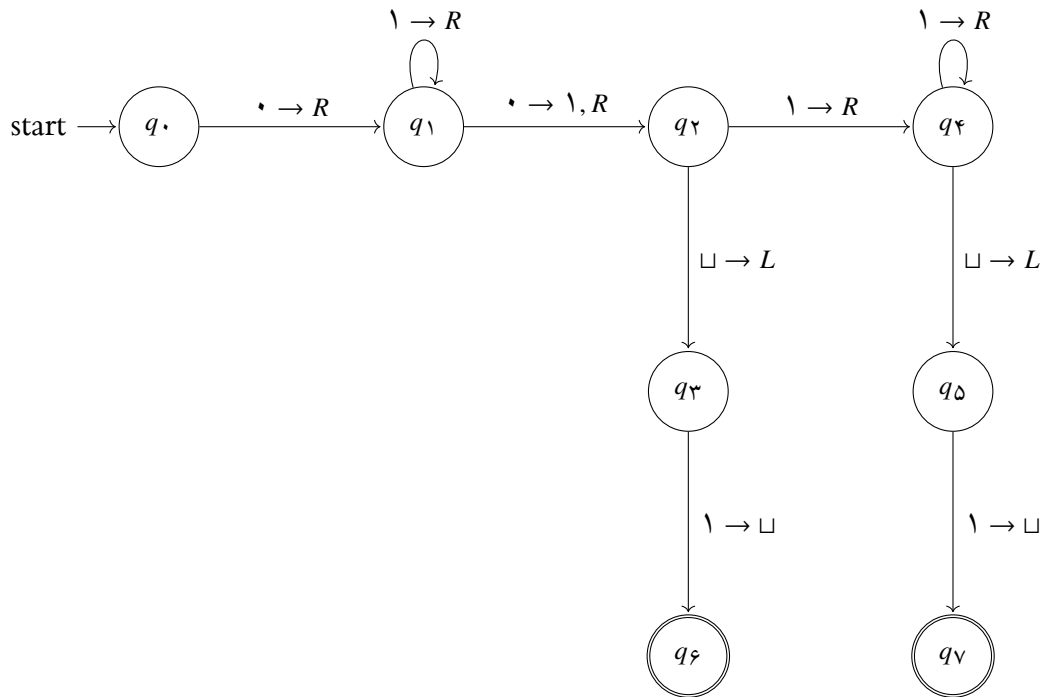


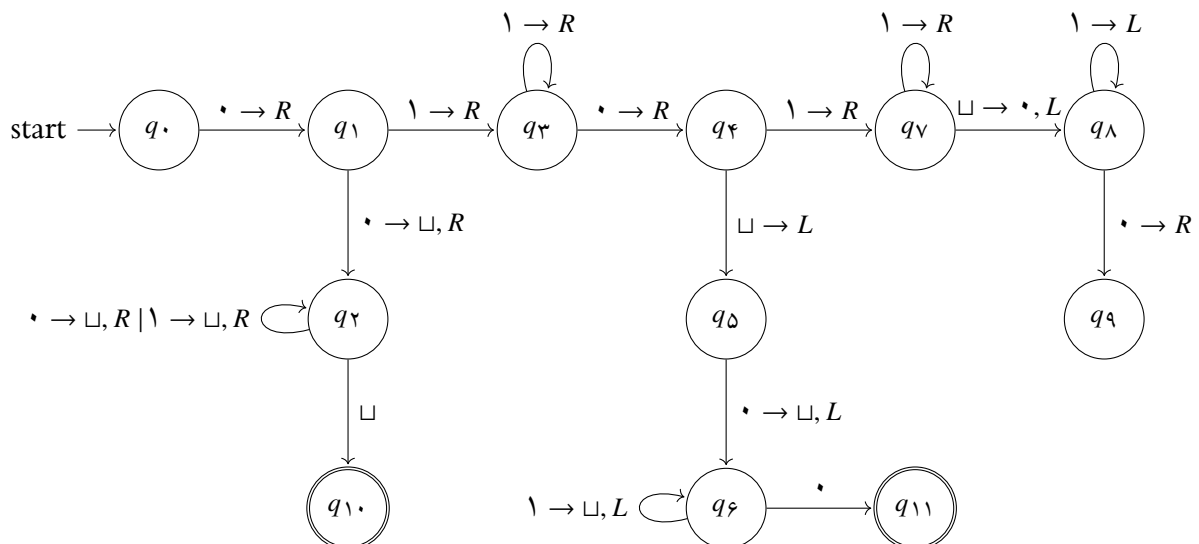
سوال ۱. تمام ۱های عدد اول را رد کرده و • بین دو عدد را تبدیل به ۱ میکنیم. اگر عدد دوم • باشد به علامت خالی (□) در نوار میرسیم. و ۱ نوشته شده را پاک میکنیم. اگر عدد دوم • نباشد پس تعدادی ۱ دارد. تا آخر حرکت میکنیم و آخرین ۱ را به جای خالی (□) تبدیل میکنیم.



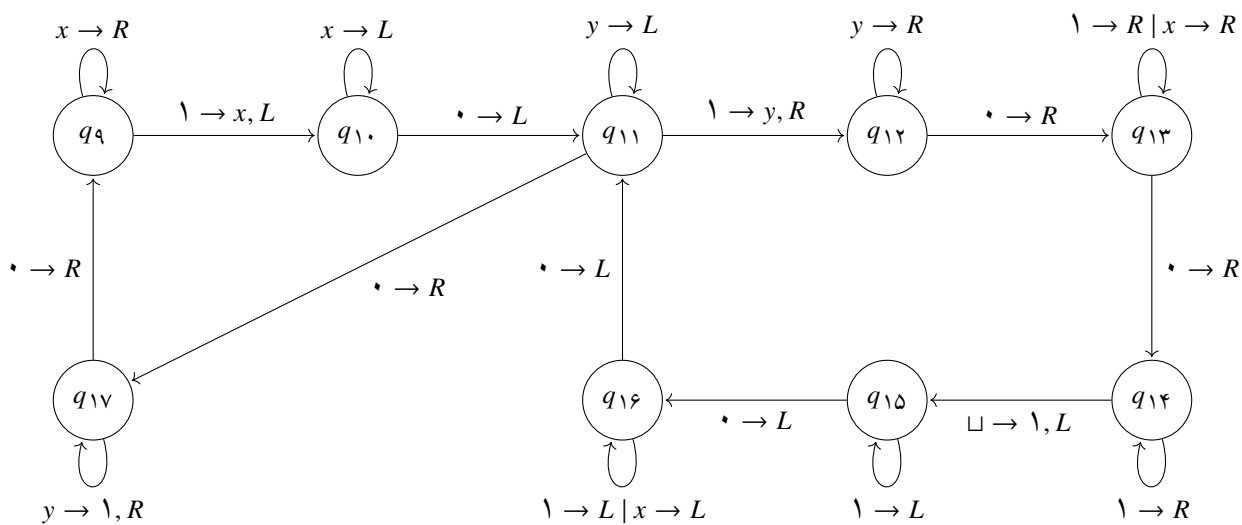
سوال ۲. ماشین از سه بخش تشکیل میشود:

۱. بررسی حالت صفر بودن ورودی‌ها
۲. محاسبه حاصل ضرب و نوشتن آنها در انتهای نوار
۳. انتقال جواب از انتها به ابتدای نوار

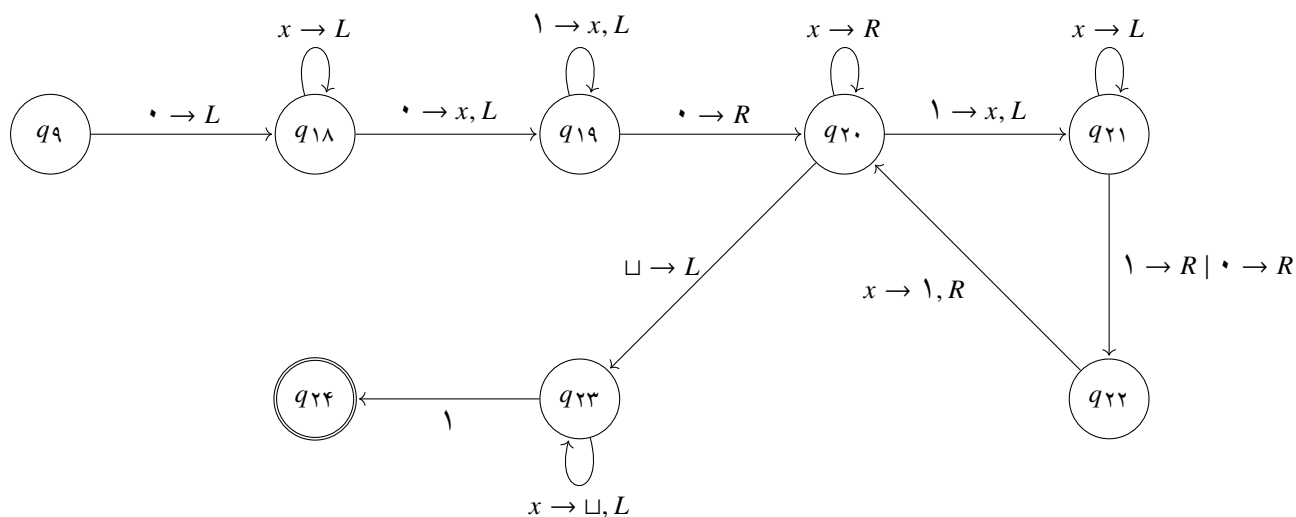
• بخش ۱:



• بخش ۲:

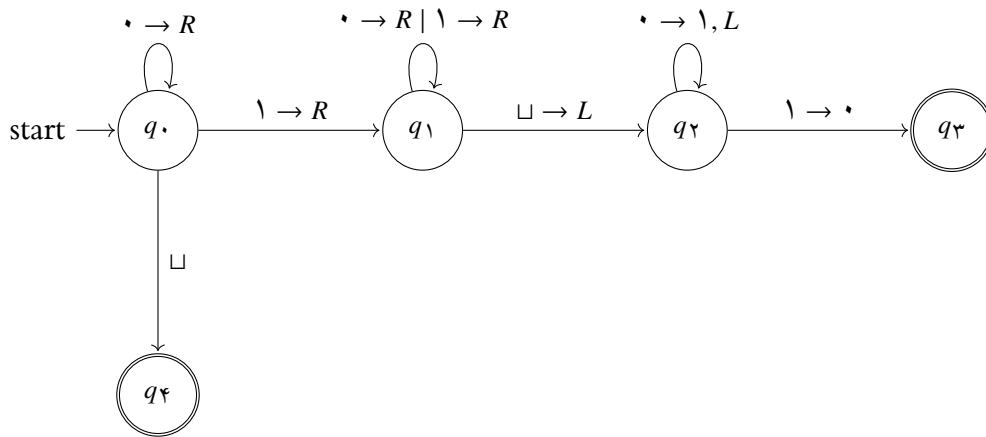


• بخش ۳:



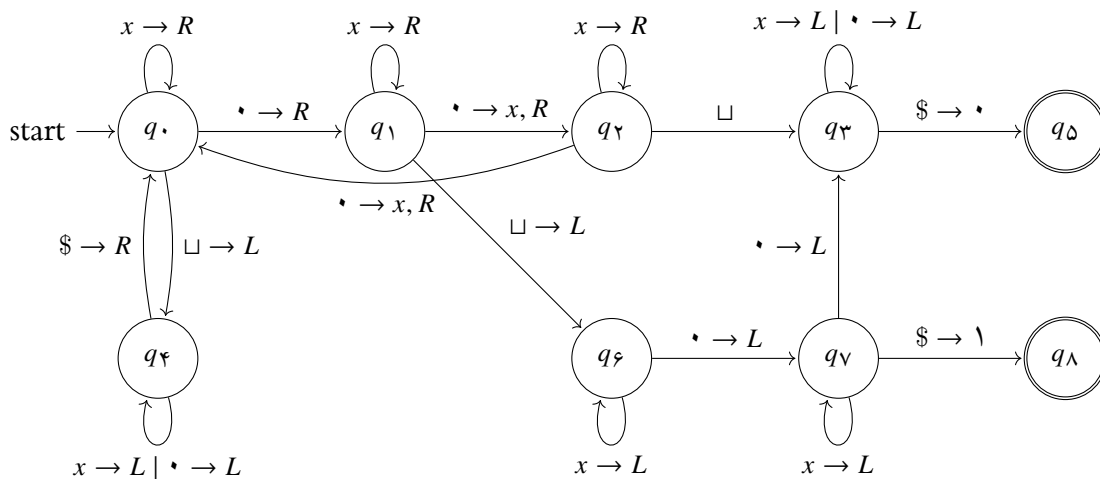
ابتدا بررسی میکنیم که هیچ یک از ورودی‌ها صفر نباشد وگرنه صفر خروجی میدهیم. سپس در انتهای رشته صفر میگذاریم و روی ورودی دوم حرکت میکنیم و به ازای هر ۱، یک کپی از ورودی اول به انتهای رشته اضافه میکنیم. و این کپی را نیز ۱ هایش را یکی یکی انتقال میدهیم. پس از اتمام، کل بخش ورودی نوار را تبدیل به بلوک x میکنیم. و سپس مرحله به مرحله آنرا به انتهای رشته نزدیک میکنیم تا در نهایت از کل یکها گذر کند و سپس آنرا پاک میکنیم.

سوال ۳. فرض کنید ورودی، نمایش دودویی عدد باشد و بعد از آن \square باشد.



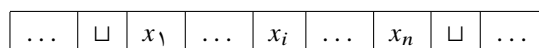
اگر ورودی تعدادی صفر باشد همان ورودی را خروجی میدهد (q_4). اگر ورودی ناصفر باشد تا انتهای ورودی می‌رود و سپس برعکس روی ورودی حرکت میکند تا به یک بیت ۱ برسیم. در این مسیر برعکس همه‌ی بیت‌ها را سوییچ میکنیم. (q_3)

سوال ۴. فرض میکنیم ابتدای ورودی \$ باشد. اگر ورودی معتبر باشد خانه اول تبدیل به ۱ و در غیراینصورت تبدیل به صفر میشود.



روی ورودی حرکت میکنیم و از بین هر سه تا ۰ دوتای آنها را تبدیل به x میکنیم. اگر تعداد صفرها مضرب سه بود پس از اتمام دوباره به اول نوار برمیگردیم و دوباره این فرآیند را تکرار میکنیم. وگرنه ریجکت میکنیم. اگر ورودی 0^{3^n} باشد پس از الگوریتم بالا به دقیقاً یک صفر میرسد که اکسپت میشود.

سوال ۵. کافیس‌ت رفتار ماشین تورینگ روی نوار را با این ماشین شبیه‌سازی کنیم. فرض کنید نوار ماشین تورینگ به فرم زیر باشد به طوری که head روی خانه x_i باشد:



آنگاه صف به شکل زیر باشد:

Queue Head

x_i
x_{i-1}
\vdots
\$
\vdots
x_{i+1}

حال دو حالت داریم:

۱. اگر دستور $x_i \rightarrow A, L$ باشد داریم:

۱. pop

۲. push A

همین عمل را روی صف عملی میکند و داریم:

x_{i-1}
\vdots
\$
\vdots
x_{i+1}
A

تنها حالتی مشکل ساز است که بعد از این عملیات، علامت \$ روی ابتدای صف قرار بگیرد. این یعنی روی نوار تورینگ از ورودی رد شده ایم و به \sqcup رسیده ایم. در اینصورت ابتدا \$ را پاپ کرده و سپس به ترتیب #, \sqcup , \$ را پوش میکنیم. سپس یکبار تمام اعضای روی صف را پاپ کرده و بلافاصله پوش میکنیم تا به # برسیم. و با پاپ کردن آن به مرحله مورد نظر در تورینگ رسیده ایم.

۲. اگر دستور $x_i \rightarrow A, R$ باشد داریم:

۱. pop x_i

۲. push #

۳. push A

سپس وارد یک زیرماشین با های state به نام الفبای نوار تورینگ میشویم. به طوریکه اگر در state یک حرف باشیم با خواندن حرف بعدی، ابتدا حرف state فعلی را پوش میکنیم. سپس حرف خوانده شده را پاپ میکنیم و به state حرف خوانده شده میرویم. هنگامی که به # رسیدیم ابتدا # را پاپ میکنیم و بلافاصله # را پوش میکنیم. سپس حرف state فعلی را پوش میکنیم و از زیرماشین خارج میشویم. سپس مثل بخش قبلی یکدور تمام حروف صف را پاپ و بلافاصله پوش میکنیم تا به # برسیم. # را که پاپ کنیم به موقعیت ماشین تورینگ بعد از دستور ذکر شده میرسیم. اما مانند بخش قبلی اگر بعد از انجام مراحل در head استک مقدار \$ باشد یعنی از ورودی رد شدیم. در این حالت \$ را پاپ میکنیم و بلافاصله پوش میکنیم و بعد یک \sqcup پوش میکنیم. سپس یکدور تمام اعضای صف را پاپ و بلافاصله پوش میکنیم تا دوباره به \$ برسیم. آنگاه اگر \$ را نیز پاپ و پوش کنیم به \sqcup در ابتدای استک میرسیم.

پس هر دو نوع عمل روی نوار ماشین تورینگ شبیه سازی شد و ثابت میشود که هر ماشین تورینگ را میتوان با این ماشین نیز شبیه سازی کرد.