

سوال ۲ - امتیازی

(۱) تعریف صوری

خودکاره مشخصه را با یک ۴-tuple به شکل  $(Q, \Sigma, n, \delta, q, P)$  شش می‌دهیم که

\*  $Q$  : مجموعه متناهی از حالات

\*  $\Sigma$  : الفبا

\*  $n$  : تعداد شمارنده

\*  $\delta$  : تابع انتقال  $\Leftarrow$

$$\delta: Q \times (\Sigma \cup \{z, nz, dc\})^n \rightarrow P(Q \times \{L, R\} \times \{-1, 0, 1\})^n$$

$z$ : zero,  $nz$ : not zero,  $dc$ : don't care

\*  $q_0$  : است آغازین

\*  $F$  : حالات پذیرش  $\Leftarrow F \subseteq Q$

این ماشین را در حالت کلی تر و غیر قطعی تعریف می‌کنیم.

با خواندن کاراکتری در اشاره‌گر روی آن است و همچنین جابجایی  $z$  یا  $nz$  یا  $dc$ ، به حالت جدید

رفته و اشاره‌گر را به  $R$  یا  $L$  می‌دهیم (چون می‌دهیم) و شمارنده را  $+$ ،  $-$  یا بدون تغییر

می‌گذاریم.



باید برای هر خودکار پشته‌ای دلخواه، یک خودکار ۲-شماره بایز.

به پشته به عنوان رشته‌ای از بیت‌ها نگاه می‌کنیم. هرکارتر از ۳ را یک عدد باینری

در نظر می‌گیریم. در این حالت  $pop$  و  $push$  و  $rem$  متد،  $pop$  را بدین شکل تعریف می‌کنیم:

بدوشتن ابیت افزودن به پشته  $pop =$

اصاح کردن ابیت به ترتیب  $push =$

حال خودکار ۲-شماره‌ای را در نظر بگیرید که  $C_1$  و  $C_2$  شماره‌های آن می‌باشند. اگر

$w_1, w_2, \dots, w_k$  در پشته باشند، می‌خواهم برای  $C_1$  (یا  $C_2$ ) مقدار  $(w_1)(w_2)(w_3) \dots (w_k)$

را نذر دارد که  $(w_k)$  غشیش باینری کارتر  $w$  است.

$pop() \Leftarrow$

$push(s) \Leftarrow$

$while (C_1 \neq 0):$

$C_1 --;$

$if (C_1 == 0): break$

$C_1 =, C_2 ++$

$while (C_1 \neq 0):$

$C_1 --;$

$C_2 ++;$

$C_2 ++;$

$if (s == 1):$

$C_2 ++$

حکار  $pop$  و  $push$  را اعمال کنیم، وظیفه ۲-شماره با هم عوض می‌شود. یعنی هر عمل  $push$

یا  $pop$  برای خودکار پشته‌ای با دو شماره قابل پیاده‌سازی است. زیرا همواره یکی از آنها،

CLASS MEMO

محتویات پشته را نگه می‌دارد و دیگری، صفراست و گنگی خود است.

\* بعد عمل  $pop$  یا  $push$  با یکدیگر به یکدیگر تعلق ندارد، اگر  $C_1$  حامل مقدار پشته باشد و  $C_2$  حامل صفرا، به عنوان یک پشته می‌شود.



ج، در بخش قبل نشان داریم هر خودکاره شسته ای را می توان با خودکاره ۲ شمارنده، شبیه سازی کرد.

در این سوال باید نشان دهیم برای خودکاره ۲ شسته ای نیز این امکان وجود ندارد. به طور واضحی

این شبیه ساز با توجه به بخش قبل، با ۴ شمارنده قابل شبیه سازی است. در این بخش این مقدار را

ابتدا به ۳ و در نهایت به ۲ کاهش می دهیم.

اگر دقت کرده باشید، هنگامی که در باره تبدیل خودکاره شسته ای به ۲ شمارنده ای صحبت می کردیم، یک شمارنده

مخوای این شسته را نگه می داشت و دیگری، به پیاده ساز  $pop$  و  $push$  کمک می کرد. بنابراین

در اینجا هم ۲ شمارنده را برای ۲ شسته و یک شمارنده را برای  $pop$  و  $push$  در نظر می گیریم.

بنابراین ۳ شمارنده را تا اینجا نشان داریم. حال باید تعداد شمارنده ها را به ۲ برسانیم.

فرض کنید ۳ شمارنده قبلی، مقادیر  $z$  و  $k$  را داشته باشند. یک شمارنده فراموشی داریم تا عدد

$2^k z^k$  را نگه دارد. در این حالت اطلاعات ۳ شمارنده قبلی را داریم. یک شمارنده دیگر هم که

لگو است. در این حالت ۳ شمارنده را با ۲ شمارنده توان عمل می دهد. برای مثال برای ۲ داریم.

$x = z \cdot k$  به برسی می کنیم  $2^k z^k$  به ۳ بخش پذیر هست یا خیر.

$x$  ل  $increment$  و  $2^k z^k$  را در ۳ ضرب می کنیم.

پس به طور خلاصه، از بخش قبل شبیه می شد خودکاره ۲ شسته ای با ۴ شمارنده به راحتی قابل پیاده سازی است.

پس این را به ۳ شمارنده کاهش داریم. بعد از آن این ۳ شمارنده را به ۲ شمارنده کاهش می دهیم.

در نتیجه هر رایانه توانمند را می توانیم با خودکاره ۲ شمارنده شبیه سازی کنیم.