



نظریه زبان ها و ماشین ها

دکتر خواسته

پروژه نهایی

پروژه نهایی درس نظریه زبان ها و ماشین ها:

لطفا قبل از انتخاب موضوع به نکات زیر توجه کنید.

پروژه ها به صورت انفرادی یا در گروه های دوفره باید پیاده سازی شوند.

محدودیتی برای انتخاب زبان برنامه نویسی مورد استفاده وجود ندارد.

گزارش کار پروژه های نهایی همراه با پیاده سازی پروژه های مورد نظر ارسال گردد که شامل موارد زیر باشد:

1. نام و شماره دانشجویی اعضای گروه

2. شماره ی پروژه ی انتخابی

3. توضیح کامل الگوریتم

4. توضیح کلی در مورد کارکرد کد

5. نحوه ی اجرا و ران گرفتن کدها

6. حداقل دو مثال از ورودی و خروجی برنامه

مهلت ارسال پروژه حداکثر تا پنج بهمن خواهد بود و تمدید نمی شود .

اعضای هر گروه نیز تا تاریخ مشخص شده ، فایل زیپ مربوط به پاسخ پروژه خود را در کوئرا آپلود کنند.

پروژه شماره ی (1)

موضوع: تبدیلگر NFA به Regex (Regular Expression)

توضیحات: برنامه ی شما یک NFA را از کاربر دریافت کرده و از روی آن Regex ای تولید می کند که رشته های پذیرفته شده در آن NFA در آن عبارت منظم وجود داشته باشد و آن را به صورت دلخواه نمایش دهد.

پروژه شماره ی (2)

موضوع: تبدیلگر NFA به DFA

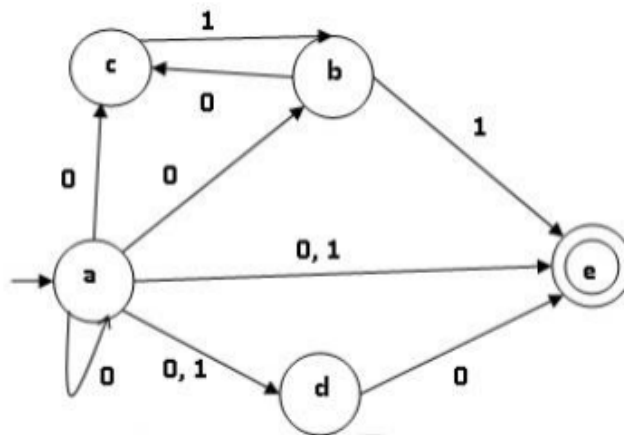
توضیحات: برنامه شما یک NFA را به فرم دلخواه از کاربر یا فایل خوانده و آن را به یک DFA تبدیل میکند و آن را به فرم دلخواه نمایش میدهد.

پروژه شماره ی (3)

موضوع: وجود یک رشته در یک NFA

توضیحات: برنامه ی شما یک NFA و یک رشته را از کاربر دریافت کرده و از روی آن NFA چک کند که آیا آن رشته در NFA داده شده Accept میشود یا نمیشود.

نکته: تشکیل درخت اشتقاق و نمایش تمام حالت ها اجباری است. برای مثال:



رشته 0001 در رشته Accept می شود و رشته 1111 پذیرفته نمی شود.

پروژه شماره ی (4)

موضوع: تبدیلگر یک CFG به فرم نرمال چامسکی

توضیحات: برنامه ی شما یک CFG و را از کاربر دریافت کرده آن CFG را به فرم نرمال چامسکی تبدیل کند و به صورت دلخواه نمایش دهد.

برای مشاهده مثال و الگوریتم تبدیل به فرم چامسکی سایت زیر را ببینید:

<https://www.geeksforgeeks.org/convertng-context-free-grammar-chomsky-normal-form/>

پروژه شماره ی (5)

موضوع: تبدیلگر یک CFG به PDA (PDA معادل CFG)

توضیحات: برنامه ی شما یک CFG و را از کاربر دریافت کرده آن CFG را به PDA تبدیل کند و به صورت دلخواه نمایش دهد.

برای مشاهده مثال و الگوریتم تبدیل CFG به PDA سایت زیر را ببینید:

https://www.tutorialspoint.com/automata_theory/pda_context_free_grammar.htm

پروژه شماره ی (6)

موضوع: درخت پارس عبارات شرطی تو در تو

توضیحات: در این پروژه شما ابتدا باید CFG ای طراحی کنید که میتواند تمام حالات شرط های تو در تو با تعداد عمق نامتناهی را تولید کند و درخت اشتقاق این CFG به گونه ای باشد که زیر شرط هر شرط در کد، زیر درخت گره متناظر با آن شرط در درخت اشتقاق باشد. پس از طراحی CFG مورد نظر، برنامه شما باید به گونه ای باشد که یک کد به زبان برنامه نویسی دلخواه که از شرط های تو در تو تشکیل شده است را از کاربر یا فایل دریافت کرده و درخت اشتقاق آن را نمایش دهد.

پروژه شماره ی (7)

موضوع: درخت پارس عبارات منطقی and و or و پرانتز

توضیحات: در این پروژه شما ابتدا باید CFG ای طراحی کنید که میتواند تمام عبارات منطقی شامل and ، or و پرانتز را تولید کند و درخت اشتقاق این CFG به گونه ای باشد که با استفاده از آن بتوان اعمال منطقی موجود در هر عبارت منطقی را با ترتیب درست انجام داد. پس از طراحی CFG مورد نظر، برنامه شما باید به گونه ای باشد یک عبارت منطقی از کاربر دریافت کرده و درخت اشتقاق آن عبارت منطقی را نمایش دهد.

پروژه شماره ی (8)

موضوع: وجود یک رشته در یک ماشین تورینگ

توضیحات: برنامه ی شما یک ماشین تورینگ و یک رشته را از کاربر دریافت کرده و از روی آن ماشین تورینگ چک کند که آیا آن رشته در ماشین تورینگ داده شده **Accept** میشود یا نمیشود.

پروژه شماره ی (9)

موضوع: درخت پارس عبارات محاسباتی شامل جمع و ضرب و پرانتز

توضیحات: در این پروژه ابتدا باید CFG ای طراحی کنید که می تواند تمام عبارات ریاضی شامل ضرب، جمع، پرانتز را تولید کند و درخت اشتقاق این CFG به گونه ای است که با استفاده از آن بتوان اعمال ریاضی موجود در هر عبارت ریاضی را با ترتیب درست انجام داد. پس از طراحی CFG ، برنامه شما باید یک عبارت ریاضی از کاربر دریافت کرده و درخت اشتقاق ریاضی را نمایش دهد.

برای توضیحات سایت زیر را مطالعه کنید:

https://en.wikipedia.org/wiki/Parse_tree

پروژه شماره ی (10)

موضوع: پیاده سازی الگوریتم CYK

توضیحات: برنامه ی شما یک CFG و یک رشته را از کاربر دریافت کرده و در آن CFG چک می کند که آیا آن CFG رشته مورد نظر را تولید می کند یا خیر.

برای مثال:

: ورودی

$$S \rightarrow aSb|\epsilon$$

$$w = abbb$$

: خروجی

Not Accept

نکته: ورودی ممکن است به فرم نرمال چامسکی نباشد.