اعضای گروه: مهدی طاهری – مریم شیرازی

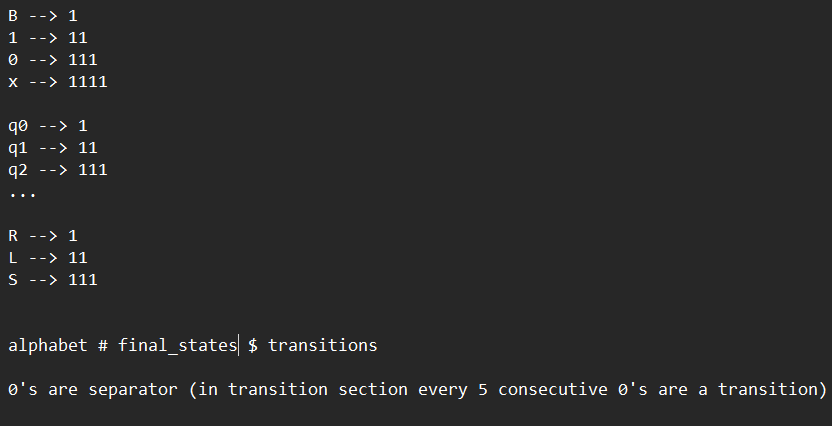
# شرح پروژه

در این پروژه یک ماشین تورینگ عمومی طراحی میکنیم که ماشین های تورینگ دیگر را میگیرد و نتیجه ی محاسبه و پذیرش یا عدم پذیرش( در صورت نامعتبر بودن رشته ورودی) را میدهد.

ماشین تورینگ عمومی سه نوار دارد که روی یکی از ان ها 1) حالت کنونی ماشین تورینگ ورودی گرفته شده ، روی دیگری 2) رشته­ی ورودی و روی دیگری 3) توصیف ماشین ورودی

# تبدیل فایل به فرمت مناسب نوار ها

برای نوار توصیف ماشین ابتدا سمبل های الفبا و بعد از ان یک علامت # و سپس حالت های پذیرش نوشته به صورت یونری جدا شده با صفر نمایش داده میشود سپس علامت $ و در نهایت قوانین ترنزیشن ماشین تورینگ داده شده به صورت یونری می اید. مثال: B10x#111011110$10110111101011101110



# نحوه عملکرد

قابل توجه است که به کمک توابع set در ابتدا نوار ها را set میکنیم به عنوان مثال نوار حالت را در ابتدا روی Start state میگذاریم و در ادامه ی کار تغییر میکند و اپدیت میشود. سپس به کمک تابع start برنامه ماشین تورینگ عمومی شروع به کار میکند.

فرض کنیم حالتqi از روی نوار مربوطه خوانده شده و همین طور head نوار ورودی روی 1 باشد ماشین تورینگ عمومی به کمک واحد کنترل خود جستجو میکند که ایا حالت qi با ورودی 1 در بین قوانین وجود دارد یا خیر. اگر وجود نداشت که هالت میکند اگر قانونی وجود داشت حالت جدید اپدیت میشود و همچنین بر اساس همان قانون به جای ورودی خوانده شده نیز کاراکتر جدید قرار میگیرد و head نوار ها یکی به چپ یا راست میرود (اابته اگر در دستورs به معنای stay داشته باشیم head روی نوار ورودی ها ثابت میماند)

انقدر ادامه میدهیم تا ماشین هالت کند اگر روی حالت پذیرش هالت کند رشته پذیرش شده و نتیجه ی محاسبه چاپ میشود در غیر این صورت رشته پذیرش نمیشود یا توی لوپ می­افتد.

# نمونه فایل های ورودی

Adding machine

alphabet:{B , 1, 0}

start\_state:{q0}

states:{q0 , q1 , q2, q3 , q4 }

final\_states:{q4}

actions:{(q0 , 1,1,R,q0) , (q0 , 0,1,R,q1) , (q1, 1,1,R, q1) ,(q1 , B,B,L,q2) , (q2,1,B,L,q3) , (q3, 1,1,L,q3) , (q3 , B,B,R,q4)}

Multiply machine

alphabet:{B ,0,1,x,y}

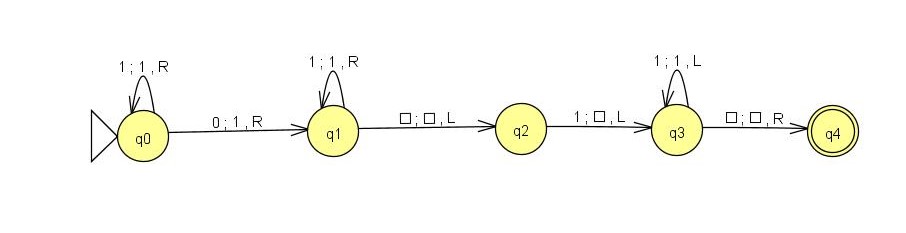
start\_state:{q0}

states:{q0 , q1 , q2, q3 , q4 ,q5 ,q6,q7 ,q8,q9,q10,q11,q12,q13}

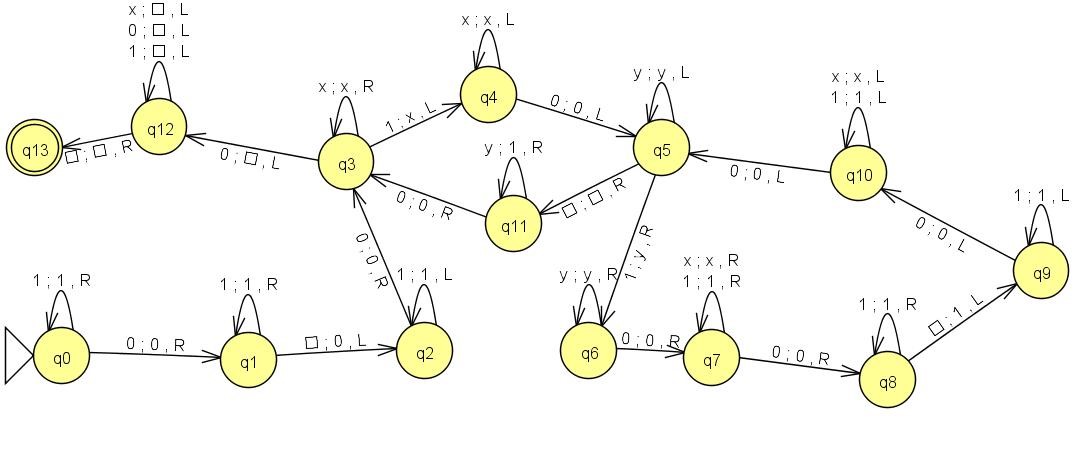
final\_states:{q12,q13}

actions:{(q0,1,1,R,q0) ,(q0,0,0,R,q1),(q1,1,1,R,q1) , (q1,B,0,L,q2) , (q2,1,1,L,q2) ,(q2,0,0,R,q3) ,(q3,x,x,R,q3),(q3,1,x,L,q4) ,(q4,x,x,L,q4),(q4,0,0,L,q5) ,(q5,y,y,L,q5) ,(q5, 1,y,R,q6) ,(q6,y,y,R,q6),(q6,0,0,R,q7),(q7,1,1,R,q7),(q7,x,x,R,q7),(q7,0,0,R,q8),(q8,1,1,R,q8),(q8,B,1,L,q9),(q9,1,1,L,q9),(q9,0,0,L,q10),(q10,1,1,L,q10) ,(q10,x,x,L,q10),(q10,0,0,L,q5),(q5,B,B,R,q11),(q11,y,1,R,q11),(q11,0,0,R,q3),(q3,0,B,L,q12),(q12,1,B,L,q12),(q12,0,B,L,q12),(q12,x,B,L,q12),(q12,B,B,R,q13)}

Computation(x + y) turing machine:



Computation(x \* y) turing machine:



برای ماشین جمع کننده فرض کنید ما میخواستیم عدد 3 را با عدد چهار جمع کنیم:

پس رشته ی نوار محتوا B11101111B بوده است اگرhead روی 1 چهارمی باشد headنوار حالت روی q1خواهد بود و transition پیدا شده (q1, 1,1,R, q1)خواهد بود در نتیجه head نوار محتوا یکی به راست میرود نوار حالت روی حالت q1 میماند و ادامه....

برای ماشین ضرب کننده نیز به همین روال ...