



مبانی نظریه محاسبه

پروژه امتیازی

بهار ۱۴۰۲

۱ مقدمه

همه‌ی ما احتمالاً تا کنون برنامه‌ای به زبان C نوشته‌ایم و پس از کامپایل^۱ شدن، اجرای آن را دیده‌ایم اما شاید به نقش اتوماتای متناهی^۲ در فرایند کامپایل شدن این برنامه توجه نکرده باشیم. یک کامپایلر ابتدا در فاز تحلیل لغوی^۳ برای کنترل نگارش برنامه، درستی بخش‌هایی را از جمله کلمات کلیدی^۴، شناسه‌ها^۵، اعداد^۶، رشته‌ها^۷، توضیحات^۸ و ... بررسی می‌کند. برای این کار، کامپایلر نگارش مورد قبول هر بخش را به صورت یک عبارت منظم^۹ به پیش‌گر^{۱۰} خود می‌دهد تا درستی برنامه را از نظر زبانی با استفاده از یک اتوماتای متناهی بیازماید. بنابراین با استفاده از الگوریتم‌هایی که در این درس آموختیم، در هر کامپایلر عبارت منظمی تبدیل به اتوماتای متناهی شده و این اتوماتا روی رشته‌هایی از برنامه اجرا می‌شود تا تطابق آن‌ها با عبارت منظم اولیه سنجیده شود. با دانستن این کارکرد، در این پروژه می‌خواهیم پیاده‌سازی این الگوریتم‌ها و شبیه‌سازی اتوماتای متناهی را تمرین کنیم.

۲ شرح پروژه

در این پروژه، باید برنامه‌ای بنویسید که با ورودی گرفتن یک اتوماتای متناهی غیرقطعی^{۱۱} و تعدادی رشته، عضویت یا عدم عضویت هر یک از رشته‌ها را در زبان اتوماتای داده شده، مشخص کند. در واقع، این پروژه از دو بخش اصلی تشکیل شده که در ادامه هر یک را توضیح می‌دهیم.

^۱ Compile

^۲ Finite Automata

^۳ Lexical Analyzer

^۴ Keyword (while, if, ...)

^۵ Identifier (a, student, ...)

^۶ Number (2, 345.37, ...)

^۷ String ("abcbb", ...)

^۸ Comment (//input, ...)

^۹ Regular Expression

^{۱۰} Scanner

^{۱۱} Nondeterministic Finite Automata

۱.۲ تبدیل اتوماتای متناهی غیر قطعی به قطعی

ابتدا با استفاده از الگوریتم گفته شده در قضیه ۱۷.۳ کتاب [۱]، Λ -انتقال‌های^{۱۲} اتوماتای ورودی را حذف کنید تا اتوماتای متناهی غیر قطعی و بدون Λ -انتقال معادل با آن به دست آید. سپس با استفاده از الگوریتم گفته شده در قضیه ۱۸.۳ کتاب [۱]، اتوماتای متناهی معادل بدون Λ -انتقال را به یک اتوماتای متناهی قطعی^{۱۳} تبدیل کنید.

۲.۲ شبیه‌سازی اتوماتای متناهی قطعی

در این مرحله، اتوماتای متناهی قطعی به دست آمده را برای هر یک از رشته‌های داده شده اجرا کنید و اگر رشته عضو زبان بود، خروجی Yes و در غیر این صورت، خروجی No چاپ کنید.

۳.۲ ورودی

می‌دانیم هر اتوماتای متناهی به صورت $M = (Q, \Sigma, q_0, A, \delta)$ تعریف می‌شود. بنابراین سطر اول ورودی شامل پنج عدد صحیح q, s, a, m و n خواهد بود که به ترتیب بیانگر تعداد حالت‌های^{۱۴} اتوماتا $(|Q| = q)$ ، تعداد حروف الفبای زبان $(|\Sigma| = s)$ ، تعداد حالت‌های پذیرش^{۱۵} $(|A| = a)$ ، تعداد انتقال‌های اتوماتا $(|\delta| = m)$ و تعداد رشته‌های مورد پرسش هستند. تمامی مقادیر بین ۱ تا ۱۰۵ قرار دارند.

حالت‌های اتوماتا به طور پیش‌فرض با اعداد ۰ تا $q - 1$ نام‌گذاری شده و در ورودی نمی‌آیند. در هر یک از s سطر بعدی، یک نماد^{۱۶} به عنوان یک حرف الفبا داده می‌شود. در سطر بعد، حالت ابتدایی^{۱۷} (q_0) و در a سطر بعدی، حالت‌های پذیرش مشخص می‌شوند.

سپس در هر یک از m خط بعدی، عدد صحیح q_i ، نماد σ و عدد صحیح q_j با فاصله از هم می‌آیند که نشان می‌دهد $\delta(q_i, \sigma) = q_j$. همچنین تضمین می‌شود که $0 \leq q_i, q_j \leq q - 1$ و $\sigma \in \Sigma$ است.

در نهایت در هر یک از n سطر پایانی نیز یک رشته $x \in \Sigma^*$ داده می‌شود. اگر تنها بخش اول پروژه را پیاده‌سازی می‌کنید، مقدار n و n سطر نهایی از ورودی حذف خواهد شد. همچنین اگر تنها بخش دوم پروژه را پیاده‌سازی می‌کنید، تضمین می‌شود اتوماتای ورودی، به صورت قطعی خواهد بود.

^{۱۲} Λ -transition

^{۱۳} Deterministic Finite Automata

^{۱۴} State

^{۱۵} Accepting State

^{۱۶} Character

^{۱۷} Initial State

5 2 1 6 2

a

b

0

4

0 a 1

1 a 0

0 b 4

0 a 2

2 a 3

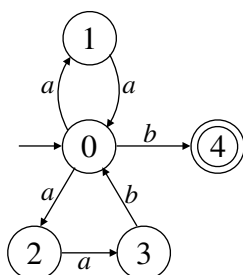
3 b 0

aaa

aab

توضیح

اتوماتای متناهی غیر قطعی داده شده در مثال بالا، به شکل زیر است.



۴.۲ خروجی

خروجی باید شامل n سطر باشد که سطر i -ام آن Yes را نشان می‌دهد اگر رشته i -ام توسط اتوماتا پذیرش شود و در غیر این صورت، No را چاپ می‌کند.

اگر تنها بخش دوم پروژه را پیاده‌سازی می‌کنید نیز خروجی به همین صورت خواهد بود ولی اگر فقط بخش اول را درست می‌کنید، در خروجی باید تعریف اتوماتای قطعی به دست آمده را چاپ کنید؛ یعنی در سطر اول، اعداد صحیح q ، a که به ترتیب بیانگر تعداد حالت‌های اتوماتا ($|Q| = q$) و تعداد حالت‌های پذیرش ($|A| = a$) هستند.

حالت‌های اتوماتا به طور پیش فرض با اعداد 0 تا $q - 1$ نام‌گذاری شده و لازم نیست در خروجی بیایند. در سطر بعد، حالت ابتدایی^{۱۸} (q_0) و در a سطر بعدی، حالت‌های پذیرش مشخص شوند. سپس در هر یک از q خط بعدی، عدد صحیح q_i ، نماد σ و عدد صحیح q_j با فاصله از هم بیایند که نشان دهد $\delta(q_i, \sigma) = q_j$.

^{۱۸} Initial State

مثال

No
Yes

توضیح

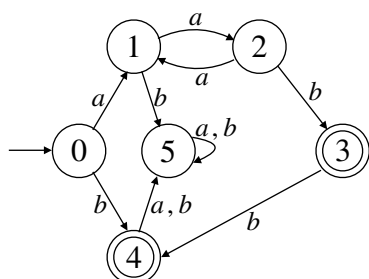
خروجی بالا مربوط به پیاده‌سازی هر دو بخش پروژه است.

مثال

6 2
0
3
4
0 a 1
0 b 4
1 a 2
1 b 5
2 a 1
2 b 3
3 a 1
3 b 4
4 a 5
4 b 5
5 a 5
5 b 5

توضیح

خروجی بالا مربوط به پیاده‌سازی تنها بخش اول پروژه است و اتوماتای متناهی قطعی به دست آمده‌ی آن، به شکل زیر خواهد بود.



۳ امتیاز

این پروژه مجموعاً ۲ نمره دارد که ۱ نمره‌ی آن مربوط به بخش اول، یعنی تبدیل اتوماتای غیرقطعی به قطعی و ۱ نمره‌ی آن مربوط به بخش دوم، یعنی شبیه‌سازی اتوماتای متناهی است. شما می‌توانید هر یک از بخش‌ها را که بخواهید یا هر دو را پیاده‌سازی کنید.

۴ نحوه‌ی تحویل

برنامه را با یکی از زبان‌های برنامه‌نویسی C، C++، Java یا Python بنویسید. استفاده از هر کتاب‌خانه‌ای^{۱۹} که به طور مستقیم مسئله را حل نکند، مانعی ندارد. برای نوشتن کدهای خود می‌توانید از هر مرجعی یا هوش مصنوعی کمک بگیرید ولی منابع استفاده شده‌ی خود را مشخص کنید. همچنین باید توانایی ارائه‌ی شفاهی و توضیح قسمت‌های مختلف کد خود را داشته باشید. پروژه به صورت فردی است و کپی کردن کدهای دیگر دانشجویان کلاس مجاز نیست.

فایل کامل پروژه‌ی خود را در قسمت مشخص شده در کورسز درس بارگذاری کنید. اگر نکات ویژه‌ای برای اجرای برنامه‌ی شما باید رعایت شود، مانند نصب کردن پکیج خاصی یا قرار دادن فایل در مسیر^{۲۰} مشخصی و ...، آن را در فایل جداگانه‌ای نوشته و همراه با پروژه‌ی خود بارگذاری نمایید. این پروژه به صورت اتوماتیک داوری نمی‌شود و لازم است برای ارائه‌ی شفاهی پروژه‌ی خود آماده باشید.

۵ مهلت ارسال

فایل نهایی پروژه‌ی خود را تا ساعت ۲۳:۵۹ روز شنبه ۳ تیر بارگذاری نمایید. ارائه‌ی شفاهی ۱۰ دقیقه‌ای پروژه روز شنبه ۴ تیر طبق جدول زمان‌بندی که متعاقباً پر خواهید کرد، انجام می‌شود. به علت نزدیکی مهلت ثبت قطعی نمرات در پرتال، امکان تمدید وجود نخواهد داشت.

۶ منابع

[1] Martin, J. C. (2003). *Introduction to Languages and the Theory of Computation Fourth Edition*. NY: McGraw-Hill.

^{۱۹} Library

^{۲۰} Directory