



جلسه‌ی ۲: اتوماتای محدود قطعی (DFA)

نگارندگان: محمد مهدی محاویری و معین زمانی

مدّرس: دکتر شهرام خزائی

۱ مقدمه

یک اتوماتای محدود قطعی^۱ که آن را به اختصار DFA می‌نامیم، مجموعه‌ی محدودی از حالت‌ها است که یکی از آنها آغازین و تعدادی از آنها حالت نهایی نامیده می‌شوند. هدف یک DFA گرفتن رشته‌ای بر روی یک الفبای خاص و پردازش آن است. یک اتوماتا بر اساس تابع انتقال حالتش بین حالت‌های مختلف گذار می‌نماید. ابتدا جهت آشنایی از یک مثال استفاده می‌کنیم.

۱.۱ بازی تنیس

بازی تنیس معمولاً در سه یا پنج set اجرا می‌شود. هر بازیکنی که تعداد set بیشتری را برده باشد، برنده بازی خواهد شد. هر set خود از چند game تشکیل شده است. نحوه‌ی امتیاز دهی هر game به شکل زیر است.

The number of services	Score
start	Love
1st service	15
2nd service	30
3rd service	40

شروع کننده‌ی بازی "سرور"^۲ و طرف مقابلش "رقیب"^۳ نامیده می‌شود. در شروع بازی امتیاز به صورت Love-Love بیان می‌شود و برای امتیازهای بعدی جدول بالا مورد استفاده قرار می‌گیرد. به عنوان مثال اگر سرور امتیاز سه سرویس و رقیب امتیاز چهار سرویس را کسب کرده باشد، امتیازها به صورت 30-40 اعلام می‌گردد. در صورتی که تعداد سرویس‌های برده شده برابر و بیشتر از سه باشد، بازی اصطلاحاً در وضعیت مساوی^۴ اعلام می‌شود.

بازیکنی که امتیاز حداقل چهار سرویس را با حداقل اختلاف دو سرویس کسب کرده باشد برنده‌ی game خواهد بود. در صورتی که تعداد سرویس‌های برده شده به چهار نرسیده باشد یا اختلاف تعداد سرویس‌های برده شده کمتر از دو باشد، بازی آنقدر ادامه پیدا می‌کند تا این شرط برقرار شود و برنده تعیین گردد. بنابراین هنگامی که بازی در

^۱Deterministic Finite Automata

^۲server

^۳opponent

^۴deuce

که اولین امتیاز را چه کسی کسب کند (یعنی اولین حرف رشته‌ی ورودی چه چیزی باشد)، بازی به حالت 15-Love می‌رود. در این حالت اگر بازیکن رقیب امتیازی کسب کند، بازی به حالت 15-all می‌رود. در غیر این صورت بازی بسته به این که در چه حالتی قرار دارد به حالت 30-Love یا Love-30 انتقال پیدا می‌کند. این کار تا جایی ادامه پیدا می‌کند که یا به یکی از حالت‌های نهایی برسیم که در این صورت دیگر ادامه‌ی رشته برای ما مهم نیست و به یک حالت مرده انتقال پیدا خواهیم کرد، یا این که به حالت deuce برسیم. به حالت مساوی برسیم، با گرفتن یک امتیاز بسته به این که چه کسی امتیاز را کسب کرده است، به حالت Ad-in یا Ad-out می‌رویم. در این حالت اگر امتیاز بعدی را همان شخص کسب کند، آن شخص برنده می‌شود و به یک حالت نهایی می‌رویم: حالت "S wins" برای زمانی که سرور برنده شده است و حالت "O wins" برای زمانی که رقیبش برنده شده است. در غیر این صورت مجدداً به حالت deuce باز خواهیم گشت.

۲ تعریف اتوماتای محدود قطعی

حال که مثالی از یک DFA را دیدیم، می‌خواهیم یک تعریف رسمی از آن ارائه نماییم.

تعریف ۱ یک DFA یک پنج‌تایی مرتب به شکل زیر است

$$D = (Q, \Sigma, \delta, q_0, F)$$

که:
 Q ، مجموعه‌ای متناهی از حالت‌ها^۵ است؛
 Σ ، یک الفبای محدود است؛
 δ که $\delta : Q \times \Sigma \rightarrow Q$ ، تابع انتقال حالت^۶ است؛
 q_0 که $q_0 \in Q$ ، حالت آغازین است؛
 F که $F \subseteq Q$ ، مجموعه‌ی حالت‌های نهایی است.

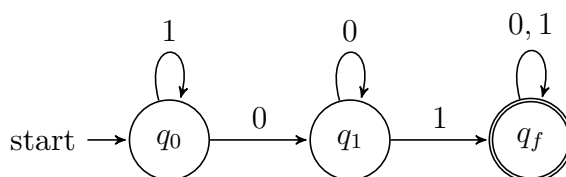
همانطور که قبلاً توضیح داده شد، D یک DFA است که از حالات Q تشکیل شده است. این DFA از حالت اولیه q_0 شروع می‌کند، رشته‌ی w را از Σ^* می‌گیرد، حرف به حرف می‌خواند و بر اساس تابع انتقال حالت δ بین حالات مختلف گذار می‌کند. اگر پس از پایان پردازش رشته‌ی ورودی، DFA در یکی از حالات نهایی F بود، این DFA رشته‌ی ورودی را پذیرفته است.

مثال ۱ می‌خواهیم یک DFA بسازیم که زبان $L = \{x^0 \setminus y \mid x, y \in \{0, 1\}^*\}$ را بپذیرد. در ابتدا باید دقت کنیم که زبان L در حقیقت مجموعه‌ی تمام رشته‌هایی است که حداقل یک 0 درون خود داشته باشند. پس برای تشخیص دادن این رشته‌ها کافی است اولین 0 را تشخیص دهیم. به همین جهت یک حالت (که همان حالت آغازین یعنی q_0 است) را برای زمانی در نظر می‌گیریم که در رشته‌ای که تا کنون پردازش شده هیچ 0 ای مشاهده نشده است. با مشاهده‌ی اولین 0 به یک حالت جدید (q_1) می‌رویم. با حضور در این حالت یعنی ما تا به حال صفر تا یا بیشتر 1 و بعد از آن حداقل یک 0 دیده‌ایم. مجدداً تا زمانی که در حالت q_1 هستیم و حرف خوانده شده از رشته 0 است در این حالت می‌مانیم. اکنون کافی است یک عدد 1 مشاهده کنیم تا رشته‌ی ورودی عضوی از زبان L شود. بنابراین با دیده شدن اولین 1 به یک حالت جدید (q_2) انتقال پیدا می‌کنیم. اگر در این حالت قرار داشته باشیم، یعنی رشته‌ی ما حداقل یک 0 را شامل می‌گردد، پس رشته‌ی ورودی عضوی از زبان L

^۵states

^۶transition function

است، مستقل از این که حروف بعدی آن چه باشند. پس با دریافت هر حرف جدیدی در این حالت می‌مانیم. این حالت، حالت نهایی اتوماتای ما خواهد بود، زیرا حضور در این حالت به معنی عضویت رشته‌ی ورودی در زبان L است. شکل زیر گراف انتقال حالت اتوماتای طراحی شده را نمایش می‌دهد.



۳ زبان یک اتوماتای محدود قطعی

تعریف ۲ (غیر رسمی) ^۵ یک DFA دنباله‌ی $w = a_0 a_1 \dots a_n$ را می‌پذیرد اگر مسیری داشته باشیم که:

- از حالت آغازین شروع شود،
- به حالت نهایی ختم شود،
- و دنباله‌ی یال‌های مسیر، دارای برچسب حرف‌های دنباله باشند.

برای رسمی ^۶ کردن تعریف بالا، ابتدا نیاز است که تابع جدیدی تعریف کنیم.

تعریف ۳ تابع انتقال حالت بسط یافته ^۷ که آن را با $\hat{\delta}$ نمایش می‌دهیم، تابعی است که به صورت زیر تعریف می‌شود.

$$\hat{\delta} : Q \times \Sigma^* \rightarrow Q$$

$$\hat{\delta}(q, \epsilon) = q$$

$$\forall a \in \Sigma \wedge x \in \Sigma^* \Rightarrow \hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$$

با استفاده از این تابع می‌توان گفت یک اتوماتا رشته‌ی w را می‌پذیرد اگر و فقط اگر $\hat{\delta}(q_0, w)$ یک حالت نهایی باشد.

حال با استفاده از این تابع، یک تعریف رسمی از زبانی که یک اتوماتا می‌پذیرد ارائه می‌دهیم.

تعریف ۴ زبان اتوماتای $A = (Q, \Sigma, \delta, q_0, F)$ را، که با $L(A)$ نمایش می‌دهیم، به صورت زیر تعریف می‌کنیم:

$$L(A) = \{w \mid \hat{\delta}(q_0, w) \in F\}$$

^۵informal

^۶formal

^۷extended transformation function