دانشکده‌ی علوم ریاضی

نظریه‌ی زبان‌ها و اتوماتا ۲ آبان ۱۳۹۱

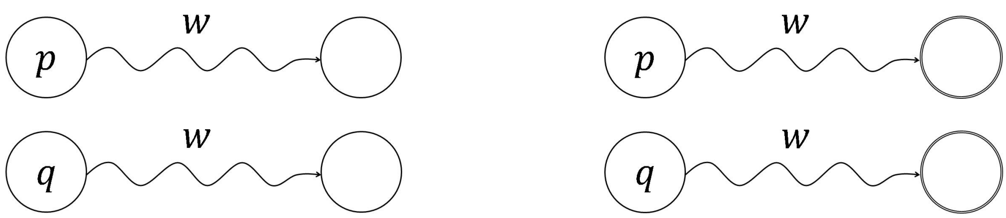
جلسه‌ی ۱۲: بدست آوردن DFA کمینه با الگوریتم پرکردن جدول، مفهوم برابری در DFAها

مدرّس: دکتر شهرام خزائی *نگارنده: سهی صالحیان قمصری*

۱ تعاریف

تعریف ۱ *فرض کنید $A = (Q, \Sigma, \delta, q_0, F)$ یک DFA باشد، حالت‌های $p, q \in Q$ را تمایزناپذیر[۱] یا معادل[۲] می‌گوییم و با $p \equiv q$ نشان می‌دهیم، اگر به ازای هر دنباله‌ی w ، هر دوی $\hat{\delta}(p, w)$ و $\hat{\delta}(q, w)$ حالت نهایی باشند یا هیچ یک نهایی نباشند. به عبارتی:*

$$p \equiv q \Leftrightarrow (\forall w \in \Sigma^* : \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F)$$

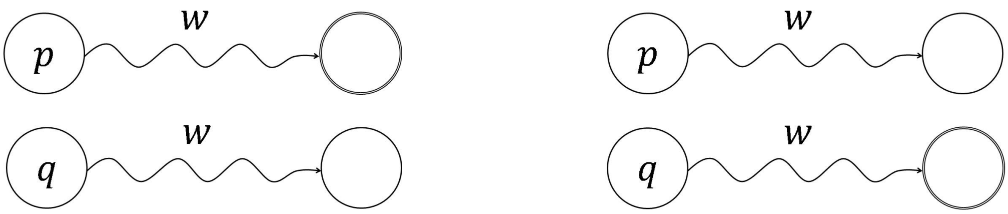


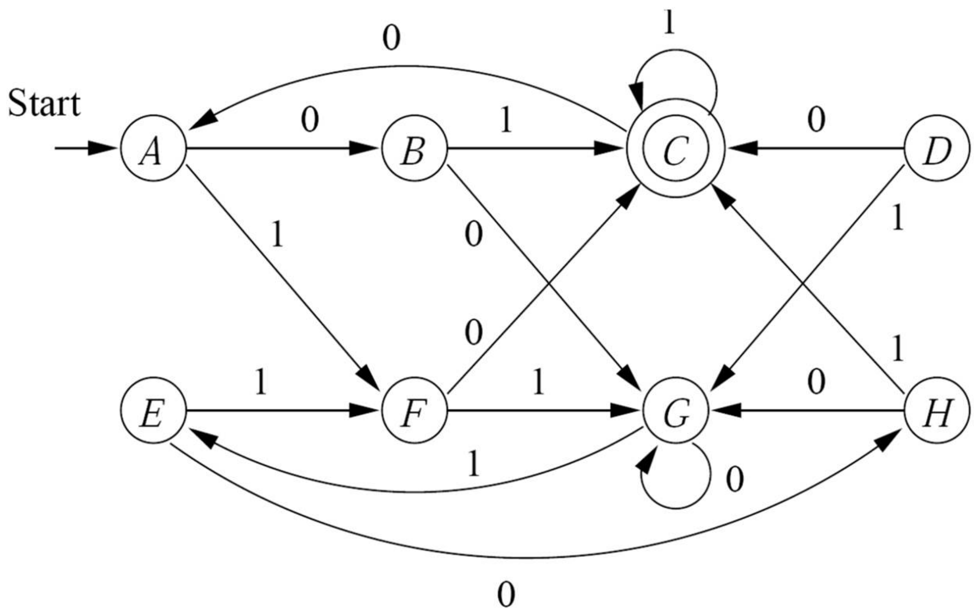
تعریف ۲ *اگر p, q معادل نباشند، می‌گوییم p, q تمایزپذیرند[۳] و با $p \not\equiv q$ نشان می‌دهیم. معادلاً p, q تمایزپذیرند، اگر رشته‌ی w ای وجود داشته باشد که از دو حالت $\hat{\delta}(p, w)$ و $\hat{\delta}(q, w)$ فقط یکی نهایی باشد. در این صورت می‌گوییم رشته‌ی w ، حالت های p, q را تمایز می‌دهد.*

۲ به دست آوردن حالت های معادل در یک DFA

DFA زیر را در نظر بگیرید:

[۱] indistinguishable
[۲] equivalent
[۳] distinguishable





در DFA داده شده ادعا می‌کنیم که $C \not\equiv G$ می‌باشد. کافیست حداقل یک رشته w پیدا کنیم که از دو حالت $\hat{\delta}(C,w)$ و $\hat{\delta}(G,w)$ فقط یکی نهایی باشد.
رشته $w=\varepsilon$ دارای این ویژگی است که به این معناست که حالت‌های C و G را تمایز می‌دهد:

$$\hat{\delta}(C,\varepsilon) \in F \wedge \hat{\delta}(G,\varepsilon) \notin F \Rightarrow G \not\equiv C$$

برای اثبات غیرمعادل بودن دو حالت کافی است یک رشته بیابیم که دو حالت را تمایز بدهد؛ مثلاً در شکل مثال ۱، A و G معادل نیستند، چون به ازای $w=۰۱$ داریم:

$$\hat{\delta}(A,۰۱) \in F$$

اما

$$\hat{\delta}(G,۰۱) \notin F$$

لم ۱ *یک حالت نهایی از یک حالت غیرنهایی تمایزپذیر است.*

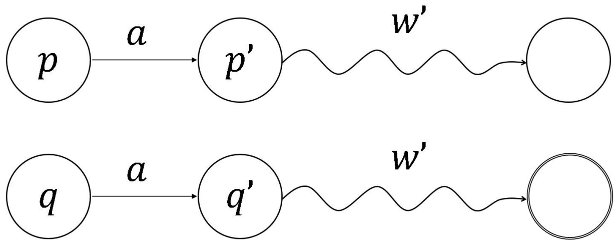
$$f \in F \wedge q \notin F \Rightarrow f \not\equiv q$$

برهان. رشته ی ε دو حالت را تمایز می دهد. ■

لم ۲ *اگر حالت های p, q معادل باشند، به ازای هر حرف a حالت های $\delta(p, a)$ و $\delta(q, a)$ نیز معادلند. به زبان ریاضیات داریم:*

$$p \equiv q \Rightarrow \forall a \in \Sigma : \delta(p, a) \equiv \delta(q, a)$$

برهان. با برهان خلف حکم را ثابت می‌کنیم. اگر $p' = \delta(p, a)$ و $q' = \delta(q, a)$ معادل نباشند، یعنی رشته‌ای مانند w' وجود دارد که یکی از آن‌ها را به حالت نهایی می‌برد و دیگری را نه. پس رشته‌ی aw' هم یکی از p, q را به حالت نهایی می‌برد و دیگری را نه. در حالی که این مطلب با فرض اولیه که p, q معادلند در تناقض است. در نتیجه p', q' هم معادل خواهند بود.



■

لم ۳ (*عکس نقیض لم ۲*) *به ازای هر دو حالت p, q در یک DFA :*

$$(\exists a \in \Sigma : \delta(p, a) \not\equiv \delta(q, a)) \Rightarrow p \not\equiv q$$

در DFA ی مثال ۱، می‌دانیم حالت‌های C و G غیرمعادلند. و داریم که:

$$\hat{\delta}(B, \circ) = G$$

و

$$\hat{\delta}(F, \circ) = C$$

در نتیجه طبق لم می‌توان گفت که B و F نیز غیرمعادلند.

۳ الگوریتم پر کردن جدول

الگوریتمی که در زیر به عنوان «الگوریتم پر کردن جدول»[۴] شرح می‌دهیم، الگوریتمی بازگشتی است که به دنبال تمام جفت‌های تمایزپذیر می‌گردد.
ابتدا جدولی داریم که هر جفت از حالت‌های DFA را یک بار دو به دو با هم مقایسه می‌کند.
- پایه: طبق لم ۱:

$$f \in F \wedge q \notin F \Rightarrow f \not\equiv q$$

[۴] Table Filling Algorithm

چون حالت‌های نهایی و غیرنهایی غیرمعادلند، ابتدا تمام جفت‌هایی که یکی نهایی و دیگری غیرنهایی است را به عنوان حالت‌های غیرمعادل علامت می‌زنیم.

- گام استقرا: با استفاده از لم ۳:

$$(\exists a \in \Sigma : \delta(p, a) \not\equiv \delta(q, a)) \Rightarrow p \not\equiv q$$

و همچنین حالت‌های غیرمعادلی که در پایه مشخص شده‌اند، سایر حالت‌های غیرمعادل را نیز به دست می‌آوریم.

این الگوریتم را روی DFA شکل ۳ پیاده می‌کنیم.

از آن جایی که تنها حالت C حالت نهایی است، پس برای پایه داریم: $A, B, D, E, F, G, H \not\equiv C$

							B
					×	×	C
				×			D
				×			E
				×			F
				×			G
				×			H
G	F	E	D	C	B	A	

جدول ۱: گام اول

(علامت × نشان دهنده‌ی آن است که دو حالت غیرمعادلند.)

گام بعد آن است که مطابق لم ۳ از غیرمعادل بودن C با سایر حالت‌ها استفاده کنیم. مثلاً: $A \not\equiv C$

به ازای هر کدام از حروف الفبا حالاتی را که با ۰ و ۱ به A و C می‌روند، می‌یابیم. چون خود A و C غیرمعادلند، حالت‌های مذکور نیز غیرمعادل خواهند بود.

به ازای حرف ۰ داریم:

$$(\delta(?, \circ) = A \Rightarrow ? = C) \wedge (\delta(?, \circ) = C \Rightarrow ? \in \{D, F\}) \Rightarrow C \not\equiv D, C \not\equiv F$$

که هیچ یک از ناهم‌ارزی‌های به دست آمده در بالا اطلاعات جدیدی برای پر کردن جدول به ما نمی‌دهد.

و از آن جایی که هیچ حالتی وجود ندارد که با حرف ۱ به حالت A برود، در نتیجه بررسی حرف ۱ را ادامه نمی‌دهیم.

به سراغ $B \not\equiv C$ می‌رویم:

مجددا به ازای حرف ۰ داریم:

$$(\delta(?, \circ) = B \Rightarrow ? = A) \wedge (\delta(?, \circ) = C \Rightarrow ? \in \{D, F\}) A \not\equiv D, A \not\equiv F$$

در این مرحله غیرمعادل بودن دو زوج (A, F) و (A, D) حاصل می‌شود. غیرمعادل بودن این دو زوج را نیز در جدول علامت $(\times)$ می‌زنیم.

در این جا نیز هیچ حالتی وجود ندارد که با حرف ۱ به B برود، در نتیجه نیازی به ادامه‌ی بررسی نیست.

به همین ترتیب برای بقیه‌ی حالت‌ها الگوریتم را ادامه داده و جدول را کامل می‌کنیم.

در نهایت جدول به صورت زیر درمی‌آید:

خانه‌هایی که علامت نخورده‌اند، زوج‌های معادل جدول‌اند که از قرار (A, E) و (B, H) و (D, F) می‌باشند؛ در واقع پس از پایان الگوریتم، خانه‌هایی از جدول که علامت نخورده‌اند، حالت‌های معادل را نشان می‌دهند.

							B
					$\times$	$\times$	C
				$\times$		$\times$	D
				$\times$			E
				$\times$		$\times$	F
				$\times$			G
				$\times$			H
G	F	E	D	C	B	A	

جدول ۲: گام دوم

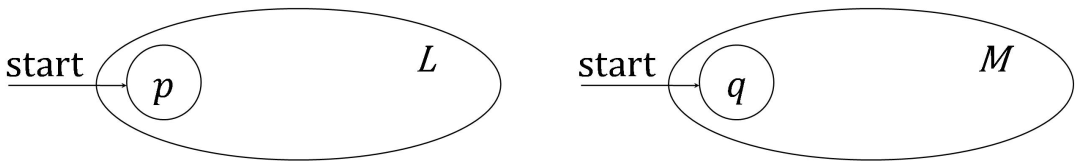
						$\times$	B
					$\times$	$\times$	C
				$\times$	$\times$	$\times$	D
			$\times$	$\times$	$\times$		E
		$\times$		$\times$	$\times$	$\times$	F
	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	G
$\times$	$\times$	$\times$	$\times$	$\times$		$\times$	H
G	F	E	D	C	B	A	

جدول ۳: گام نهایی

قضیه ۴ *اگر پس از خاتمه‌ی الگوریتم پرکردن جدول حالت‌های به عنوان حالت‌های تمایزپذیر (غیرمعادل) شناخته نشوند، آن‌گاه این حالت‌ها تمایزناپذیرند (معادلند).*

۴ مسئله برابری دو زبان

جلسه‌ی پیش با استفاده از تشکیل DFA حاصلضرب یک راه حل برای پاسخ به این سوال که آیا دو زبان داده‌شده برابرند یا نه ارائه دادیم. حال با استفاده از مفهوم حالت‌های معادل در یک DFA، روش دیگری برای بررسی برابری دو زبان ارائه می‌کنیم.
در واقع دو زبان L و M داده شده‌اند، می‌خواهیم ببینیم آیا $L = M$ است؟
(فرض می‌کنیم که DFAهای دو زبان را داریم.)



می‌توان الگوریتم پرکردن جدول را برای اجتماع این دو اوتوماتا در نظر گرفت (دقت کنید که DFA اجتماع دارای

دو حالت اولیه است، اما این مسئله حائز اهمیت نیست؛ چون الگوریتم پرکردن جدول از اولیه بودن یک حالت هیچ استفاده‌ای نمی‌کند).

قضیه ۵ *$p_\circ$ را حالت آغازین L و $q_\circ$ را حالت آغازین M در نظر بگیرید. اگر الگوریتم پرکردن جدول را بر اجتماع DFAهای زبان‌های L و M اعمال کنیم، داریم:*

$$p_\circ \equiv q_\circ \Leftrightarrow L = M$$

۵ افراز مجموعه حالت ها و محاسبه DFAی کمینه

قضیه ۶ *رابطه‌ی تمایزناپذیری (معادل بودن) حالت‌ها، یک رابطه‌ی هم ارزی است یعنی خواص زیر را دارا می باشد:*

۱. *بازتابی*

$$p \equiv p$$

۲. *تقارنی*

$$p \equiv q \Rightarrow q \equiv p$$

۳. *تعدی*

$$(p \equiv q) \wedge (q \equiv r) \Rightarrow (p \equiv r)$$

برهان. اثبات قسمت‌های اول و دوم واضح بوده و به خواننده واگذار می‌شوند. قسمت سوم را با برهان خلف اثبات می‌کنیم. فرض کنید $p \equiv q$ و $q \equiv r$ ولی $p \not\equiv r$. یعنی رشته‌ی wای وجود دارد که مثلاً با پیمایش آن از p به حالت نهایی می‌رسیم و از r به حالت نهایی نمی‌رسیم. چون p و q معادلند، پس از حالت q نیز با پیمایش w به حالت نهایی می‌رسیم و نیز چون r و q معادلند، از حالت q با پیمایش w به حالت نهایی نمی‌رسیم. که این دو حکم با هم در تناقض‌اند، در نتیجه p و r معادلند. ■

نتیجه‌ی قضیه ۳ این است که می‌توانیم مجموعه حالت‌ها در یک DFA را به کلاس‌های هم‌ارزی افراز کنیم.

در واقع اگر $A = (Q, \Sigma, \delta, q_0, F)$ یک DFA باشد، رابطه‌ی معادل بودن، Q را به مجموعه حالت‌های هم‌ارزیِ $S_1, S_2, ..., S_t$ افراز می‌کند.

حال هم‌ارزی DFA داده شده در شکل ۳ را تعیین می‌کنیم. داریم:

$$Q = A, B, C, D, E, F, G, H$$

پس از اجرای الگوریتم کلاس‌های هم‌ارزی را به صورت زیر به دست می‌آوریم:

$$S_1 = \{A, E\}, S_2 = \{B, H\}, S_3 = \{C\}, S_4 = \{D, F\}, S_5 = \{G\}$$

تعریف ۳ *DFAی $A = (Q, \Sigma, \delta, q_0, F)$ را داریم. DFAی معادل آن (A_{eq}) را به صورت زیر تعریف می‌کنیم:*

$$A_{eq} = (Q_{eq}, \Sigma, \delta_{eq}, q_{eq}, F_{eq})$$

به صورتی که:

$$Q_{eq} = \{S_1, S_2, , S_t\}$$

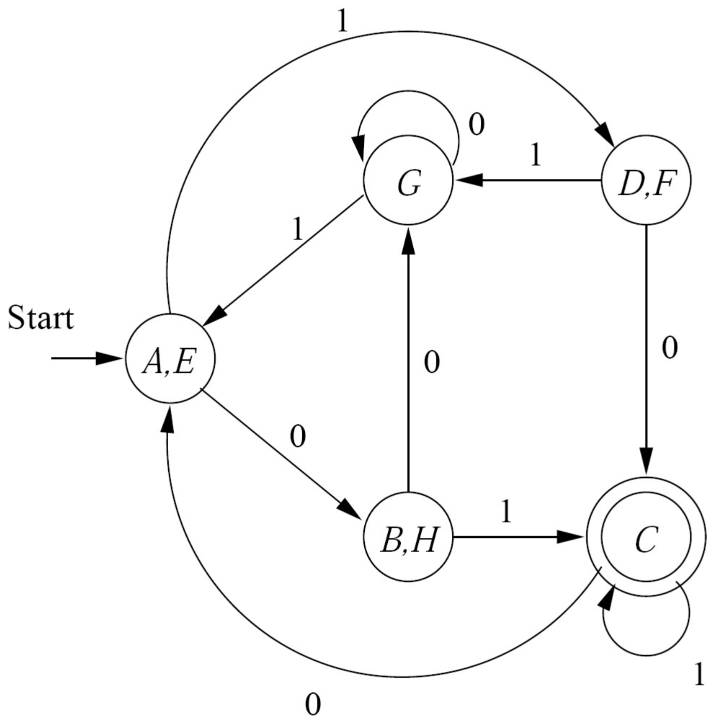
$$q_{eq} = S_i | q_0 \in S_i$$

$$F_{eq} = \{S_i | S_i \cap F \neq \emptyset\}$$

δ_{eq} را نیز بدین صورت تعریف می‌کنیم که به ازای هر $S \in Q_{eq}$ و هر $a \in \Sigma$ تعریف می‌کنیم $\delta_{eq}(S, a) = T$ که $T \in Q_{eq}$ مجموعه‌ایست که شامل $\delta(p, a)$ می‌باشد که p یک عضو دلخواه S است. یعنی به زبان دیاضی داریم:

$$(\delta_{eq}(S, a) = T) | (S \in Q_{eq}) \wedge (\delta(p, a) \in T) \wedge (p \in S)$$

بنابراین DFA معادل حاصل برای DFA شکل ۳ به صورت زیر درمی‌آید:


1
0
1
G
D,F
1
Start
A,E
0
0
0
B,H
1
C
1
0