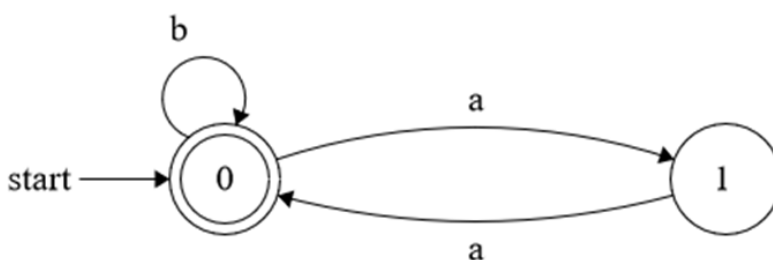


پروژه دوم درس ساختمان داده‌ها

پیاده‌سازی خودکارهای حالت‌متناهی

در این پروژه قصد داریم با استفاده از گراف‌ها، خودکارهای حالت‌متناهی^۱ را مدل‌سازی کنیم. همان‌طور که در درس مدارهای منطقی خوانده‌اید (و در درس نظریه زبان‌ها و خودکارها به طور مفصل درباره آن‌ها خواهید دید)، با استفاده از خودکارهای حالت‌متناهی می‌توان یک زبان صوری^۲ را بیان کرد و روی آن پردازش کرد. در ساده‌ترین حالت، تشخیص عضویت یک رشته^۳ در زبان است. به طور مثال، خودکار زیر، رشته‌هایی را که شامل تعداد صفر یا بیش از آن جفت 'a' و تعداد دلخواه 'b' هستند را می‌پذیرد؛ یعنی با شروع از حالت آغازین، در صورتی که هر تعداد (صفر یا بیشتر) 'b' در اول رشته وجود داشته باشد و سپس 'aa' در رشته وجود داشته باشد و همین الگو مجدداً تکرار شود، آن رشته پذیرفته^۴ می‌شود.



پروژه

در این پروژه، باید خودکارهای حالت‌متناهی با توجه به موارد زیر پیاده‌سازی شوند و قسمت‌های زیر را داشته باشد:

۱. همه‌ی ساختمان داده‌های مورد نیاز با توجه به تعریف پروژه باید پیاده‌سازی شوند. استفاده از ساختمان داده‌های آماده مجاز نیست. فقط استفاده از آرایه (ساده، ArrayList یا Vector) مجاز است. صف، پشته، لیست پیوندی، درخت یا هر ساختمان داده دیگر مورد استفاده باید پیاده‌سازی شود (به عنوان درس توجه کنید!). همچنین برنامه شما باید دارای رابط کاربری گرافیکی (GUI) مناسب باشد.

۲. **تعریف یک خودکار حالت‌متناهی:** کاربر می‌تواند یک خودکار حالت‌متناهی دلخواه را تعریف کند. منظور از تعریف یک خودکار حالت‌متناهی، ارائه جدولی مانند چیزی است که در ادامه آورده شده است. این جدول بیانگر همان مثال قبلی است. پس از آن، کاربر باید حالت اولیه^۵ و حالت(های) پایانی^۶ را مشخص کند. هر خودکار فقط یک حالت اولیه دارد ولی می‌تواند چندین حالت پایانی (از جمله همان حالت اولیه) را شامل شود. در مثال شکل بالا، حالت شماره صفر، هم حالت اولیه و هم حالت پایانی است.

¹ Finite-State Automata

² Formal Language

³ String

⁴ Accept

⁵ Initial State

⁶ Final State(s)

State	0	1
0	B	a
1	A	-

برنامه شما می‌بایست بتواند چنین جدولی را از کاربر دریافت کند و خودکاره متناظر آن را بسازد. توجه داشته باشید که ممکن است چند حرف نیز روی یک یال نوشته شوند (مثلاً در جدول بالا، عنصر سطر و ستون صفر، به جای 'b'، مقدار 'a,b' قرار بگیرد). همچنین محدودیتی برای تعداد حالت‌ها (گره‌ها) یا گذارها (حرف روی یال‌ها) وجود ندارد. شما باید ساختمان داده مناسب برای انجام این کار را استفاده نمایید.

۳. گرفتن لیست خودکاره‌های حالت‌متناهی تعریف‌شده: برنامه شما می‌بایست بتواند لیست خودکاره‌های حالت‌متناهی که تا به حال تعریف شده‌اند را در قالب لیست همسایگی نمایش بدهد. به عنوان مثال،

State	Adjacent
0	(0, b) (1, a)
1	(0, a)

۴. نمایش گرافیکی خودکاره حالت‌متناهی انتخاب‌شده: کاربر می‌تواند با انتخاب یکی از خودکاره‌های تعریف‌شده، شکل گرافیکی آن را ببیند. برای این کار می‌توانید از ابزار [GraphViz](#)، که ابزار بسیار قدرتمندی برای نمایش گرافیکی گراف‌ها محسوب می‌شود، استفاده کنید. (در صورتی که از زبان جاوا برای پیاده‌سازی استفاده می‌کنید، می‌توانید از [این قطعه کد](#) برای اضافه کردن این ابزار به کد برنامه خود استفاده کنید. کتاب‌خانه این ابزار برای سایر زبان‌های برنامه‌نویسی نیز وجود دارد.)

۵. تشخیص عضویت یک رشته: برنامه یک رشته دلخواه را از کاربر دریافت کرده و عضویت یا عدم عضویت آن رشته در خودکاره حالت‌متناهی انتخاب‌شده را نشان می‌دهد. منظور از عضویت آن است که آیا رشته‌ای که کاربر وارد کرده است، توسط خودکاره پذیرفته می‌شود یا خیر. برای مثالی که از ابتدای این تعریف پروژه روی آن صحبت شده است، داریم:

aabaabaaaa → True

aabaabaaabb → False

۶. تشخیص وجود دور در خودکاره: برنامه باید بتواند برای خودکاره حالت‌متناهی انتخاب‌شده، تشخیص دهد که در آن دور (حلقه) وجود دارد یا خیر.

۷. حذف دورهای موجود در خودکاره: برنامه باید بتواند دورهای موجود در خودکاره را حذف نماید. به نظر شما، خودکاره حاصل‌شده از حذف دورها، از نظر قدرت پردازش و محاسبه با خودکاره اولیه قبل از حذف دورها معادل است؟

نکات پیاده‌سازی

۱. توجه کنید که **خوانا بودن کد** (مانند نام گذاری مناسب متغیرها و توابع، مازول‌بندی منطقی، دندانه‌داربودن^۷ کد، کامنت گذاری) نیز مورد ارزیابی قرار خواهد گرفت و قسمتی از نمره شما را تشکیل خواهد داد.
۲. پیاده‌سازی باید به صورت **تک نفره** باشد و محدودیتی برای زبان پیاده‌سازی وجود ندارد و می‌توان از هر زبانی استفاده کرد اما دقت کنید که استفاده از ساختمان داده‌های آماده و به صورت کتابخانه مجاز نیست.
۳. بحث و بررسی برای فهم الگوریتم‌ها بین دانشجویان آزاد است اما هر دانشجو موظف است به تنهایی پروژه را انجام دهد. همچنین با مواردی که تقلب و کپی‌کردن تشخیص داده شوند، برخورد خواهد شد (برای تشخیص درصد شباهت کدها از سامانه‌ی [Moss](#) استفاده می‌شود).
۵. در زمان تحویل حضوری، برنامه‌ی شما باید برای موارد آزمون^۸ی که مصحح آن‌ها را به برنامه می‌دهد، جواب صحیح بدهد. همچنین دانشجو باید به تمام جزئیات پیاده‌سازی کد کاملاً مسلط باشد. در مورد قسمت‌هایی از کد و نحوه عملکرد برنامه نیز از دانشجو سوال خواهد شد.
۶. برای پرسش و پاسخ درباره پروژه از [این فروم](#) استفاده کنید.
۷. موعد تحویل این پروژه تا **ساعت ۲۳:۵۵ روز دوشنبه ۲۷ دی ۱۳۹۵** خواهد بود. پوشه مربوط به کد پروژه را همراه با یک فایل pdf حاوی شرح انجام پروژه، نحوه اجرای برنامه و گزارش مربوط به تحلیل ساختمان داده‌های مورد استفاده را در قالب یک فایل zip به شکل زیر بارگذاری کنید. زمان و چگونگی نحوه تحویل حضوری متعاقباً اعلام می‌شود.

StudentNumber-FirstName-LastName-Proj2.zip

e.g. 9431555-Ali-Ahmadi-Proj2.zip

⁷ Indentation

⁸ Test Cases