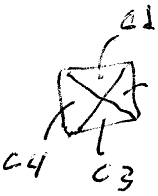
EA869 - a colecção escrita na lousa

pags 68a →

Harel: fls: 1 - 58

Tile = azulejo

Um tile é um quadrado  com orientação fixa

c1 c2 c3 c4

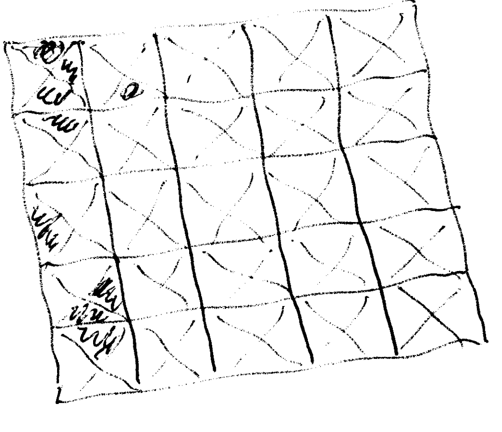
Uma entrada é um conj. T de tiles

Problema: preencher uma área (quadrada) finita com tiles sob a restrição de que a cor de tiles q se tocam tem a mesma cor no lugar do toque (ci).

Exemplo

seja uma área de 5x5 tiles e

T = 

⟹  etc → OK

Mudar T para: 

Para facilitar área 3 x 3

etc

⟹ não dá!

Pode tentar

Não há algoritmo, e nunca haverá, p/ resolver o "tiling problem".

⟹ NÃO COMPUTÁVEL.

INDEPENDENTE DO PROGRAMADOR E DA MÁQUINA!

Problema da parada

Seja 1. while $X \neq 1$ do

if X par, $X \leftarrow X/2$

if X impar, $X \leftarrow 3X+1$

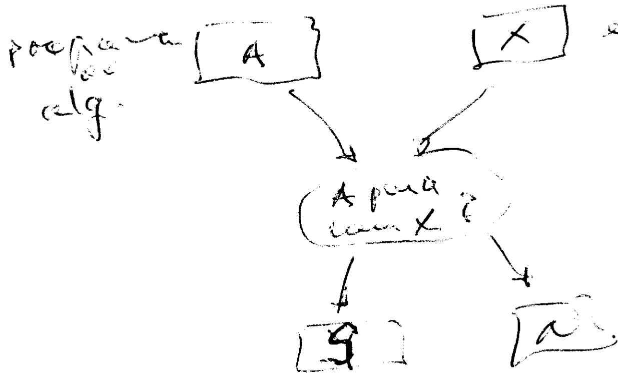
2. stop

exemplo: $X = 7$

$\Rightarrow \{ 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1 \}$! parou

Ao longo dos anos esta simples sequência tem sido testada para inúmeros valores iniciais de X. Nenhuma periodicidade pode ser obtida e nenhuma prova que não termina. Por outro lado, não foi provada a parada p/ ∀ nº inteiro

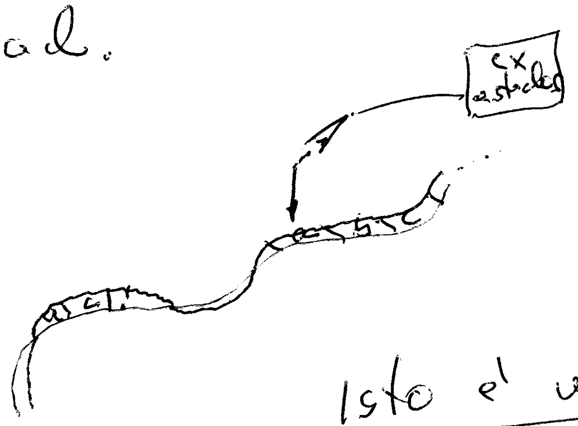
Genericamente o problema da terminação não é decidível (computável), ou seja:



Este é um problema de verificação ⇒ não computável.

Um dispositivo Computacional Elementar

- considere que o dispositivo que estaremos definindo aceita entrada/saída através de uma fita, unidimensional, consistindo de quadrados cada um podendo conter 1 símbolo de um alfabeto.
- o dispositivo pode "olhar" (através de uma cabeça) um quadrado da fita e examinar o símbolo nele contido ou sobrescrever o símbolo no quadrado.
- o dispositivo pode mover/~~se~~ na fita p/a esq. ou p/a direita
- o dispositivo possui ~~uma~~ uma cx de mudanças com várias (finitas) posições — seus estados.
- o dispositivo decide a próxima ação após ler a fita e ~~definindo~~ considerando o estado atual.



Isto é uma Máquina de Turing → TM

Exemplo

Seja uma TM programada para somar 2 nºs decimais X e Y.

A entrada é dada na fita.

Sejam: X ≡ 36

Y ≡ 519

+ ≡ soma

≡ branco/espaço

! ≡ separador

cabeça à esquerda na fita

FITA NO passo 1 (ao início): .... | # | # | # | # | 3 | 6 | + | 5 | 1 | 9 | # | # | ....

- Cabeça percorre a fita, posição a posição, sem fazer alterações até encontrar "+". Então move uma posição (1 quadrado) à esquerda
- então apaga (ou seja escreve #) sobre "6" e "memoriza" o estado "6" (há 10 estados possíveis: 0 a 9)

.... | # | # | # | # | 3 | # | + | 5 | 1 | 9 | # | # | ....

- então a cabeça move-se p/a direita até encontrar "#" e então uma posição à esquerda e apaga "9" memorizando o estado da soma de "5" e "9" e se há ou não "carry". São, neste caso, 20 estados possíveis (soma de 2 dígitos decimais c/ou sem carry)

#(0+0 a 0+9 com e sem carry) #

resposta $a_1 + a_2 = \{0, 1, 2, \ldots, 9\}$ c/ou sem carry $\Rightarrow$ 20 estados

- então a cabeça move p/ a esquerda de X, escreve "!" separador do resultado e escreve "5" (carry fica no estado, vai dar 20)

...	#	#	#	#	3	#	+	5	1	#	#	...

#	#	5	!	3	#	+	5	1	#	#	...

- ideia agora inicialmente "pegando" "3" de X e somando c/ carry e então o "1" de Y e

.....	#	5	5	!	#	#	+	5	#	#	#	...

- então X não tem + dígitos, copiar Y dígito a dígito se for o caso considerando "carry" + ao final "!"

...	#	!	5	5	5	!	#	#	+	#	#	#	#	...

UFA!

NOTE

- TM tem qtde finita de estados
- cada estado é finito
- só pode ler e reescrever símbolos
- alfabeto é finito
- entretanto pode somar nos de qq tamanho!

  => lenta + funciona!

$\boxed{\text{TM podem resolver qq algoritmo}}$

Church-Turing Thesis ~1930

Assim voltando à transp. 66 TM resolvem os problemas computáveis, ou seja os ~~resolvidos~~, ou resolvíveis ou decidíveis definem uma classe robusta e invariante c/ computadores e linguagens.