Esboço da prova:

Queremos simular uma MTND $M = (\Sigma_0, \Sigma, Q, q, F, \Delta)$

Como sabemos, a "ramificação" de cada configuração é no máximo $r \in \mathbb{N}$ que só depende de M.

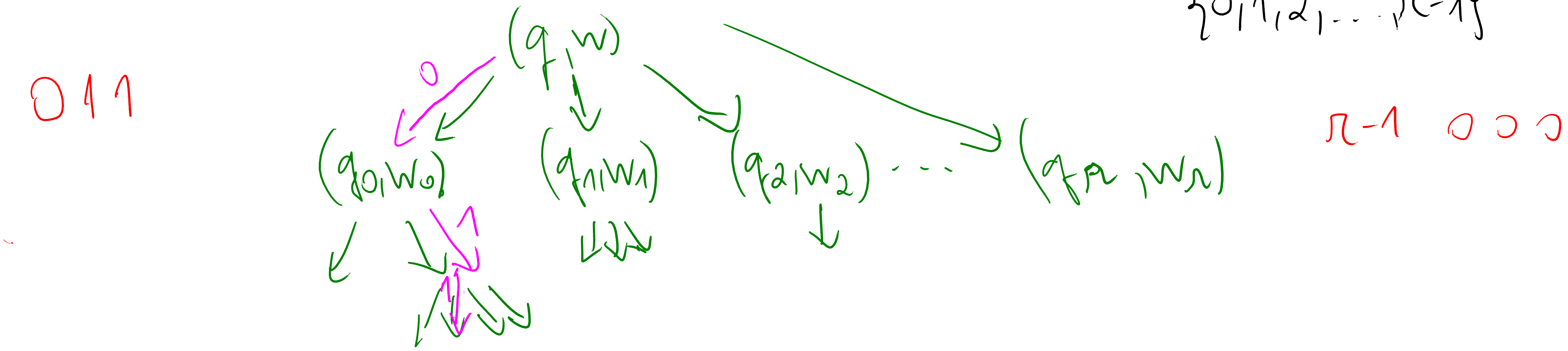
Fato: Existe uma máquina de Turing N que enumera o conjunto $\{0, 1, 2, \dots, r-1\}^*$

$$\{0,1,2,\dots,r-1\}^* = \{\varepsilon, 0, 1, \dots, r-1, 01, 02, 03, \dots\}$$

Exercício

Para simular M com uma M.T. determinística, Vamos usar 3 fitas: • a primeira guarda a entrada original de M

• a segunda fita guarda o "caminho atual" na árvore de execução de M: uma palavra em $\{0,1,2,\dots,r-1\}^*$



- a terceira fita é onde a execução é de fato simulada: usando a entrada (guardada na primeira fita), vamos ver se o caminho dado pela segunda fita chega a uma configuração terminal.
  Se sim, também terminamos (num estado correspondente, caso seja decisor)
  Se não, apagamos a terceira fita, usamos a máquina N para gerar um novo caminho na segunda fita, e tentamos novamente ∎

- Propriedades de Fechamento de Rec & R.E.

  - Rec é fechada para $\cap$, $\cup$, $\overline{}$, $-$

    diferença

    complemento

  - R.E. é fechada para $\cap$, $\cup$

    mas não $\overline{}$

Rascunho: • Rec fechada para $\cap$:

$L_1, L_2 \in Rec$, $L_3 = L_1 \cap L_2$

↳ decididas por M_1, M_2

definindo M_3

Duas fitas

entrada

Roda M_1 e depois M_2 com a entrada

- R.E fechada para $\cap$.

  $L_1, L_2 \in$ R.E., aceitas por M_1, M_2.

  Definindo aceitador M_3 para $L_3 = L_1 \cap L_2$ (a ideia acima!)

- R.E. fechada para $\cup$

  $L_1, L_2 \in$ R.E., aceitas or M_1, M_2.

  A ideia anterior ("em sequência") não funciona!

  Rodamos em paralelo: 3 fitas: entrada

  M_1

  M_2

trocamos a execução de M_1 para M_2 periodicamente

Com ideia parecida, temos:

Teorema: $L \in Rec \iff L, \overline{L} \in R.E.$

Logo, se encontrarmos $L \in R.E. \setminus Rec$, teremos que R.E. não é fechada para complemento.

- Indecidibilidade

À parte: como vimos, podemos codificar cada M.T. como um número natural.

As linguagens correspondem a conjuntos de números naturais

Teorema de Cantor: A quantidade de naturais é estritamente menor que a de conjuntos de naturais

Ideia da prova: Cada conjunto de naturais pode ser visto como uma sequência infinita de 0s e 1s: a i-ésima posição é 1 se i está no conjunto, 0 senão. Queremos então provar que há *mais* sequências infinitas de 0s e 1s do que naturais.

Note que há *pelo menos* tantas destas sequências quanto há naturais. (binário completada com 0s, ou $n \mapsto 0^{n-1}1000\ldots$)

Logo, se (para contradição) *não há* mais sequências inf. de 0s e 1s do que naturais, é porque há *a mesma* quantidade.

Isso quer dizer que podemos listar naturais e tais sequências lado a lado

$\mathbb{N}$	seqs.									
0	$\boxed{0}$	1	0	0	0	1	0	0	...	1
1	0	$\boxed{0}$	0	0	0	0	0		...	1
2	1	1	$\boxed{1}$	1	1	1	1		...	0
3	0	1	0	$\boxed{1}$	0	1	0		...	0
4	1	1	1	0	$\boxed{1}$	1	1		...	0
5	1	1	0	0	0	$\boxed{1}$	1		...	0
6	0	1	1	0	0	1	$\boxed{1}$		...	0
⋮	⋮									⋮

Diagonalizar!

Definimos sequência inf. de 0s e 1s que <u>não</u> aparece na tabela: na i-ésima posição, colocamos 0 se a i-ésima posição da i-ésima sequência temos 1, e colocamos 1 caso contrário

Afirmação: A sequência construída não aparece na tabela!
(De fato, ela difere de cada sequência da tabela em pelo menos uma posição)

Isso contradiz a suposição de que a tabela contém todas as seqs infs de 0s e 1s. ∎

Corolário: Existem linguagens não decidíveis.
(a quantidade de linguagens é maior que a de decisores)

Queremos um teorema mais refinado! ($RE \setminus Rec \neq \emptyset$)
Isso será dado pelo Problema da Parada.

- Máquina de Turing Universal

  ⇝ Uma máquina U que, ao receber como entrada uma descrição de uma máquina M e uma descrição de uma entrada w para M, simula a execução de M em w, atingindo o mesmo resultado

<u>Duas etapas</u> ① como descrever MTs de forma que a máquina universal "possa entender"

② descrever o funcionamento de U

---

② Usamos 3 fitas:
- a primeira contém a descrição de M (como funcionam as transições de M)
- a segunda contém o estado "atual" de M
- a terceira é uma cópia do conteúdo "atual" da fita de M

SPDG, assumimos que M é de 1 fita e determinística

→ mantemos os dois primeiros cabeçotes nos começos das fitas, e o terceiro onde o cabeçote de M de fato estaria.

→ para simular 1 passo da execução de M, fazemos:

1) ler o estado q de M na segunda fita
2) ler o símbolo σ que está sob o cabeçote na terceira fita
3) varrer a primeira fita buscando a instrução de M correspondente $(q, \sigma) \longmapsto (q', x)$
4) se não encontrar, fim da simulação (se for decisor, temos que conseguir inter-

5) se encontrar, agimos de acordo com x:

a) se x for um símbolo de M, o escrevemos na terceira fita na posição atual

b) se x for $\leftarrow$ ou $\rightarrow$, movemos o cabeçote da terceira fita de acordo.

pular q como aceitação/negação, e entrar no estado correspondente de U

6) Trocamos o estado na segunda fita de q para q'

(1) Fixamos os alfabetos das máquinas: cada símbolo será da forma aw com $w \in \{0,1\}^*$, sendo a um símbolo novo.

(2) Cada estado será da forma qw com $w \in \{0,1\}^*$ sendo q um símbolo novo

(3) Para descrever uma máquina $M = (\Sigma_0, \Sigma, Q, q, F, \delta)$, escolhemos $n \in \mathbb{N}$ como o menor natural tal que $2^n \geq |\Sigma| + 2$ (+2 porque trataremos $\leftarrow, \rightarrow$ "como símbolos")

e $m \in \mathbb{N}$ o menor tal que $2^m \geq |Q|$.

Assumimos que

vazio de M $\rightarrow$ $\sqcup$ é $a0^n$

início de M $\rightarrow$ $\triangleright$ é $a0^{n-1}1$

$\leftarrow$ é $a0^{n-2}10$

$\rightarrow$ é $a0^{n-2}11$

estado inicial é $q0^m$

Se M é decisor, s "vem antes" de n na ordem lexicográfica

Para descrever δ, usamos mais três símbolos novos

( ) ,

e montamos uma sequência

$(q_0, a_0, q_0', x_0), (q_1, a_1, q_1', x_1), (q_2, a_2, q_2', x_2), \ldots, (q_k, a_k, q_k', x_k)$

onde q_i, q_i' correspondem a estados (são palavras do tipo qw com $w \in \{0,1\}^*$)

a_i corresponde a símbolo

x_i corresponde a símbolo ou $\leftarrow$ ou $\rightarrow$

(q_i, a_i, q_i', x_i) indica que δ manda M, se estiver no estado correspondente a q_i lendo o símbolo " a a_i,

trocar para o estado corr. a q_i' e
fazer a ação corr. a α_i

As quádruplas em W aparecem em ordem
lexicográfica