

Linguagens Formais e Autômatos

Aula 22 - Não-determinismo em Máquinas de Turing

Prof. Dr. Daniel Lucrédio
Departamento de Computação / UFSCar
Última revisão: ago/2015

Referências bibliográficas

- **Introdução à teoria dos autômatos, linguagens e computação / John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman** ; tradução da 2.ed. original de Vandenberg D. de Souza. - Rio de Janeiro : Elsevier, 2002 (Tradução de: Introduction to automata theory, languages, and computation - ISBN 85-352-1072-5)
 - Capítulo 8 - Seção 8.4
- **Introdução à teoria da computação / Michael Sipser** ; tradução técnica Ruy José Guerra Barretto de Queiroz ; revisão técnica Newton José Vieira. -- São Paulo : Thomson Learning, 2007 (Título original : Introduction to the theory of computation. "Tradução da segunda edição norte-americana" - ISBN 978-85-221-0499-4)
 - Capítulo 3 - Seção 3.2

Definição formal da função de transição de uma MT

- Função de transição
 - $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{E,D\}$
 - Os argumentos de $\delta(q,X)$ são:
 - um estado q e um símbolo de fita X
 - O valor de $\delta(q,X)$ (se definido) é uma tripla (p,Y,S) , onde:
 - p é o próximo estado em Q
 - Y é o símbolo, em Γ , gravado na célula que está sendo varrida, substituindo o símbolo que estava nessa célula
 - S é um sentido, ou direção, em que a cabeça se move
 - E = esquerda, D = direita
 - L = left, R = right

Não-determinismo em MTs

- Com base na definição do slide anterior, responda: “a definição de MTs que vimos até agora é determinística ou não-determinística?”
 - Resposta: determinística – a todo momento, sempre se sabe o que fazer
- Hum, e agora?
 - Em autômatos finitos, não-determinismo não aumenta a capacidade de reconhecimento de linguagens!
 - Em autômatos de pilha, não-determinismo aumenta a capacidade de reconhecimento de linguagens!
 - E em máquinas de Turing???

Não-determinismo em MTs

- Antes, veremos o que significa não-determinismo em MTs
- A única diferença está na função de transição:
 - $\delta: Q \times \Gamma \rightarrow \underline{\mathbf{P(Q \times \Gamma \times \{E,D\})}}$
 - Os argumentos de $\delta(q,X)$ são:
 - um estado q e um símbolo de fita X
 - O valor de $\delta(q,X)$ (se definido) é um conjunto de triplas (p,Y,S)

MTs não-determinísticas

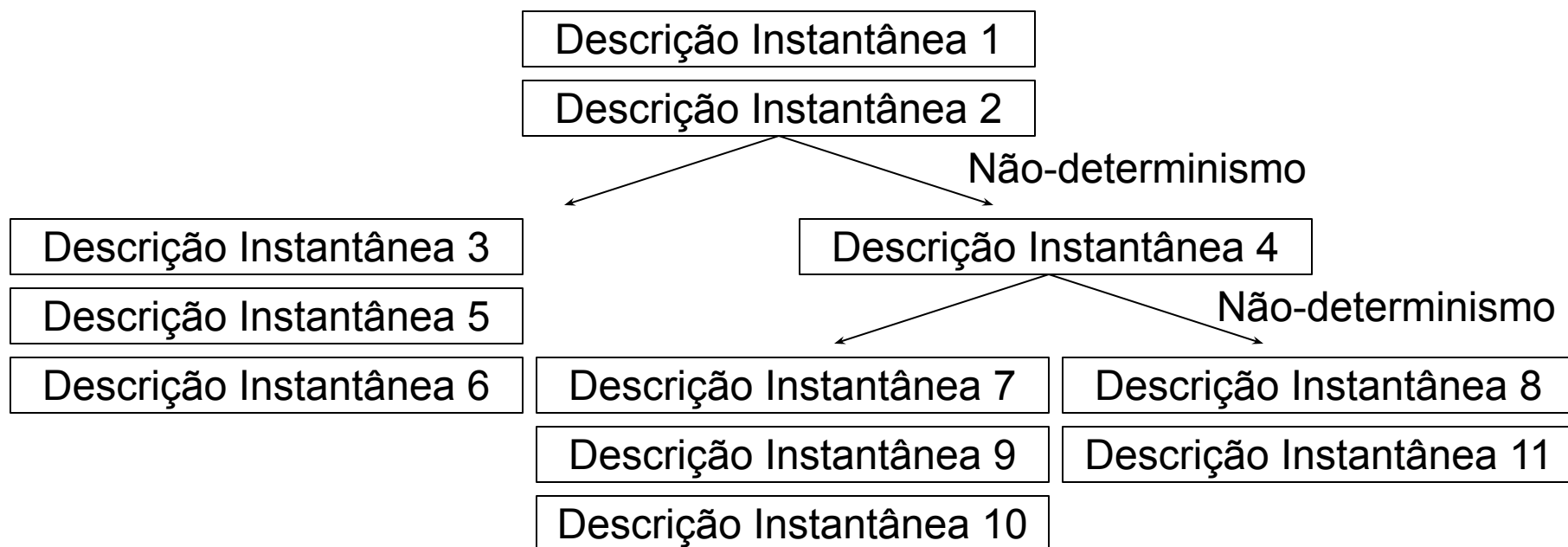
- A linguagem aceita por uma NTM (Nondeterministic Turing Machine) é definida de forma similar a outros autômatos
 - A cada ponto de dúvida, analisam-se todas as possibilidades, se nenhuma levar à aceitação, a cadeia não é aceita. Se ao menos uma levar à aceitação, a cadeia é aceita
- É possível provar que, para toda NTM que reconhece uma linguagem L , é possível construir uma DTM que reconheça L
 - Ou seja: NTMs e DTMs são equivalentes em termos de capacidade de reconhecimento de linguagens!!

Conversão NTM \rightarrow DTM

- A prova é uma construção:
 - Dada uma NTM, construiremos uma DTM que a simula, aceitando quando a NTM aceitar, e rejeitando quando a NTM rejeitar
- Muita atenção na construção a seguir
 - Ela possui implicações importantes na sua vida!!
 - Você pode ganhar um milhão de dólares se conseguir perceber uma coisa...
 - Você pode perder um milhão de dólares se alguém conseguir perceber essa mesma coisa...
 - ... se é que existe tal coisa!!

Conversão NTM → DTM

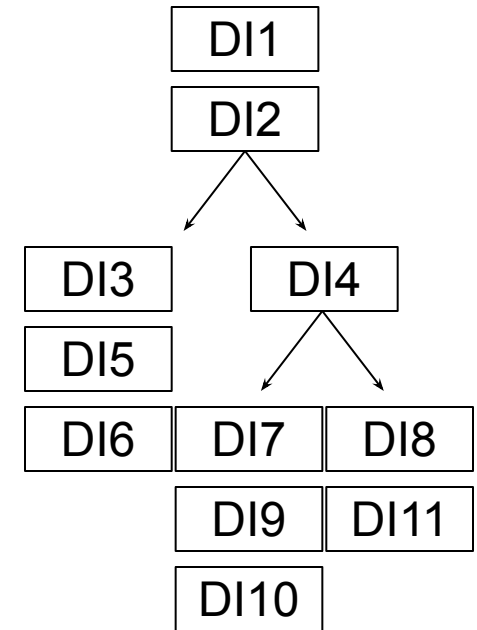
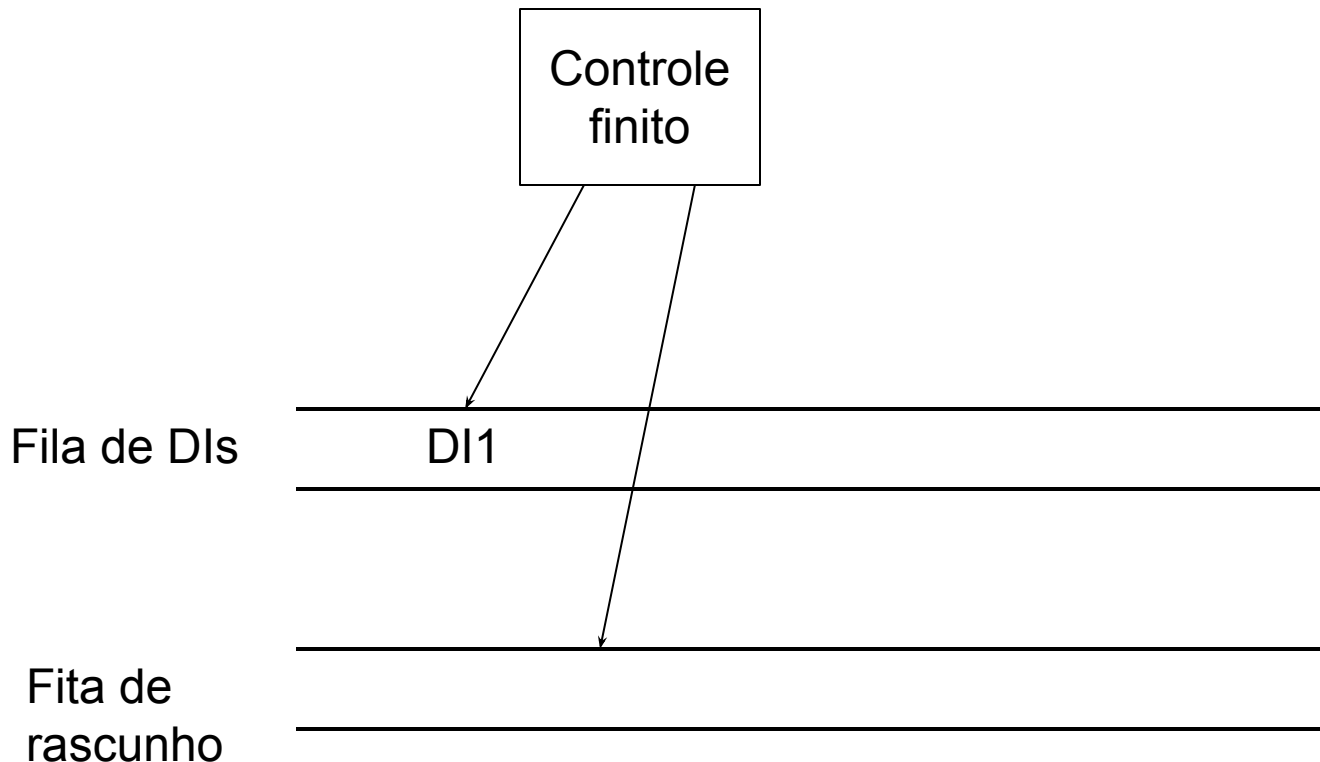
- A idéia é construir uma DTM que explora os diferentes pontos de não-determinismo, testando todas as possibilidades
- Como faríamos no papel?



Conversão NTM \rightarrow DTM

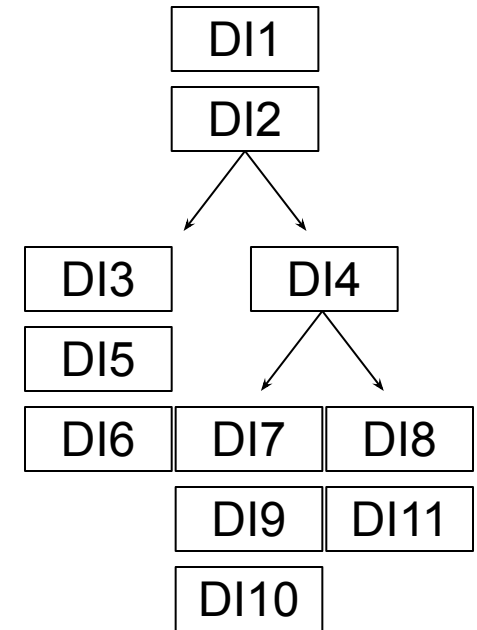
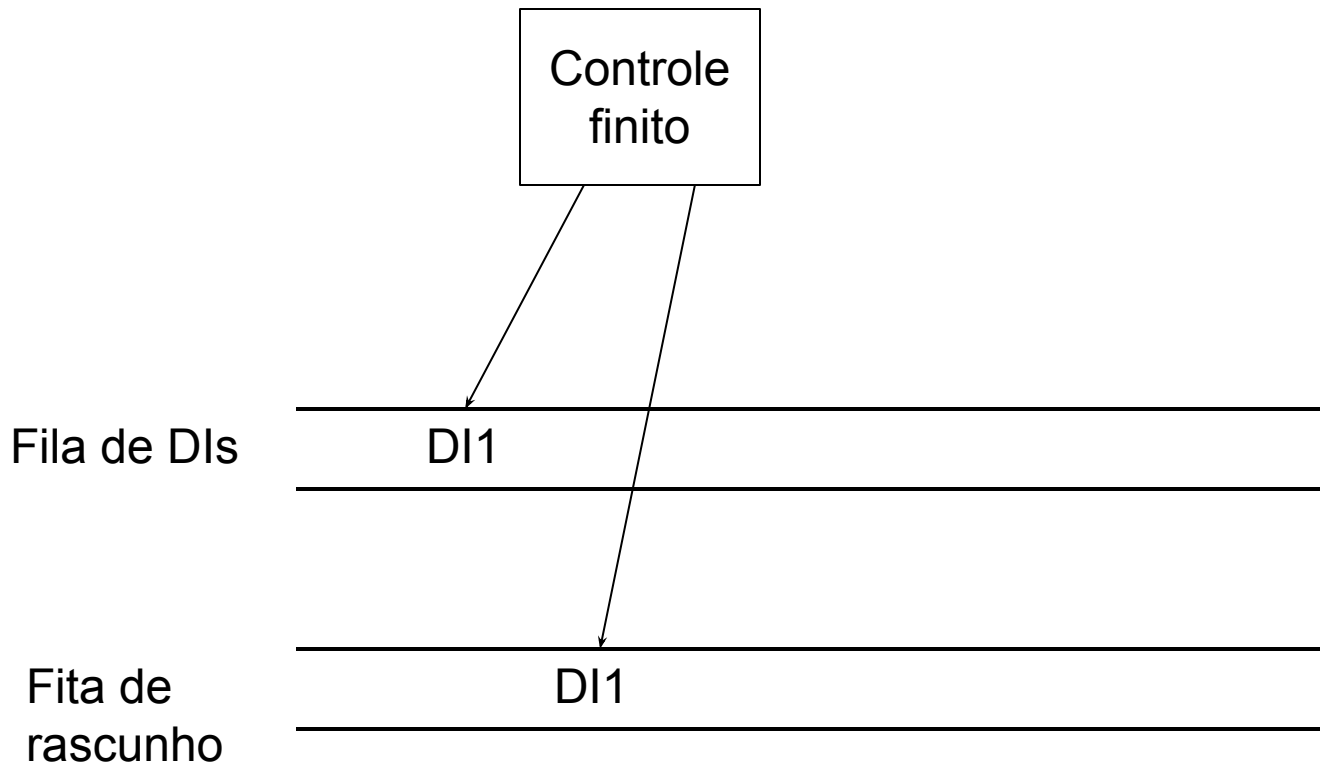
- A DTM fará isso
 - Usará uma fita para armazenar, numa fila, as DIs a serem testadas
 - Nessa fita, uma trilha adicional é usada para “marcar” qual a DI sendo atualmente testada
 - Usará outra fita para ajudar a fazer “cópias” de uma DI imediatamente antes de um ponto de não-determinismo
 - Se, em uma determinada DI, existem k possibilidades de escolha, serão criadas k cópias daquela DI
 - Essas cópias serão armazenadas no fim da fila (fita 1)

Conversão NTM \rightarrow DTM



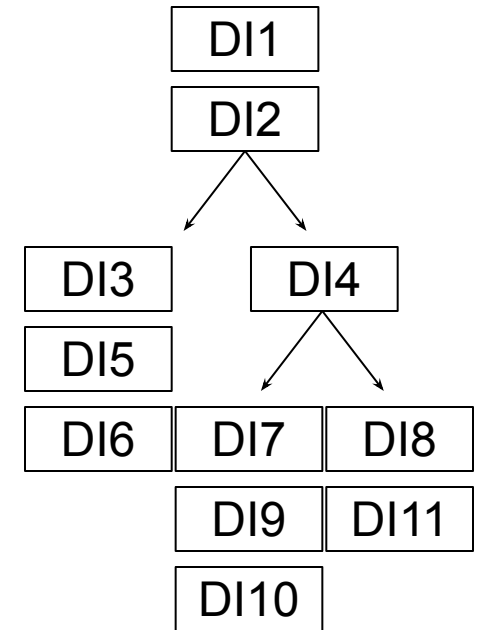
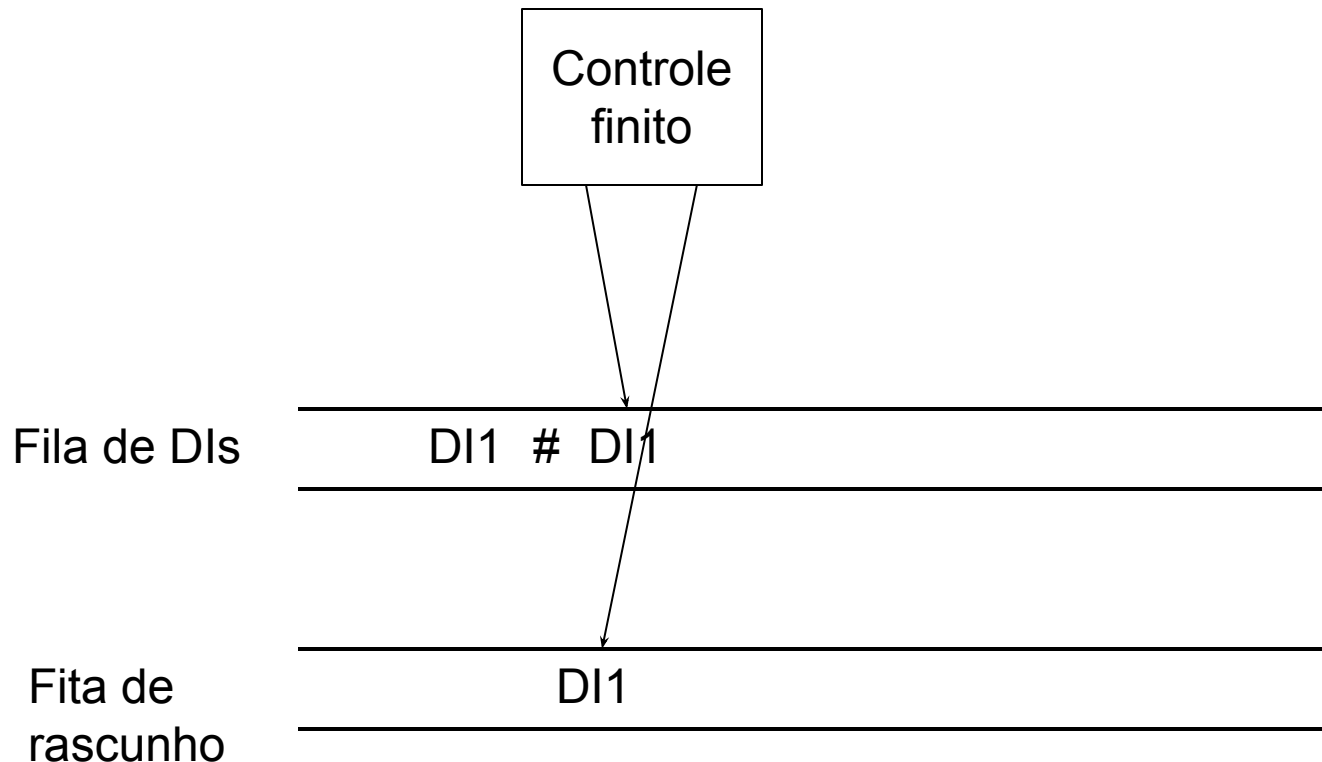
A execução começa. DI1 está na primeira fita. DI1 não é uma DI de aceitação, portanto DTM precisa seguir em frente.

Conversão NTM \rightarrow DTM



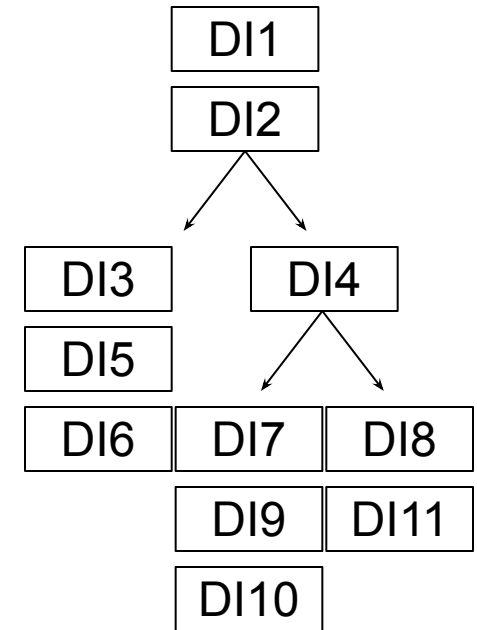
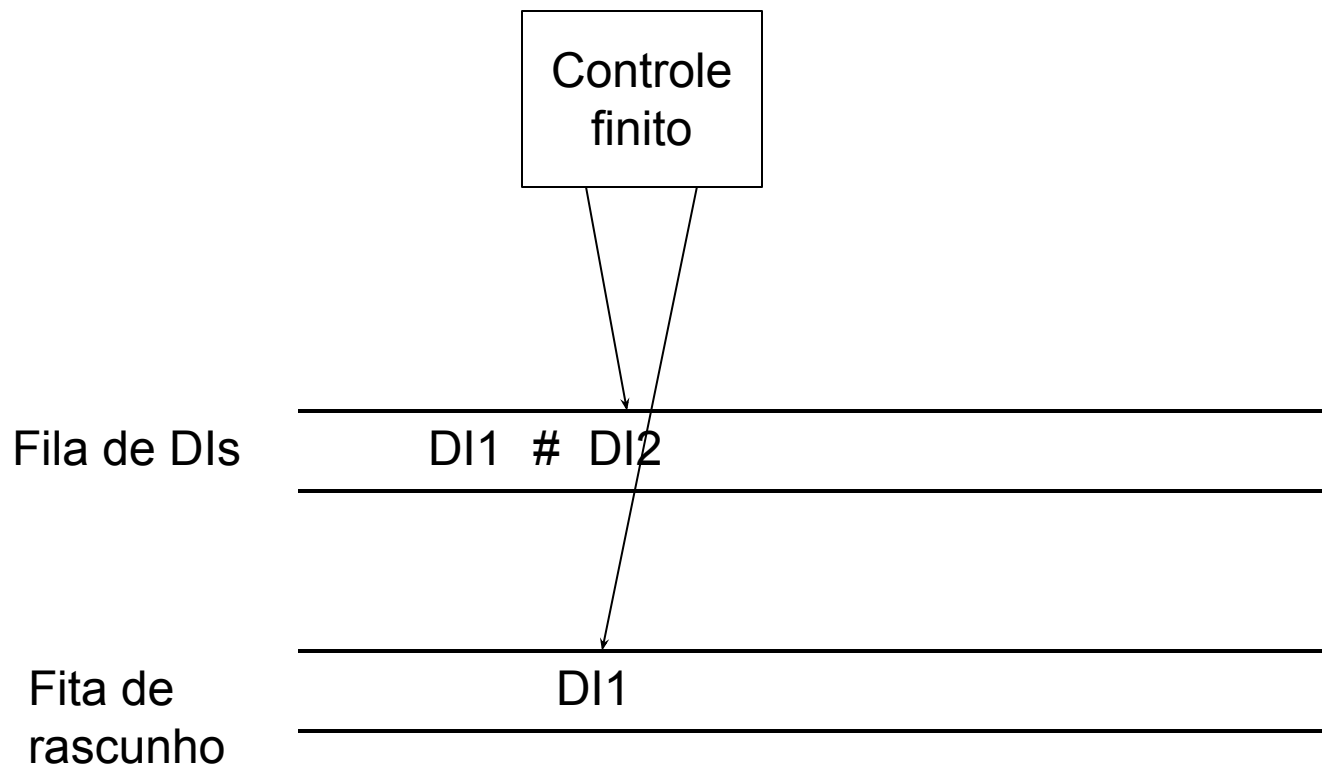
DTM avalia o número de possibilidades a partir de DI1. Só existe 1 alternativa, e portanto DTM vai criar uma cópia de DI1. Para isso, ela insere DI1 temporariamente na fita de rascunho, que irá servir de “memória” durante a cópia

Conversão NTM \rightarrow DTM



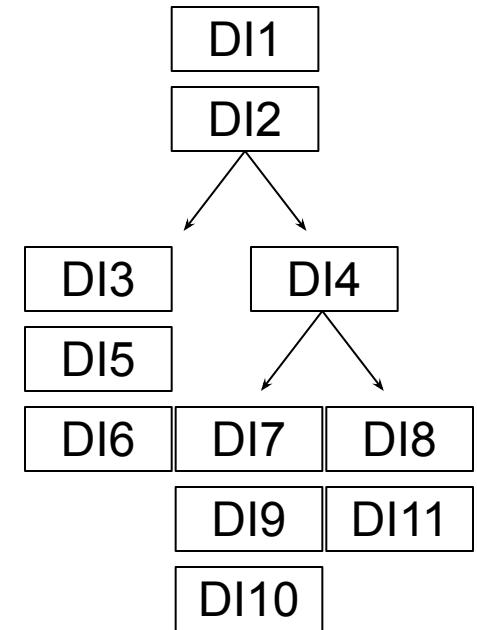
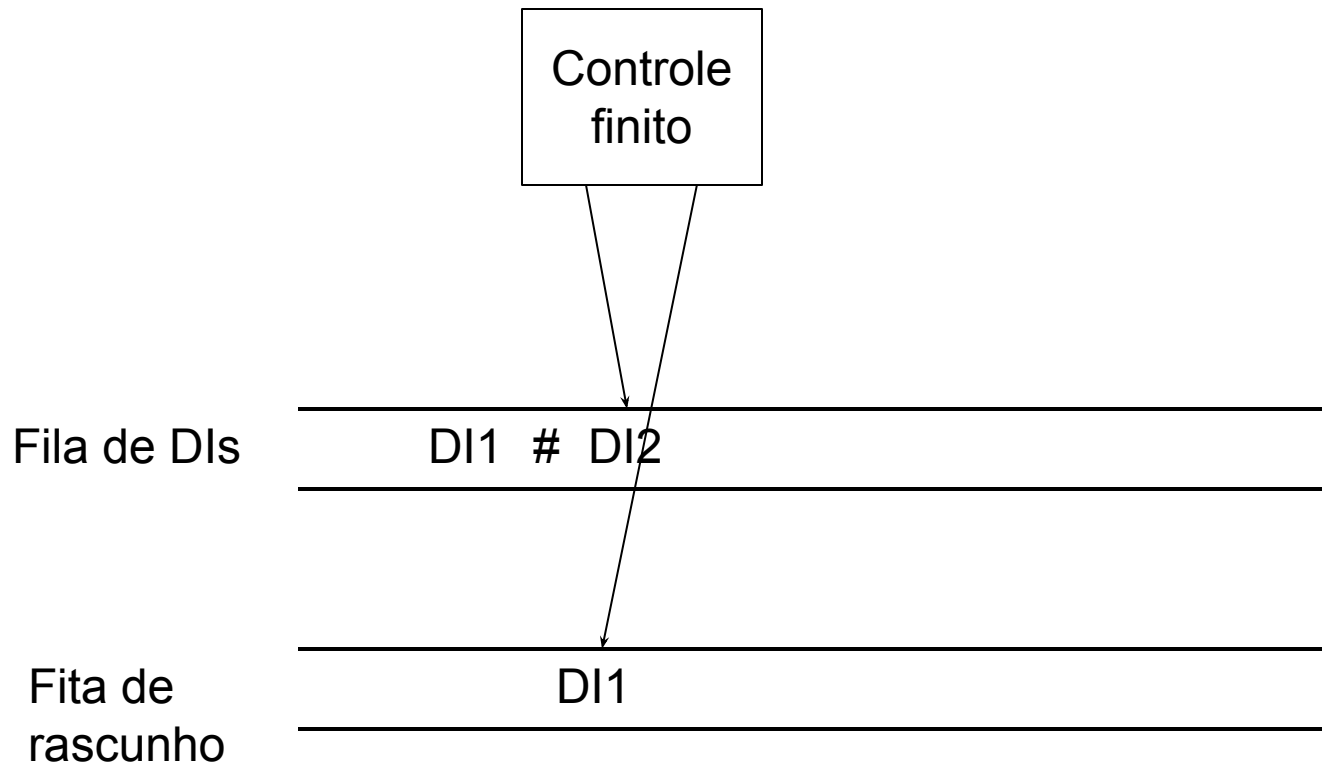
Tendo armazenado DI1 na fita de rascunho, a DTM faz uma cópia, inserindo-a no final da fila. Um símbolo especial (#) serve de separador entre as DIs na fila

Conversão NTM \rightarrow DTM



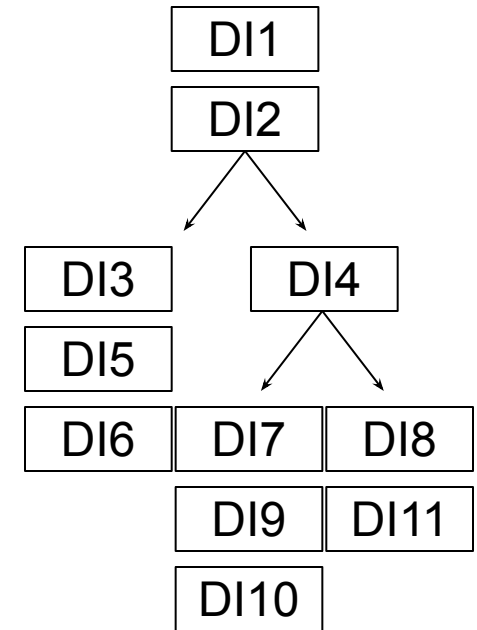
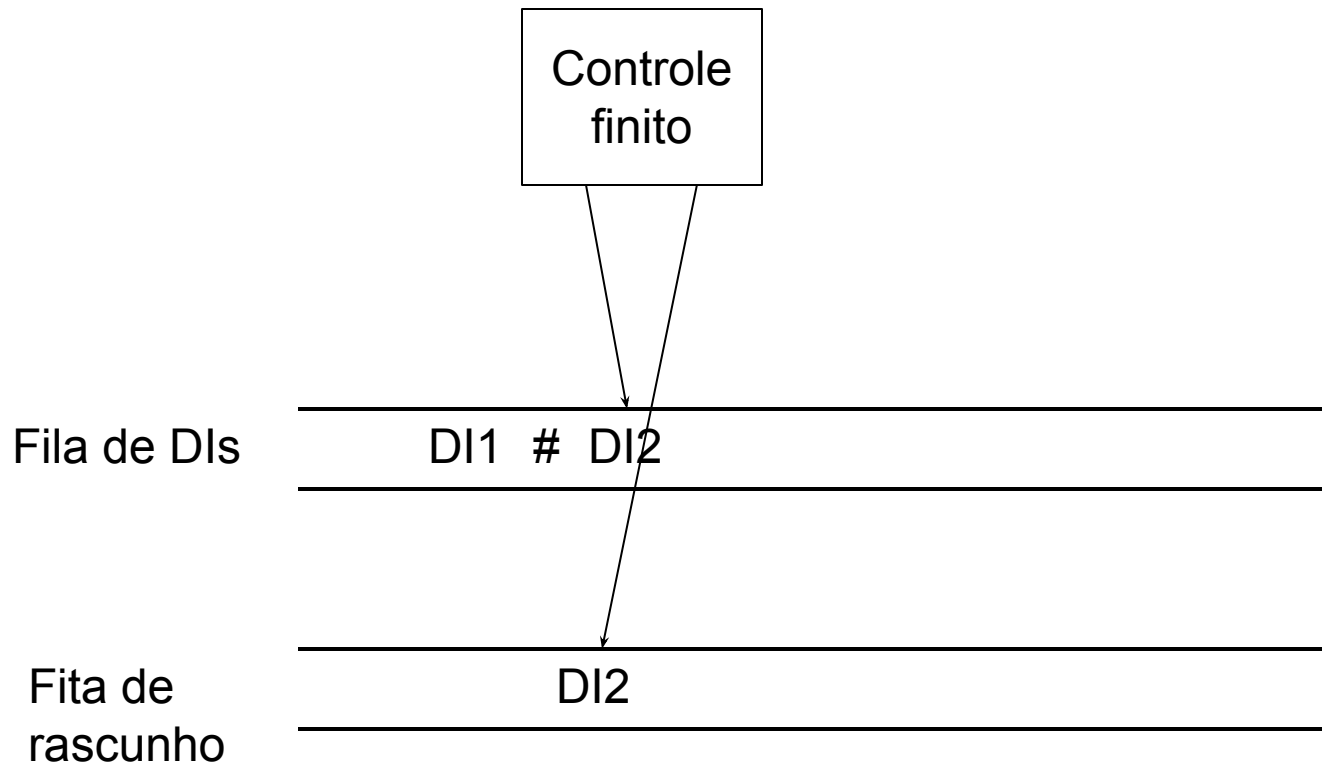
DTM então modifica cada cópia de acordo com as possibilidades de movimento. Nesse caso, só há uma cópia, e uma possibilidade de movimento (DI2).

Conversão NTM \rightarrow DTM



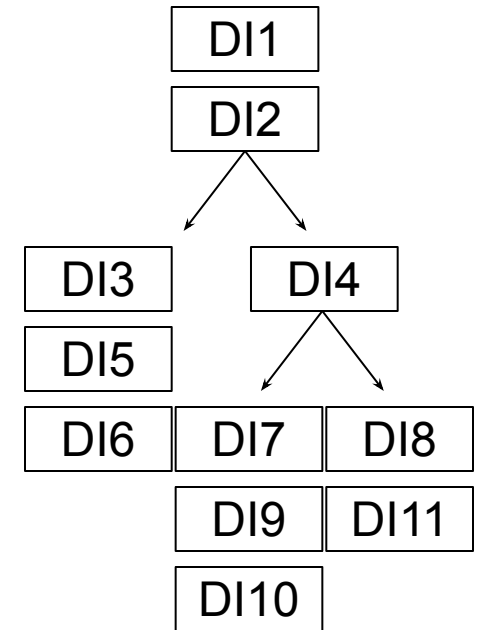
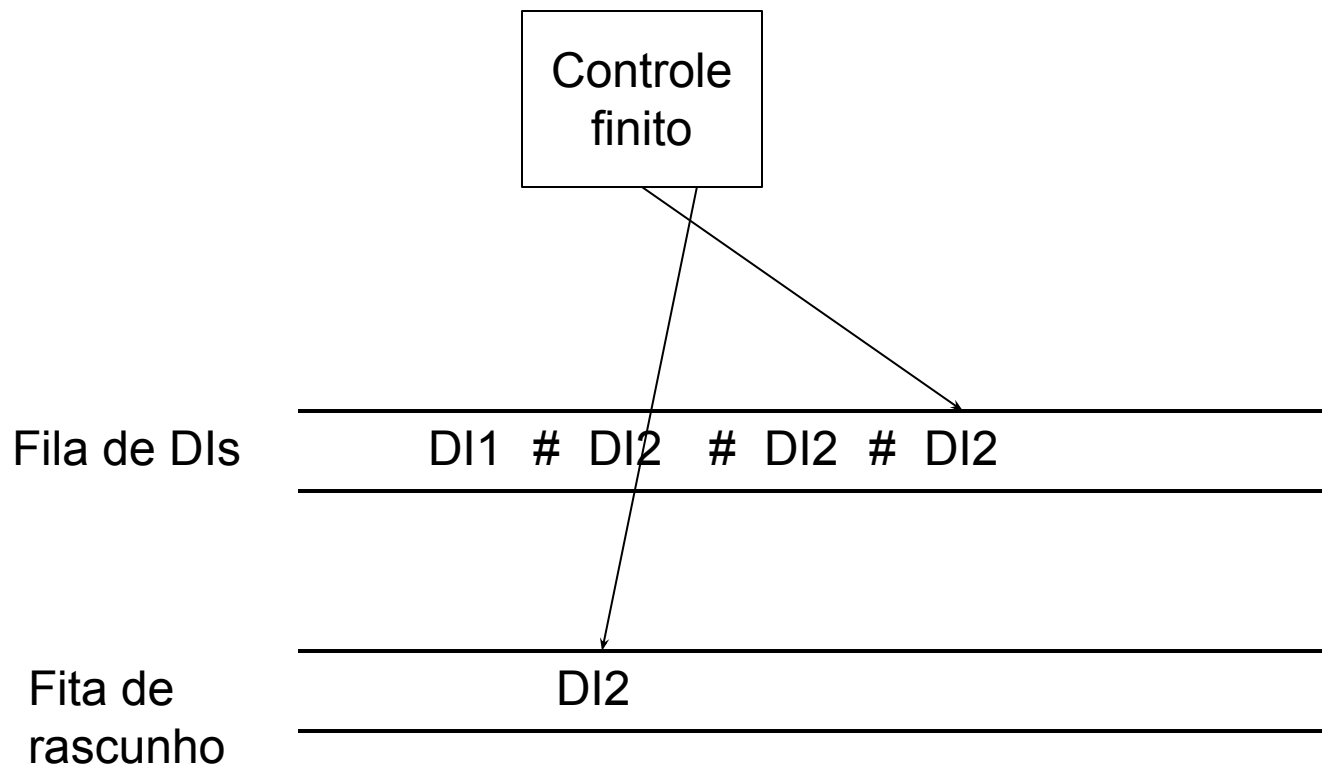
Avaliando DI2, DTM descobre que esta não é uma DI de aceitação. Portanto, ela precisa ir em frente. Avaliando as possibilidades, ela identifica 2 alternativas, portanto DI2 deverá ser copiada 2 vezes.

Conversão NTM \rightarrow DTM



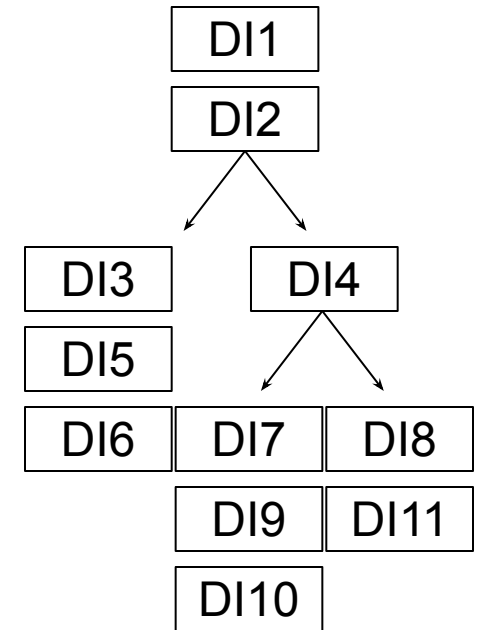
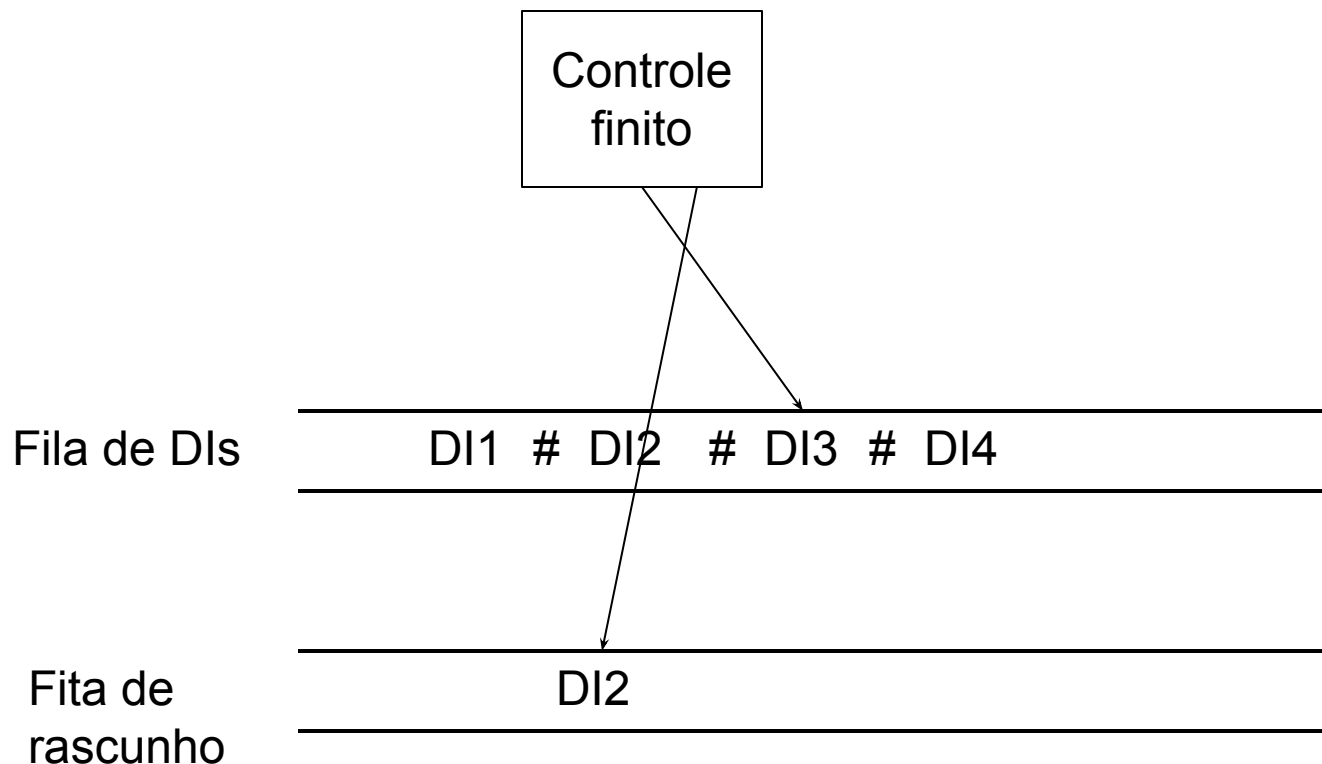
Para fazer a cópia, DTM usa a fita de rascunho como memória, inserindo DI2 no lugar do valor anterior.

Conversão NTM \rightarrow DTM



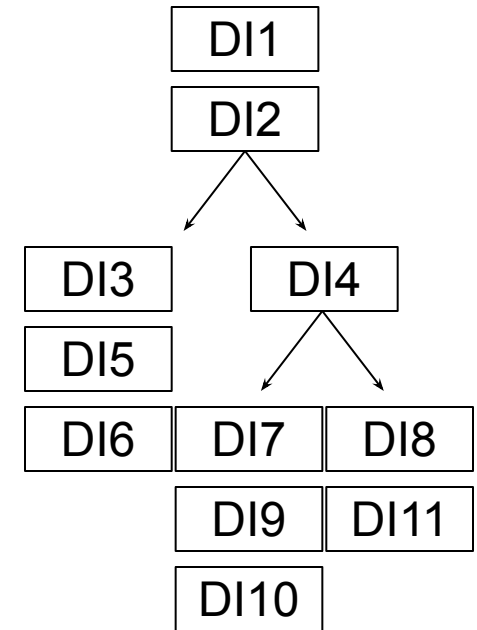
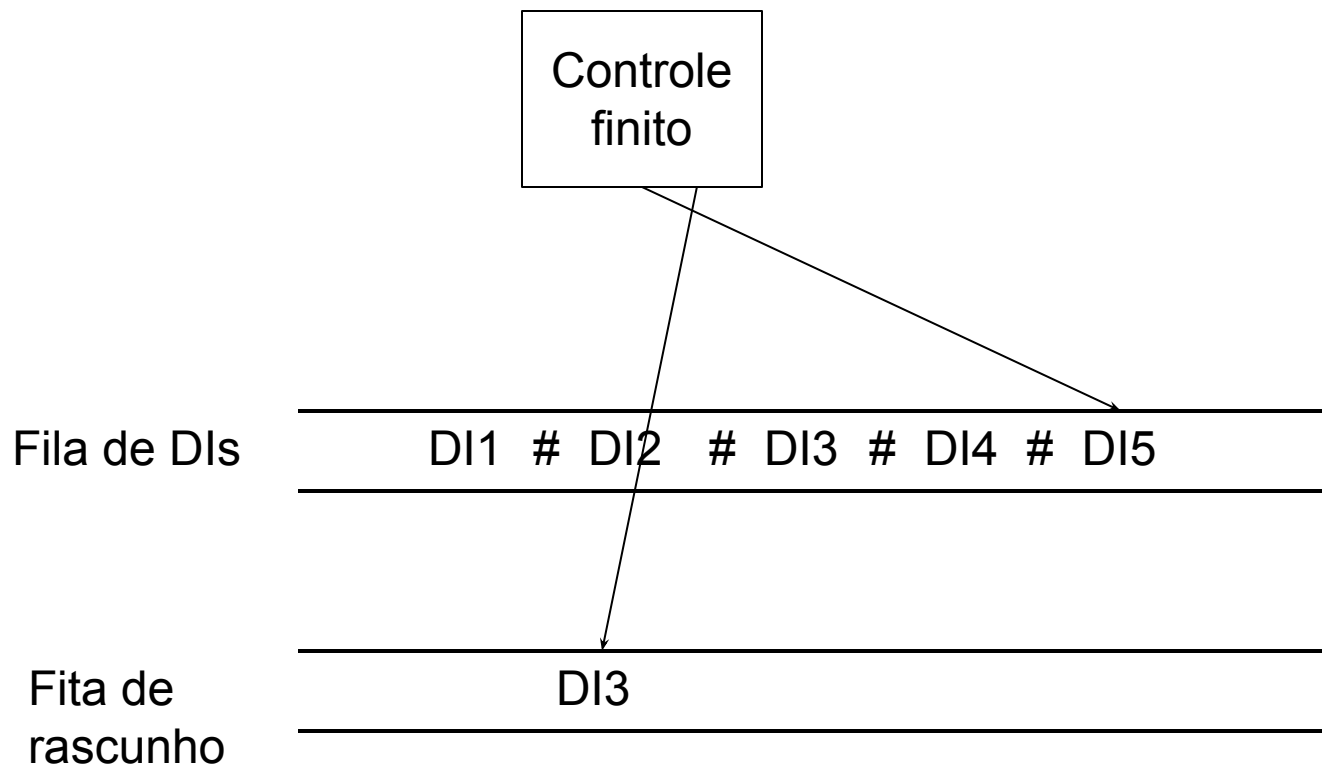
Em seguida, DTM insere 2 cópias de DI2 no final da fila, copiando a partir do rascunho.

Conversão NTM \rightarrow DTM



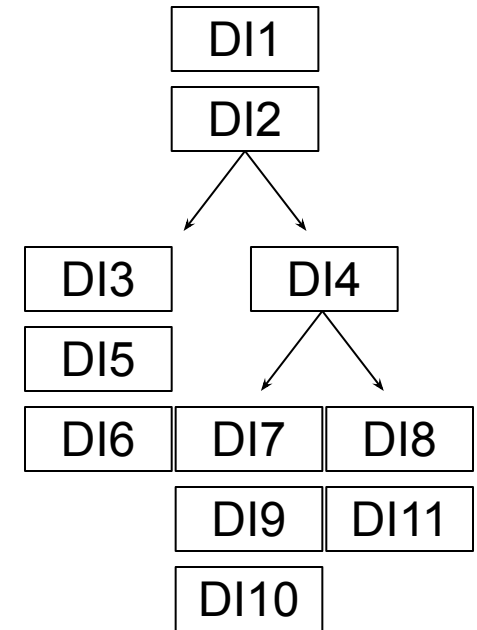
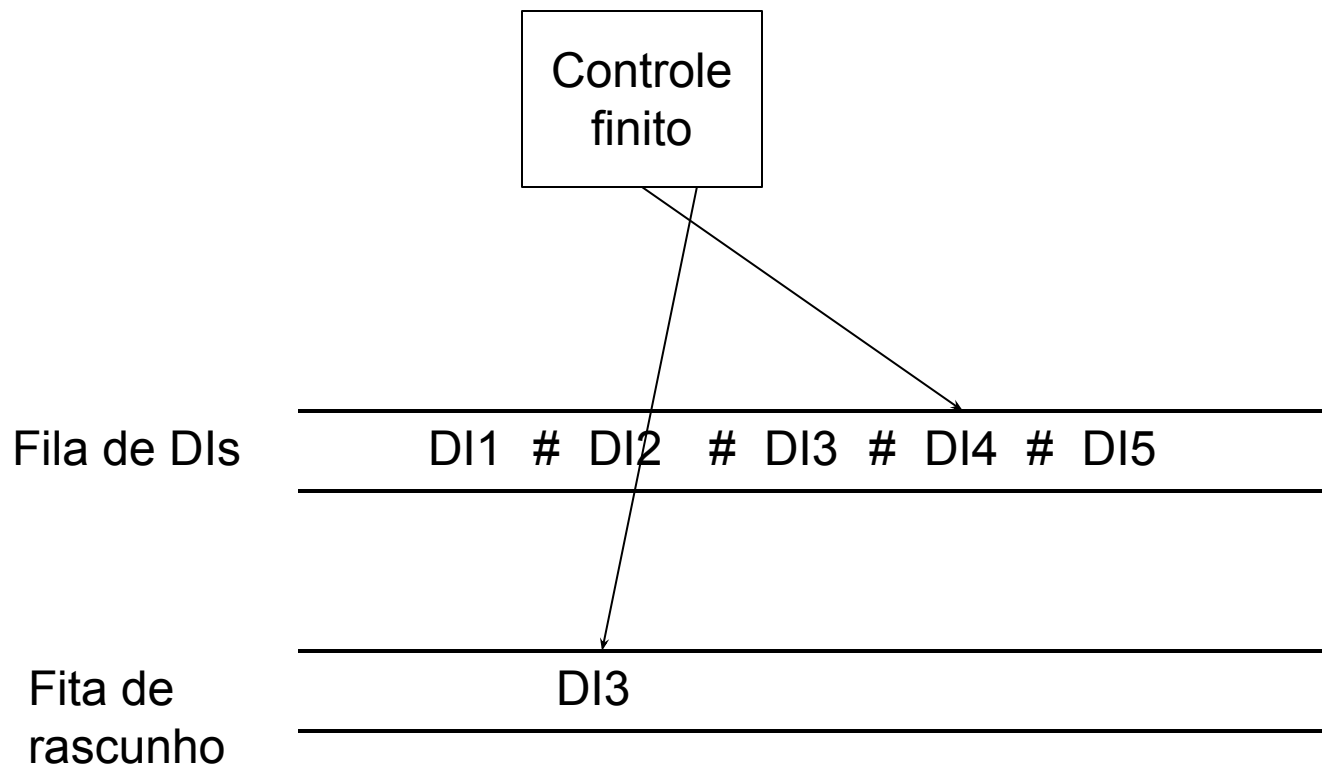
DTM faz o retorno, substituindo cada cópia de `DI2` por uma alternativa diferente, no caso, `DI3` e `DI4`

Conversão NTM \rightarrow DTM



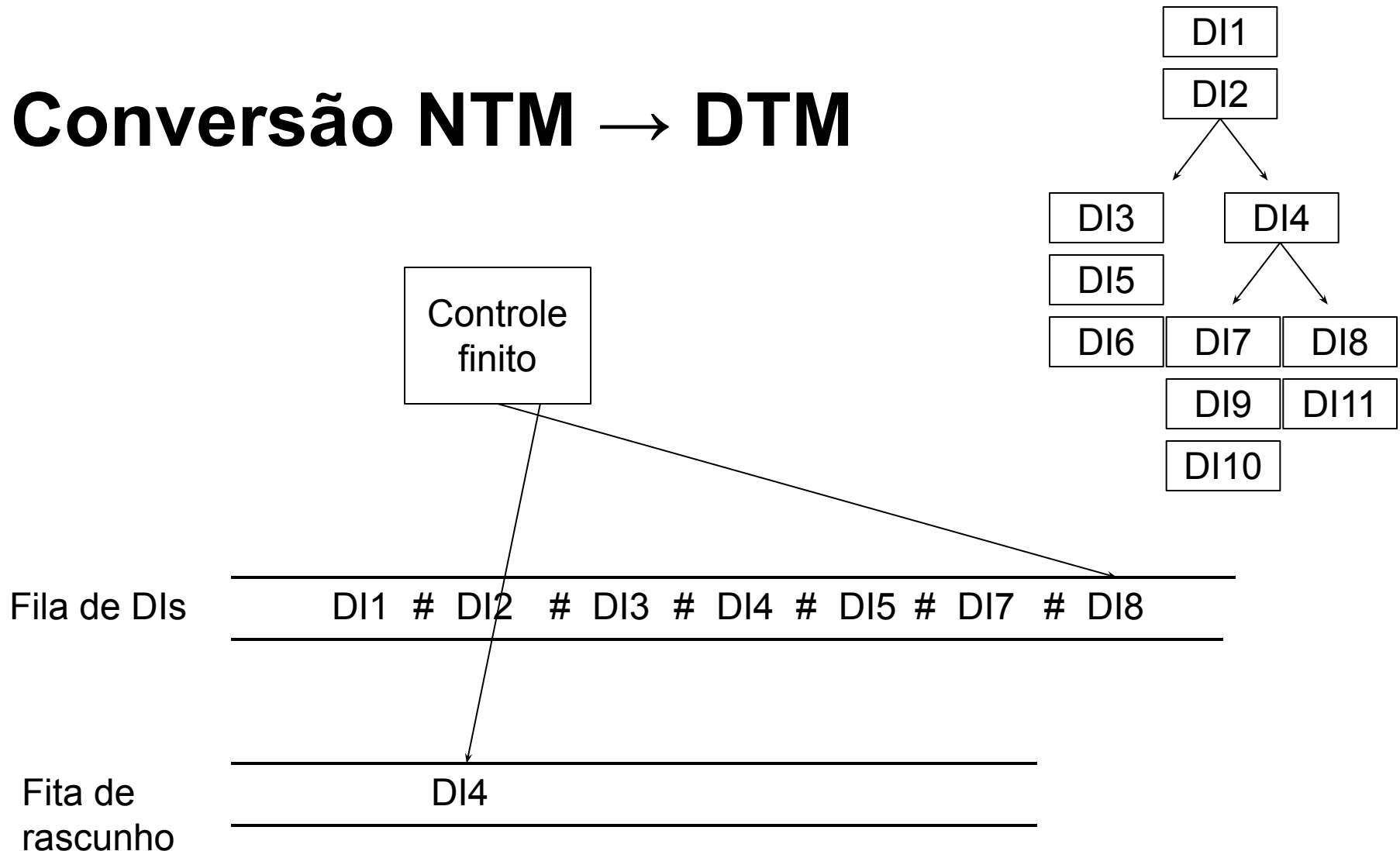
Não sendo uma DI de aceitação, DTM faz a análise de alternativas a partir de DI3, criando cópias no final da fila e alterando-as para o próximo movimento possível. Nesse caso, só existe a possibilidade de ir para DI5

Conversão NTM \rightarrow DTM



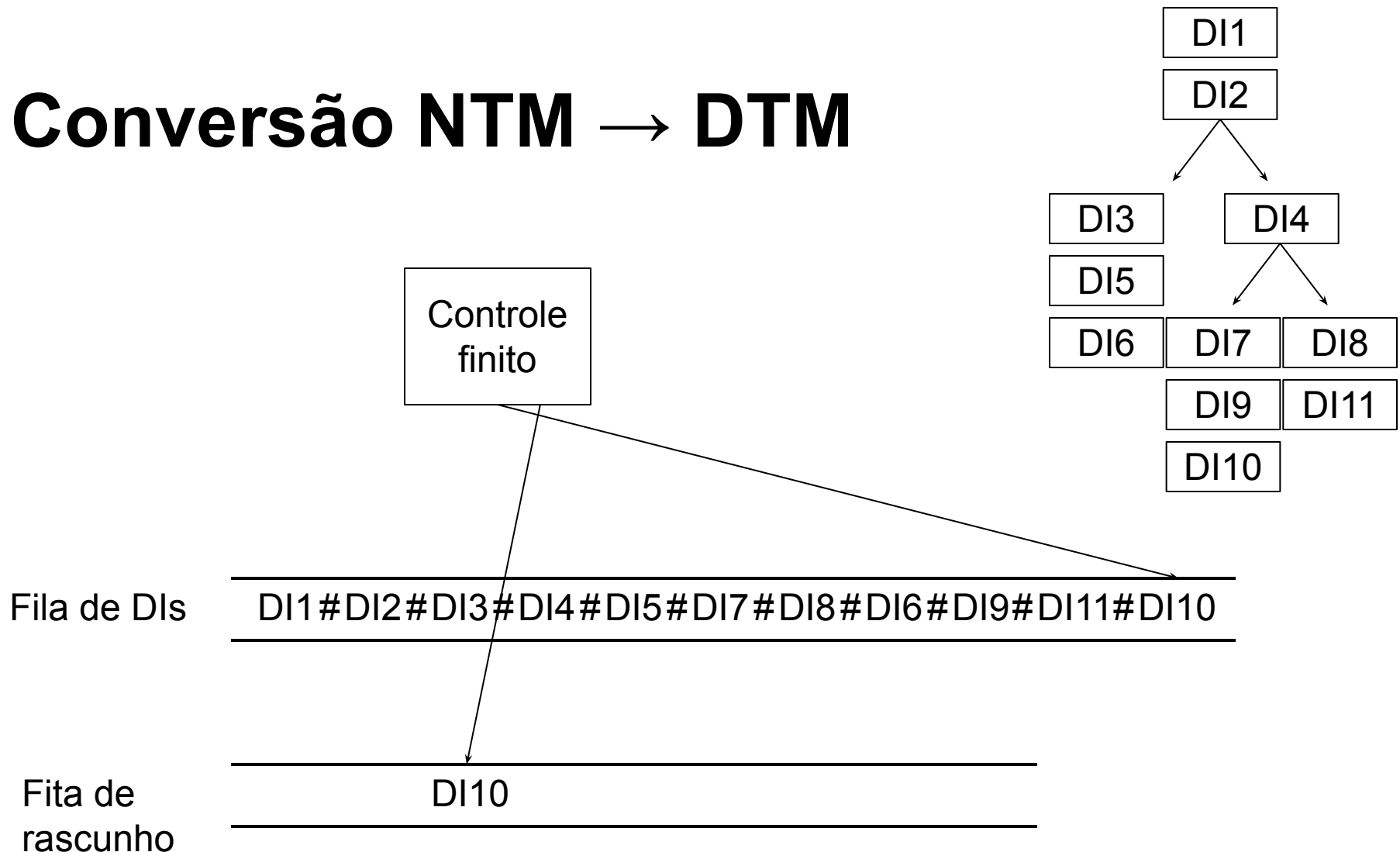
Antes de avaliar DI5, porém, DTM volta para a próxima DI não analisada, ou seja, DI4

Conversão NTM \rightarrow DTM



DI4 também não é uma DI de aceitação, e portanto novas DIs serão inseridas no final da fila, representando as possibilidades, DI7 e DI8

Conversão NTM \rightarrow DTM



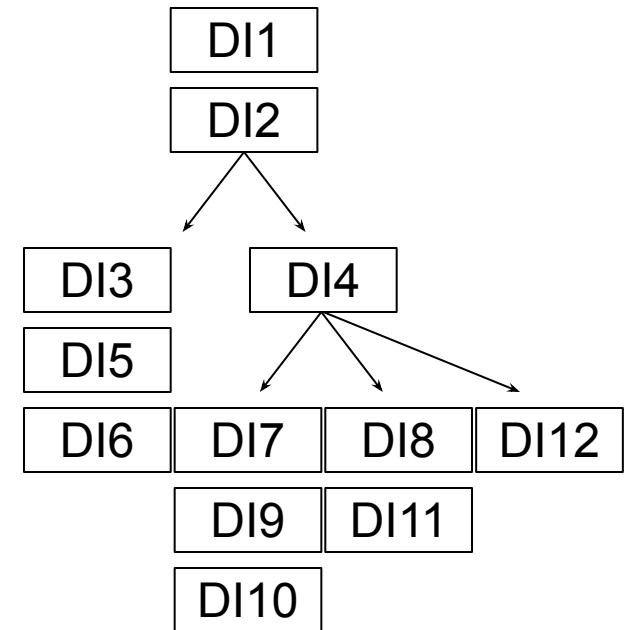
Esse procedimento continua até serem esgotadas todas as possibilidades.
Analise a fita resultante e note que se trata de uma busca “primeiro em amplitude”

Conversão NTM \rightarrow DTM

- Durante esse percurso, caso a máquina entre em uma DI de aceitação, ela aceita e encerra a execução
 - Caso encontre uma DI de rejeição, ela continua na próxima DI da fila
- Essa simulação é precisa
 - DTM só aceita se NTM pode entrar em uma DI de aceitação

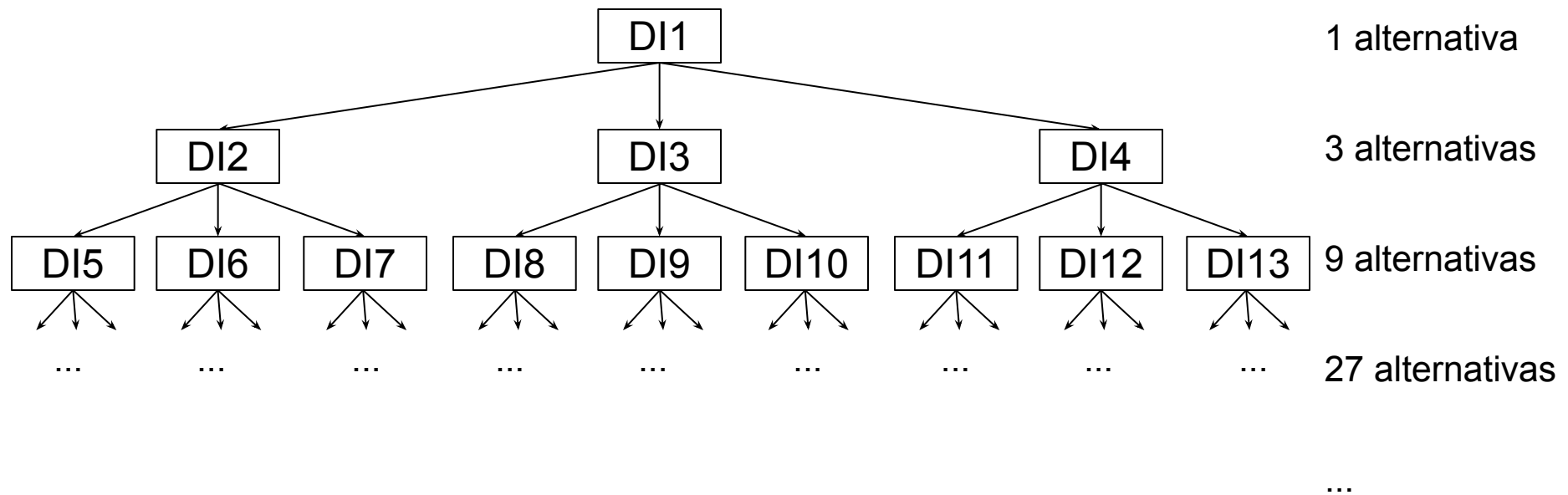
Conversão NTM \rightarrow DTM

- Seja m o número máximo de escolhas que NTM tem em qualquer configuração
 - Nesse exemplo à direita, $m=3$ (DI4 \rightarrow DI7, DI8, DI12)
- Vamos então analisar um caso extremo!
 - Em toda DI, há 3 escolhas!



Conversão NTM \rightarrow DTM

- Essa figura ilustra um caso extremo de não-determinismo, onde, a cada DI, temos 3 escolhas



Conversão NTM \rightarrow DTM

- Ou seja, vemos uma progressão:
 - 1,3,9,27,...
 - Ou: $3^0, 3^1, 3^2, 3^3, 3^4, \dots$
 - Generalizando: $m^0, m^1, m^2, m^3, m^4, \dots$
 - Onde m é o número máximo de escolhas que a NTM pode fazer em todas as Dis
- Pergunta: por quantas DIs a NTM pode passar, no máximo, em:
 - 1 movimento? Resp: 1 DI
 - 2 movimentos? Resp: 1+3 DIs
 - 3 movimentos? Resp: 1+3+9 DIs
 - 4 movimentos? Resp: 1+3+9+27 DIs

Conversão NTM \rightarrow DTM

- Generalizando: em n movimentos, a NTM pode alcançar no máximo:
 - $1 + m + m^2 + m^3 + \dots + m^n$ DIs
 - Esse número é no máximo nm^n DIs
- A DTM equivalente faz uma busca primeiro em amplitude, isto é:
 - Primeiro ela testa as DIs que seriam alcançadas em 0 movimentos da NTM
 - Depois ela testa as DIs que seriam alcançadas em 1 movimento da NTM
 - Depois as DIs que seriam alcançadas em 2 movimentos da NTM
 - E assim por diante

Conversão NTM \rightarrow DTM

- Generalizando:
 - Se NTM leva n movimentos para chegar à aceitação...
 - A DTM levará no máximo nm^n movimentos!!
- Eventualmente a NTM chegará na DI de aceitação
 - Ou seja, se a NTM aceita, a DTM também o fará (mesmo que demorando mais)
 - Concluimos que $L(\text{NTM}) = L(\text{DTM})$

Conversão NTM \rightarrow DTM

- Vimos que a DTM demora exponencialmente mais tempo que a NTM
 - Essa diferença tem importantes consequências práticas
- Mas não se sabe se essa lentidão é realmente necessária
 - Existe um modo melhor de simular uma NTM de forma determinística?
 - A resposta, e mais detalhes sobre isso, veremos mais adiante

Fim

Aula 22 - Não-determinismo em Máquinas de Turing