

mac.1 Estados de Parada

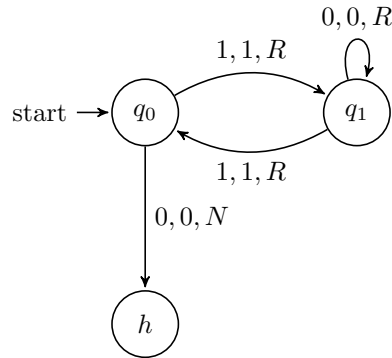
tur:mac:hal:
sec

Embora tenhamos definido nossas máquinas para parar apenas quando não há instrução a executar, representações comuns de máquinas de Turing têm um *estado de parada* dedicado h , de modo que $h \in Q$.

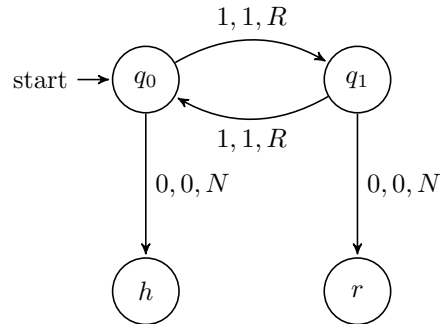
explanation

A ideia por trás de um estado de parada é simples: quando a máquina terminou a operação (está pronta para aceitar a entrada, ou terminou de escrever a saída), ela entra em um estado h onde para. Algumas máquinas têm dois estados de parada, um que aceita a entrada e outro que rejeita a entrada.

Exemplo mac.1. *Estados de parada.* Para elucidar esse conceito, comecemos com uma alteração da máquina dos pares. Em vez de fazer a máquina parar no estado q_0 se a entrada for par, podemos adicionar uma instrução para enviar a máquina a um estado de parada.



Vamos expandir ainda mais o exemplo. Quando a máquina determina que a entrada é ímpar, ela nunca para. Podemos alterar a máquina para incluir um estado de *rejeição* substituindo a instrução em laço por uma instrução para ir a um estado de rejeição r .



Adicionar um estado de parada dedicado pode ser vantajoso em casos como este, em que deixa explícito quando a máquina aceita/rejeita certas entradas. No entanto, é importante notar que nenhum poder computacional é ganho ao

explanation

adicionar um estado de parada dedicado. Da mesma forma, uma noção menos formal de parada tem suas próprias vantagens. A definição de parada usada até agora neste capítulo torna a prova do *Problema da Parada* intuitiva e fácil de demonstrar. Por essa razão, continuamos com nossa definição original.

Créditos das Imagens

Referências Bibliográficas