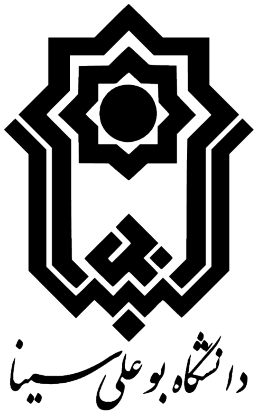


پروژه نهایی درس برنامه سازی پیشرفته

آزمایشگاه نظریه اتوماتا

مدرس : دکتر مرتضی یوسف صنعتی

کمک اساتید: امیرعباس اسدی، امیرحسین بابائیان، احمد جعفری، فاطمه طاهری نژاد

بهار ۱۳۹۸

مختصری در مورد نظریه اتوماتا

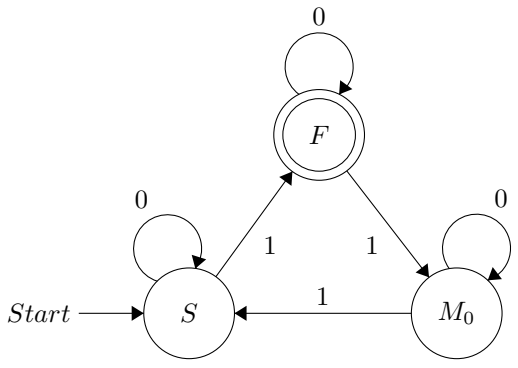
آشنایی با نظریه اتوماتا

یکی از مهم ترین شاخه های علوم کامپیوتر Automata Theory یا نظریه اتوماتا است. یکی از اهداف این نظریه تعیین قدرت ماشین های مختلف از جمله کامپیوتر ها در حل مسائل است.برای رسیدن به این مقصود از جزئیات صرف نظر شده و ماشین هایی مجرد برای انجام محاسبات تعریف می شود. در این نظریه ماشین هایی با امکانات مختلف تعریف شده و قدرت آن ها در حل مسئله با یکدیگر مقایسه می شود.نظریه اتوماتا از مباحث مورد تحقیق علوم کامپیوتر در دوره معاصر است که بسیاری از مسائل آن بدون پاسخ مانده است.در ادامه با تعدادی از ماشین های تعریف شده در این نظریه آشنا می شویم.برای کسب اطلاعات بیشتر در مورد این نظریه می توانید به کتاب های زیر مراجعه کنید:

- Introduction to Automata Theory, Languages, and Computation, John E. Hopcroft
- An Introduction to Formal Languages and Automata, Peter Linz
- Introduction to the Theory of Computation, Michael Sipser

ماشین DFA

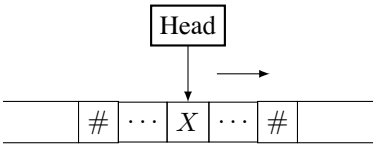
یکی از ساده ترین ماشین های نظریه اتوماتا DFA یا Deterministic Finite Automaton است. این ماشین از یک مجموعه حالت و قواعد جابجایی بین این حالات تشکیل می شود.یکی از این حالات به عنوان حالت شروع و تعدادی از آن ها به عنوان حالت پایان تعریف می شوند.این ماشین یک ورودی دریافت کرده و حالت آن متناسب با ورودی داده شده تغییر می کند.بعد از این که تمام ورودی توسط ماشین خوانده شد، اگر ماشین در حالت پایانی قرار داشت می گوییم ماشین ورودی را پذیرفته و اگر در حالتی غیر پایانی قرار داشت می گوییم ماشین ورودی را رد کرده است.ورودی این ماشین به صورت یک رشته است.در ابتدا ماشین در حالت آغازین قرار دارد و شروع به خواندن ورودی از سمت چپ می کند.به ازای هر حرفی که از ورودی خوانده می شود، حالت ماشین ممکن است عوض شود.معمولا این ماشین ها را به صورت گرافی از حالات نمایش می دهند. به طوریکه هر حالت آن را به صورت یک راس گراف و قواعد انتقال از هر حالت به حالت دیگر را با استفاده از یال های گراف مشخص می کنند.برای مثال ماشین زیر را در نظر بگیرید:

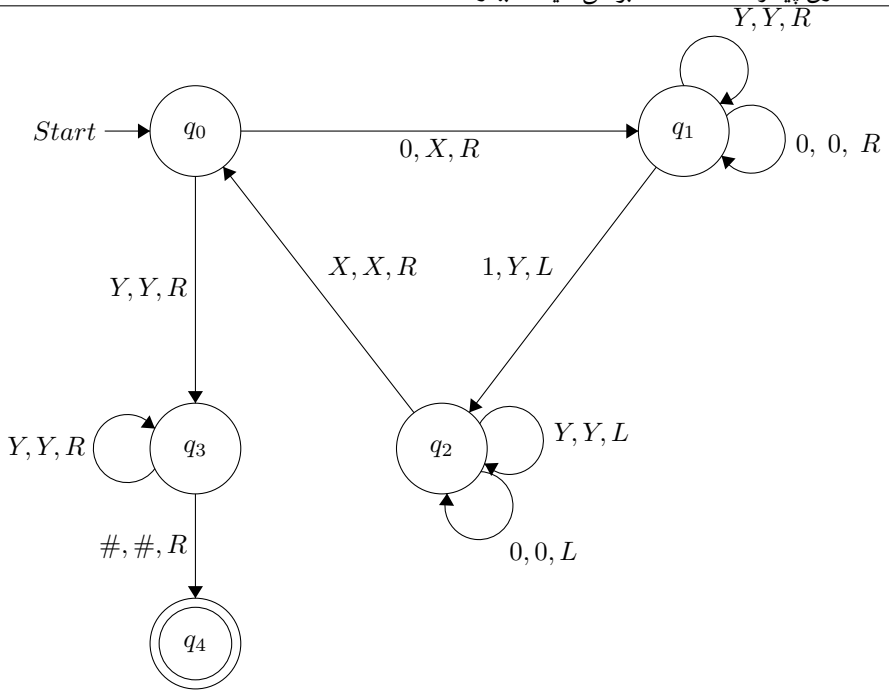


ماشین بالا از مجموعه حالت های $\{S, F, M_0\}$ تشکیل شده است.حالت آغازین این ماشین S و F حالت پایانی آن است. حالت های پایانی را با کشیدن دایره آن ها به صورت دوخطی متمایز می کنند. فرض کنیم ورودی 01110 را به ماشین بدهیم.ماشین ابتدا در حالت آغازین یعنی در S قرار دارد.حال شروع می کند به خواندن ورودی از سمت چپ.اولین حرف ورودی 0 است.گراف ماشین را ببنید. در این گراف یک یال از راس S به خودش وصل شده که روی آن حرف 0 نوشته شده است.این یعنی بعد از خواندن 0 ماشین در حالت S باقی می ماند.بعد از این ماشین حرف 1 را می خواند. طبق گراف ماشین اگر در حالت S باشیم و 1 بخوانیم به حالت F منتقل می شویم. در اینجا حروف ما متشکل از 0 و 1 است.برای هر حالت مانند گراف بالا باید مشخص باشد که به ازی هر حرف از هر حالت به چه حالتی منتقل می شویم.بعد از خواند حروف بقی مانده ماشین به حالت S می رسد و ورودی به پایان می رسد. چون بعد از پایان ورودی ماشین در حالتی توقف کرده که یک حالت نهایی نیست می گوییم ماشین ورودی 01110 را رد کرده است.اگر امتحان کنید خواهید دید که این ماشین بعد از خواندن ورودی 11101 در حالت F متوقف می شود. چون F یک حالت پایانی است بنابراین می گوییم ماشین ورودی 11101 را می پذیرد.با کمی دقت می توان متوجه شد این ماشین ورودی هایی را می پذیرد که باقیمانده تقسیم تعداد 1 ها در آن ها به عدد 3 برابر با 1 است.پس از نگاه دیگر می توان گفت این ماشین قدرت محاسبه عمل باقیمانده تقسیم به 3 را دارد.اما آیا یک DFA قادر به حل هر مساله ای است؟ مثلا می توانیم یک DFA تعریف کنیم که ورودی هایی را بپذیرد که تعداد 1 ها در آن ها مربع کامل است؟ پاسخ منفی است. ماشین های DFA قدرت حل مساله مشخصی دارند و مثلا قادر به حل مساله ذکر شده نیستند.برای حل مسائل سخت تر ماشین های قدرتمندتری تعریف می شود.

ماشین تورینگ

یکی دیگر از مدل های محاسباتی که در نظریه اتوماتا تعریف می شود ماشین تورینگ است.ماشین تورینگ هم مانند DFA متشکل از تعدادی حالت است که یکی از آن ها به عنوان حالت شروع و تعدادی به عنوان حالت پایانی تعریف می شوند.ماشین تورینگ برخلاف DFA دارای حافظه است. حافظه ماشین تورینگ شکل بسیار ساده ای دارد.حافظه ین ماشین یک نوار مانند آرایه ای است که از سمت چپ و راست طول بی نهایت دارد.در هر کدام از خانه های این نوار می توان یک حرف را نوشت.ورودی ماشین تورینگ در ابتدا روی این نوار نوشته شده است.باقی خانه های نوار به طور پیشفرض حاوی کاراکتر # هستند که نشان دهنده خالی بودن آن هاست.یک اشاره کننده به نام Head به یکی از خانه های نوار حافظه اشاره می کند.ماشین تورینگ قادر است حرفی که Head به آن اشاره می کند را بخواند یا عوض کند.همچنین می توان Head را به سمت راست یا چپ حرکت داد. نمایش یک ماشین تورینگ مانند DFA است با این تفاوت که انتقال بین حالات علاوه بر حالت فعلی به حرفی که Head روی آن قرار دارد هم بستگی دارد.این ماشین در ابتدا در حالت شروع قرار دارد و Head نوار حافظه هم روی اولین حرف ورودی ماشین قرار دارد.اگر در حالت q_i باشیم و روی یک یال به حالت q_j عبارت α, β, γ نوشته شده باشد به این معنی است که اگر حرفی که Head نوار روی آن است α باشد، به حالت q_j منتقل می شویم و حرف β به جای α در خانه ای از نوار که Head روی آن قرار دارد نوشته می شود.γ می تواند مقادیر L و یا R داشته باشد که به ترتیب یعنی Head یک خانه به سمت چپ یا راست حرکت کند.بدین ترتیب میتوان محتوای نوار حافظه را تغییر داد. در شکل زیر نمونه نوار ماشین تورینگ آمده است.همچین در صفحه بعد یک نمونه ماشین تورینگ رسم شده است. این ماشین ورودی هایی را قبول می کند که به شکل 0000...1111 باشند به طوریکه تعداد 1 ها و تعداد 0 ها برابر باشد.لازم به ذکر است اگر در حالتی که در آن قرار داریم برای حرف روی نوار انتقالی تعریف نشده باد آنگاه ماشین ورودی را نمی پذیرد.در آخر اگر ماشین تورینگ بعد از خواندن ورودی در یک حالت پایانی قرار گرفت می گوییم ورودی را پذیرفته است.





نکته بسیار جالب در مورد ماشین تورینگ قدرت حل مساله آن است.علی رغم ساختار ساده این ماشین ، تورینگ در تز خود ادعا میکند این ماشین قادر است تمام مسائلی که هر رایانه ای قادر به حل آن ها است حل کند و در کل اگر مسئله ای قابل حل باشد حتما ماشین تورینگ می تواند آن را حل کند. قدرت ماشین تورینگ در حل مسئله از هر کامپیوتری بیشتر است!

شرح پروژه آزمایشگاه نظریه اتوماتا

هدف از این پروژه طراحی یک برنامه برای کار با مدل های محاسباتی تعریف شده در نظریه اتوماتا است.امکانات این برنامه در ادامه آمده است.برای درک بهتر عملکرد برنامه خواسته شده توصیه می شود نرم افزار JFLAP را بررسی کنید.

قابلیت تعریف و کار با DFA

در این برنامه بخشی برای کار با DFA وجود دارد. امکانات برنامه در مورد تعریف DFA به صورت زیر است:

- تعریف یک DFA
- افزودن تعداد حالات دلخواه به ماشین تعریف شده و حذف حالات
- تعیین حالت شروع و حالت های پایانی
- تعیین قواعد انتقال بین حالات و تغییر آن ها
- تغییرات مذکور بایستی با موس روی شکل گراف ماشین قابل اعمال باشد.
- برنامه شما بایستی یال های گراف ماشین را به نحوی رسم کند که گراف حاصل حتی الامکان مسطح باشد(یعنی گرافی که یال های آن یکدیگر را قطع نکنند.)
- ذخیره ماشین طراحی شده در فایل و فراخوانی ماشین ذخیره شده در یک فایل

بعد از تعریف DFA امکانات برنامه برای کار با DFA تعریف شده شامل موارد زیر است:

- دادن ورودی به ماشین و نمایش نتیجه نهایی
- دادن ورودی به ماشین و نمایش مرحله به مرحله پردازش ورودی توسط ماشین. در هر مرحله حالت ماشین باید نمایش داده شود.
- دادن چندین ورودی به ماشین و نمایش نتیجه همه ورودی ها
- برای هر ورودی نتیجه به صورت Accept و یا Reject نمایش داده شود.

قابلیت های کار با ماشین تورینگ

بخشی از این برنامه به کار با ماشین تورینگ اختصاص دارد که امکانات آن در مورد تعریف این ماشین به صورت زیر است:

- تعریف یک ماشین تورینگ
- افزودن تعداد حالات دلخواه به ماشین تعریف شده و حذف حالات

- تعیین حالت شروع و حالت های پایانی
- تعیین قواعد انتقال بین حالات و ویرایش نوار و جابجایی Head
- تغییرات مذکور بایستی با موس روی شکل گراف ماشین قابل اعمال باشد.
- برنامه شما بایستی یال های گراف ماشین را به نحوی رسم کند که گراف حاصل حتی الامکان مسطح باشد(یعنی گرافی که یال های آن یکدیگر را قطع نکنند.)
- ذخیره ماشین طراحی شده در فایل و فراخوانی ماشین ذخیره شده در یک فایل

بعد از تعریف یک ماشین تورینگ امکانات زیر در مورد کار با آن وجود دارد:

- دادن ورودی به ماشین و نمایش نتیجه نهایی و محتوای نهایی نوار حافظه
- دادن ورودی به ماشین و نمایش مرحله به مرحله پردازش ورودی توسط ماشین. در هر مرحله حالت فعلی ماشین و محتوای نوار حافظه و محلی که Head به آن اشاره می کند باید نمایش داده شود
- دادن چندین ورودی به ماشین و نمایش نتیجه همه ورودی ها
- برای هر ورودی نتیجه به صورت Accept و یا Reject نمایش داده شود.

نمرات مثبت مربوط به امکانات برنامه

افزودن مواد زیر به پروژه حائز نمره مثبت است.میزان نمره مثبت تعلق گرفته به هر یک به میزان سختی آنها و کیفیت انجام آن ها وابسته است:

- قابلیت استفاده از یک ماشین طراحی شده به عنوان یک بخش از ماشین در حال طراحی
- پیاده سازی مدل های دیگر تعریف شده در نظریه اتوماتا مانند NFA، NPDA و یا انواع دیگر ماشین تورینگ.
- قابلیت ساخت ماشین اجتماع دو ماشین از روی دو ماشین طراحی شده. ماشین اجتماع دو ماشین ، ماشینی است که ورودی هایی را می پذیرد که حداقل یکی از آن دو ماشین بپذیرد.
- قابلیت ساخت ماشین اشتراک دو ماشین از روی دو ماشین طراحی شده. ماشین اشتراک دو ماشین، ماشینی است که ورودی هایی را بپذیرد که توسط هر دو ماشین پذیرفته می شوند.
- افزودن راهنمای کاربردی به برنامه
- ذخیره سازی تصویر ماشین طراحی شده در قالب PDF و یا فرمت های تصویری.
- پیاده سازی قفل نرم افزاری برای برنامه، به طوریکه کاربر در حالت پیشفرض صرفا قادر به استفاده از بخش DFA باشد اما با وارد کردن شماره سریال برنامه بخش ماشین تورینگ هم فعال شود.سنجش اعتبار سریال وارد شده توسط کاربر و تکراری نبودن آن باید از طریق ارتباط با سرور انجام گیرد.

نکات مربوط به پیاده سازی پروژه

- رعایت اصول شی گرایی و طراحی مناسب کلاس ها و استفاده صحیح از مفهوم چندریختی
- پروژه باید دارای رابط گرافیکی باشد. حتما باید شکل گراف ماشین در برنامه موجود باشد و تغییرات باید به وسیله موس روی آن شکل قابل انجام باشد.
- برای پیاده سازی پروژه می توانید از یکی از زبان های C++, Java, C# استفاده کنید.
- استفاده از فریموورک ها و کتابخانه های مختلف به جز کتابخانه های طراحی شده برای نظریه اتوماتا مجاز است.
- در حین پیاده سازی از Git استفاده کنید.

نمرات مثبت مربوط به پیاده سازی

- خلاقیت در طراحی شی گرا
- پیاده سازی تست خودکار برای کلاس های نوشته شده
- مستند سازی کامل و اصولی

نکات مربوط به تحویل پروژه

- پروژه باید به صورت **تک نفره** انجام شود.
- نوشتن گزارش برای پروژه الزامی است.
- تاریخ تحویل پروژه روز شنبه ۱۹ مرداد است. تحویل پروژه صرفا به صورت **حضوری** انجام می شود.
- فایل های پروژه اعم از کد ها و گزارش بایستی تا ساعت ۱۰ صبح روز جمعه ۱۸ مرداد در سامانه Quera ارسال شوند.
- این پروژه برای پیشرفت مهارت های شما طراحی شده است. مستقل از توانایی شما در برنامه نویسی و نتیجه ای که به آن دست خواهید یافت، نمره ای که به پروژه های برون سپاری شده تعلق می گیرد نمره پایین تری نسبت به حاصل کار خود شما خواهد بود.(به طور دقیق تر برابر با 0.25 به عنوان نمره نهایی درس).