i>b</i>
<i>aa, bb, cc</i>	4	3	dfa, słowa zawierające <i>aa</i>	39, 1	33, 1	-
<i>aaa, bbb, ccc</i>	7	3	-	111, 4	-	-
<i>AT, CG</i>	3	4	-	52, 1	-	-
<i>alice, bob</i>	7	26	-	5462, 3	-	-
<i>alice</i>	5	7	-	345, 2	-	-
<i>bob</i>	3	7	-	173, 2	-	-
<i>alice, bob</i>	7	7	maszyna mealy'ego rozpoznająca <i>bob</i>	522, 3	18, 3	-
<i>alice, bob</i>	7	7	maszyna mealy'ego rozpoznająca <i>alice</i>	522 , 3	17 , 3	-
<i>bob</i>	3	4	-	18, 2	-	-
<i>boobie</i>	6	4	-	176, 2	-	-
<i>bob, boobie</i>	6	4	maszyna mealy'go rozpoznająca <i>bob</i>	157 , 3	8 , 3	-
<i>bob, boobie</i>	6	4	maszyna mealy'go rozpoznająca <i>boobie</i>	157 , 3	13 , 3	-
<i>abba</i>	4	4	-	94, 2	-	-
<i>abbc</i>	4	4	-	94, 2	-	-
<i>abba, abbc</i>	4	4	maszyna mealy'ego rozpoznająca <i>abba</i> 7	94, 2	7, 2	-
<i>abba, abbc</i>	4	4	maszyna mealy'ego rozpoznająca <i>abbc</i>	94, 2	9, 2	-

Maszyna Mealy'ego			Wiezy	ilość zapytań bez wyroczeni	ilość zapytań z wyrocznia	uwagi do wiezów
Opis	Q	Σ				
<i>abba, abbc, abbd</i>	4	4	maszyna mealy'ego rozpoznajaca <i>abba</i>	94, 2	14, 2	-
<i>abba, abbc, abbd</i>	4	4	maszyna mealy'ego rozpoznajaca <i>abba</i> i <i>abbc</i>	94, 2	7, 2	-
<i>abcd</i>	5	4	-	132, 2	-	-
<i>dcd</i>	3	4	-	62, 2	-	-
<i>abcd, dcd</i>	5	4	maszyna mealy'ego rozpoznajaca <i>dcd</i>	204, 3	14, 3	-
<i>abcd, dcd</i>	5	4	maszyna mealy'ego rozpoznajaca <i>abcd</i>	204, 3	19, 3	-
<i>abcd, dcd, bdb</i>	9	4	maszyna mealy'ego rozpoznajaca <i>dcd</i>	288, 4	35, 4	-
<i>abcd, dcd, bdb</i>	9	4	maszyna mealy'ego rozpoznajaca <i>abcd</i>	288, 4	44, 4	-
<i>abcd, dcd, bdb</i>	9	4	maszyna mealy'ego rozpoznajaca <i>bdb</i>	288, 4	43, 4	-
<i>abcd, dcd, bdb</i>	9	4	maszyna mealy'ego rozpoznajaca <i>dcd</i> i <i>bdb</i>	288, 4	17, 4	-
<i>abcd, dcd, bdb</i>	9	4	maszyna mealy'ego rozpoznajaca <i>dcd</i> i <i>abcd</i>	288, 4	18, 4	-
<i>bla, blabla</i>	7	4	maszyna mealy'ego rozpoznajaca <i>bla</i>	204, 3	10, 3	-

Maszyna Mealy'ego			Wiezy	ilość zapytań bez wyroczni	ilość zapytań z wyrocznia	uwagi do wiezów
Opis	Q	Σ				
<i>dcd</i>	3	4	-	62, 2	-	-
<i>dcd, bdb</i>	5	4	maszyna mealy'ego rozpoznająca <i>dcd</i>	116, 3	10, 3	-