

Maquina de Turing

Um problema, duas soluções

Universidade Federal do Estado da Bahia

Adalberto Neto
adalberto.manoel@ufba.br
Eng. da Computação

David Ferrari
david.ferrari@ufba.br
Eng. da Computação

Mário Santos
mariosbps@ufba.br
Ciência da Computação

Jéssica Correia
jessica.correia@ufba.br
Eng. da Computação

Paloma Arize
paloma.arize@ufba.br
Eng. da Computação

Novembro, 2023

1 Introdução

Durante a década de 1930, dois renomados matemáticos, Alan Turing¹. e Kurt Gödel², emergiram como figuras proeminentes no cenário intelectual, contribuindo de maneiras significativas para a teoria da computação. O notório projeto de Turing, concebido como um desdobramento de décadas de esforços coletivos, contou com a participação de ilustres nomes como Leopold Kronecker³, Bertrand Russell⁴, David Hilbert⁵, Paul Bernays⁶ e John Von Neumann⁷. (BRANDAO, 2017). Em 1936, Turing publicou um artigo seminal

¹ Matemático inglês Alan Mathison Turing (1912 – 1954), considerado o “pai da computação”.

² Kurt Friedrich Gödel (Brünn, 28 de abril de 1906 - Princeton, 14 de janeiro de 1978) foi um filósofo, matemático e lógico austríaco, naturalizado norte-americano. Considerado um dos mais importantes lógicos da história.

³ Leopold Kronecker (1823-1891) foi um matemático alemão que trabalhou em teoria dos números, álgebra e lógica. Ele é conhecido por suas contribuições na aritmética da análise, e diversos teoremas Kronecker.

⁴ Bertrand Russell (1872-1970) foi um matemático, filósofo, ensaísta, historiador e lógico britânico. Conhecido por suas contribuições em filosofia analítica, pensamento livre e filosofia ocidental. Russell foi um dos mais influentes matemáticos, filósofos, ensaístas, historiadores e lógicos do século XX. Ele ganhou o Prêmio Nobel de Literatura em 1950. Russell foi um pacifista e defendia o anti-imperialismo. Ele foi preso por seu pacifismo durante a Primeira Guerra Mundial.

⁵ David Hilbert (1862-1943) foi um matemático alemão que trabalhou em teoria dos números, álgebra e lógica. Conhecido por suas contribuições na aritmética da análise e por seus teoremas em matemática.

⁶ Paul Bernays (1888-1977) foi um matemático suíço que contribuiu significativamente com a lógica matemática, teoria axiomática dos conjuntos e filosofia da matemática. Foi assistente e grande colaborador de David Hilbert

⁷ John von Neumann (1903-1957) foi um matemático húngaro-americano que trabalhou em física teórica, matemática aplicada, computação, economia e estatística. Conhecido por suas contribuições em teoria dos jogos, mecânica quântica, análise funcional, cálculo de variações, equações diferenciais parciais, computação, física nuclear e estatística

que delineava uma máquina teórica, conhecida como a Máquina de Turing, representando um marco na evolução da computação.

Durante a Segunda Guerra Mundial, Turing, a serviço do Governo Inglês, direcionou seus conhecimentos para a construção de máquinas físicas, desempenhando um papel crucial na decifração da máquina de criptografia nazista. Sua contribuição não apenas salvaguardou informações cruciais, mas também desempenhou um papel essencial na prevenção de diversos ataques.

O título de "pai da computação" atribuído a Turing é mais do que merecido, uma vez que, mesmo em uma época distante dos computadores digitais (1936), a Máquina de Turing, em seu sentido preciso, apresenta-se como um modelo abstrato de um computador. Esse modelo foca estritamente nos aspectos lógicos do funcionamento de um computador, como memória, estados e transições, sem se ater à sua implementação física. A Máquina de Turing, por sua abstração, tornou-se a base para a compreensão teórica de qualquer computador digital, transcendendo as barreiras temporais e é de suma importância no avanço computacional.

2 Objetivos

Este trabalho acadêmico tem como objetivo aprofundar nosso entendimento sobre a Máquina de Turing e suas aplicações, colocando-o à prova por meio da resolução de um problema específico⁸ de maneiras distintas. Para alcançar esse propósito, serão projetadas duas Máquinas de Turing:

1. MT1: Máquina de Turing padrão (apenas uma fita e determinística);
2. MT2: Variação de Máquina de Turing padrão (n fitas).

Ambas as soluções serão submetidas à seguinte indagação: quantos passos a máquina executa no pior caso em relação ao tamanho da entrada? Este questionamento será abordado mediante um cálculo formal e detalhado, expresso em termos de notação assintótica, proporcionando uma compreensão clara da complexidade computacional envolvida.

Ao concluir este trabalho, espera-se não apenas fornecer soluções eficientes para o problema em questão, mas também enriquecer o entendimento sobre as implicações teóricas e práticas da Máquina de Turing.

3 Solução para o problema

Temos um problema de tema geográfico a ser resolvido, portanto, nossa solução advém de uma equação matemática já difundida na área. A Terra é aproximadamente esférica e leva-se cerca de 24 horas para completar sua rotação de 360 graus, ao realizar a divisão de 360 por 24 temos que 15 de longitude equivale a 1 hora e portanto equivale a um fuso horário.

Hora de B a partir de A

⁸ Disponível no Anexo I, pg 41.

Longitude astronômica e hora local são valores iguais expressos em grandezas diferentes, pois a grandeza hora representa intervalo de tempo, enquanto longitude é uma distância angular. Então: a diferença de hora local entre dois pontos é o mesmo que a diferença de longitude entre esses pontos. Matematicamente, temos:

$$H_B = H_A + F_B - F_A \quad (1)$$

Em que:

- H_B é a hora referente ao ponto B.
- H_A é a hora referente ao ponto A.
- F_B é a longitude de B em horas.
- F_A é a longitude de A em horas.

(GIBRA', 2008)

Vale lembrar que há uma regra de sinais a ser aplicada nessa equação uma vez que quando se trata de longitudes à Oeste, por convenção é considerado negativa e por sua vez o graus à Leste do Greenwich são positivos. Além disso como se trata de voo, não apenas precisamos pensar no horário do local de partida mas também temos a duração do voo, então para a nossa solução adicionaremos mais uma variável que iremos chamar de T . Por conseguinte ambas soluções se guiarão por meio desta equação para resolver o problema.

Equação final:

$$C = T + P + F_C - F_P \quad (2)$$

Em que:

- C é a hora referente ao ponto de chegada.
- T é a quantidade de horas do voo.
- P é a hora da partida.
- F_C é a longitude do ponto da chegada em horas.
- F_P é a longitude do ponto da partida em horas.

4 Solução para MT1

4.1 Definição formal

Uma Máquina de Turing (MT) padrão trata-se de um autômato determinístico que possui uma fita de trabalho de comprimento infinito sendo ilimitada à direita mas limitada à esquerda, de modo que a primeira célula da fita possui o símbolo especial «'», que não faz parte do alfabeto de entrada e não pode ocorrer em nenhuma outra célula, e a palavra de entrada da fita é iniciada a partir da segunda célula. Além de apenas ler o conteúdo apontado pelo cabeçote, também é possível realizar a escrita na célula, o que possibilita que a MT padrão também possa ser usada como transdutora, isto é, seja a palavra w na fita então a Máquina de Turing produz na própria fita a saída respectiva. (VIEIRA; BARBOSA, 2010)

Como é possível obter um algoritmo para realizar o cálculo do horário de chegada de um voo, considerando os fusos horários das cidades de origem e destino e o tempo de duração do voo, então tal problema é efetivamente computável, e pela tese de Church-Turing é computável por meio de uma MT. Desse modo, a Máquina de Turing para computar esse horário de chegada consiste em uma transdutora definida formalmente pela seguinte óctupla,

$$M = (E, \Sigma, \Gamma, <, v, \delta, i, F)$$

onde:

- E é um conjunto finito de estados, que devido a quantidade existente nessa MT os estados serão indicados nos módulos descritos mais a frente;
- $\Sigma \subseteq \Gamma$ é o alfabeto da palavra w de entrada, assim $\Sigma = \{1, L, O\}$
- Γ é o alfabeto que contém todos os símbolos que podem ocorrer na fita, assim $\Gamma = \{<, ', R, A, X, N, F, 1, L, O, R\}$, sendo que as aspas simples ' ' são apenas para destacar o símbolo de espaço.
- $<$ é o primeiro símbolo da fita, por se tratar do símbolo responsável por limitar a fita à esquerda ele não é alterado durante a computação;
- v representa o símbolo do branco, que no caso desta MT1 foi definido como o carácter de espaço, assim $v = '$;
- δ é uma função de transição, formalmente indicada por $\delta(q0, a) = [q1, b, D]$, onde $q0$ é o estado atual, $q1$ o estado seguinte, a o símbolo lido pelo cabeçote, b o símbolo escrito (substituindo a) e D a direção que o cabeçote é deslocado. Como foi utilizada uma ferramenta online que simula a execução de uma MT padrão, que é a turingmachine.io, então as funções de transição foram escritas de acordo com a linguagem dessa ferramenta. No entanto, apesar da função de transição possuir uma sintaxe diferente nessa ferramenta, tal sintaxe é equivalente com a definição formal de δ .
- i é um estado especial de E : o inicial, o registrador da MT padrão conterá esse estado na configuração inicial, então a unidade de controle da MT iniciará as operações a partir de i , assim $i = \text{put_R}; e$
- $F \subseteq E$ é o conjunto de estados finais, assim $F = \{ \text{erase_R} \}$.

Além da definição da máquina, é necessário determinar também como será a representação de uma instância desse problema. Para os valores numéricos foi utilizada a representação unária, contemplando o tempo dado em horas fechadas, sem minutos, e os múltiplos de 15, mas a longitude da cidade também abrange a localização se é leste ou oeste, o que será representado respectivamente pelos símbolos L ou O . Assim, define-se os parâmetros são denotado do seguinte modo:

- A duração do voo em horas = T ; Representação = $R<T>$.
- O horário em que o voo partiu (referente à cidade de origem) = P ; Representação = $R<P>;$
- Localização geográfica (leste ou oeste) = G ; Representação = $R<G>.$

- O fuso em graus da cidade de origem = Fp; Representação = R<Fp>;
- A longitude da cidade de origem = G1; Representação = R<G1> = R<G>R<Fp>;
- O fuso em graus da cidade de destino = Fc; Representação = R<Fc>;
- A longitude da cidade de destino = G2; Representação = R<G2> = R<G>R<Fp>;

Dessa forma, a palavra de entrada completa w foi definida como a concatenação dos parâmetros do seguinte modo:

$$w = R < T > R < P > R < G1 > R < G2 >$$

4.2 Módulos desenvolvidos

O cálculo do horário de chegada do voo pode ser realizado mediante um procedimento com as seguintes etapas:

1. Somar a hora de partida com a duração do voo;
2. Dividir por 15 as longitudes da cidade de destino e de origem;
3. Calcular a hora de chegada a depender do posicionamento identificado:
 - Caso a cidade de origem esteja a oeste então ocorrerá a subtração de seu fuso horário da soma realizada
 - Caso a cidade de origem esteja a leste então ocorrerá a subtração de seu fuso horário da soma realizada
4. Verificar se a hora calculada resultou em um valor válido, e caso não seja corrigir esse valor.

Desse modo, a Máquina de Turing padrão MT1 desenvolvida para computar o horário de chegada pode ser entendida por meio de módulos, onde cada módulo realiza uma das etapas indicadas para o cálculo. Além disso é necessário utilizar módulos auxiliares para realizar a manipulação dos símbolos na fita, tanto para que a palavra possua apenas a representação dos parâmetros de entrada (sem símbolos auxiliares de computo) como para que ocorra o cálculo correto. Em resumo, a MT1 possui os seguintes módulos:

1. Inicialização - O símbolo R é acrescentado na primeira célula após o final da palavra de entrada e o cabeçote é retornado para a primeira célula da fita;
2. Somar hora de partida e duração - Registra o resultado da soma após o R;
3. Divisor por 15 e decisão - Divide o fuso horário por 15 e decide qual será a operação correta;
4. Adição - Responsável por realizar a adição ao resultado
5. Subtração - Responsável por realizar a subtração ao resultado
6. Verificação e correção para o formato 24h - Identifica se o resultado está no formato correto e corrige caso não esteja

7. Finalização - Remove todos os símbolos residuais e desloca o resultado para o início da fita.

Vale frisar que o turingmachine.io é uma ferramenta que gera um diagrama automaticamente de acordo com o desenvolvido na linguagem e sua interface permite visualizar cada configuração instantânea. Assim, nas imagens a seguir, que ilustrarão o funcionamento dos módulos, o estado atual é indicado por meio do círculo amarelo em torno do mesmo, e a posição do cabeçote na fita é indicado por um quadrado de mesma cor em torno da célula. Além disso, os círculos que representam os estados foram reposicionados com o objetivo de facilitar a visualização. Conforme representação definida para a palavra de entrada, os parâmetros utilizados no exemplo possuem os seguintes valores:

- $T = 1; R < T > = R < 1 > = 1;$
- $P = 1; R < P > = R < 1 > = 1;$
- $G1 = 15^\circ \text{ Leste}; R < G1 > = R < G > R < Fp > = L111111111111111111;$
- $G2 = 15^\circ \text{ Oeste}; R < G2 > = R < G > R < Fp > = O111111111111111111;$

4.2.1 Inicialização

Este módulo abrange as funções de transição definidas para os estados put_R e restart. Tal como a denominação do módulo busca indicar, ele possui o estado inicial, logo é o ponto de partida da computação e trata-se de um módulo auxiliar para as próximas operações. Ao concluir a funcionalidade de acrescentar o R e retornar para o início da fita a MT1 avançará para o estado do próximo módulo (start_add).

```

1  input: '<11L11111111111110111111111111111111111'
2  blank: ''
3  start state: put_R
4  table:
5
6  put_R:
7    < : {write: <, R}
8    1 : {write : 1, R}
9    0 : {write: 0, R}
10   L : {write: L, R}
11   ' ' : {write: R, L: restart}
12
13  restart:
14    1 : {write : 1, L}
15    0 : {write: 0, L}
16    L : {write: L, L}
17    < : {write: <, R: start_add}
18
19  start_add:
20
21
22
```

Figura 1 – Código das funções de transição na inicialização

4.2.2 Somar hora de partida e duração

Como estratégia de desenvolvimento desta MT1 o resultado das operações realizadas ficam registrados após o símbolo R, assim, este módulo possui a funcionalidade de efetivar a soma dos parâmetros de hora de partida e duração do voo, preenchendo o resultado nas

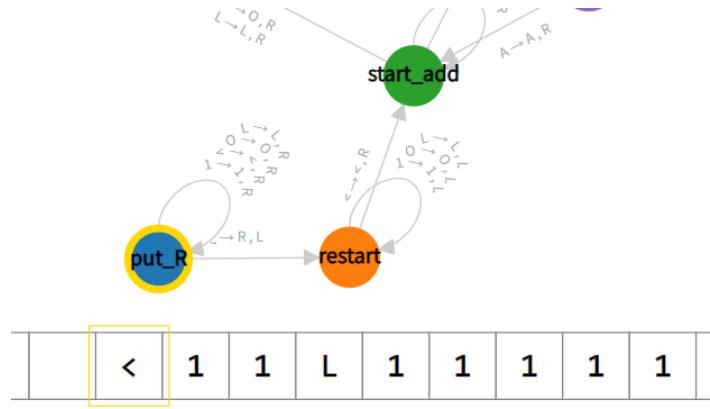


Figura 2 – Configuração inicial da MT1

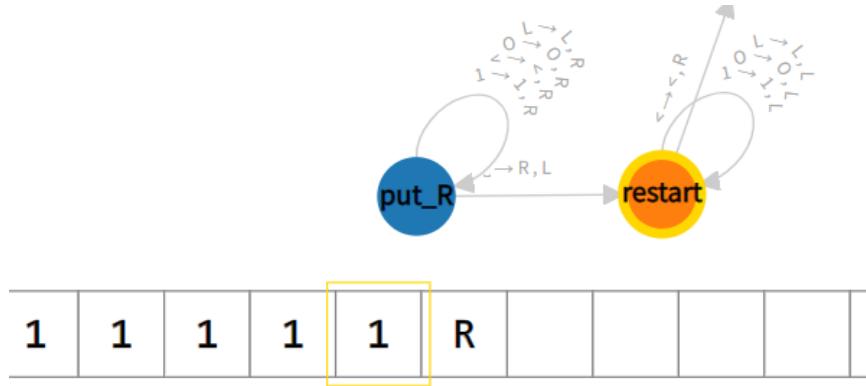


Figura 3 – Símbolo R acrescentado na primeira célula após o final da palavra de entrada

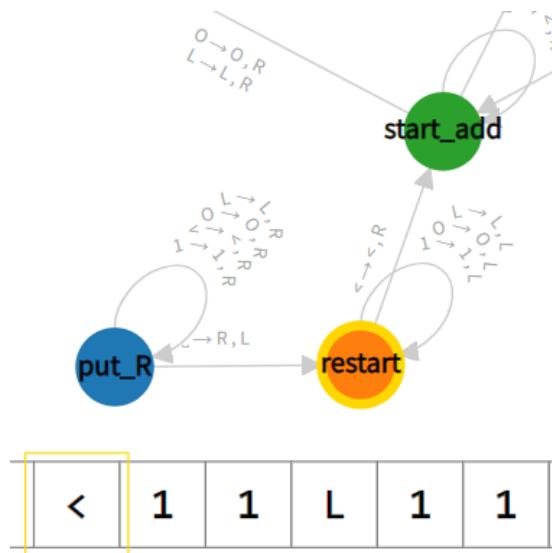


Figura 4 – Configuração instantânea da última função de transição deste módulo

células apóis o R. Vale destacar que, na representação da entrada, R<T> e R<P> foram concatenados com o objetivo de facilitar essa operação.

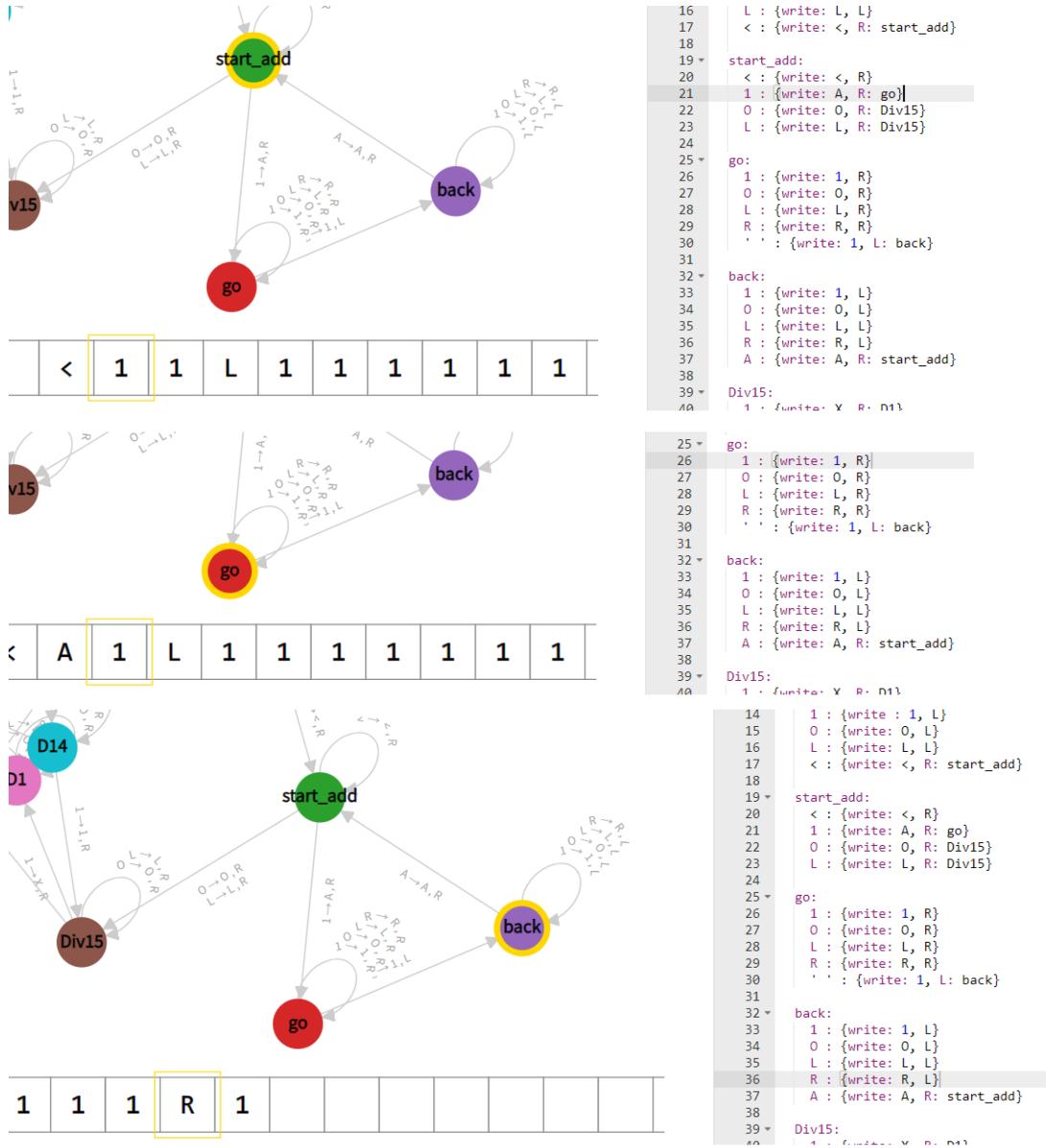


Figura 5 – Primeiras funções de transição da soma da hora de partida com a duração

Assim, este módulo abrange as funções de transição definidas para os estados start_add, go e back, elas farão com que todos os 1 que estão no início da fita sejam substituídos por A, e reescritos nas células apóis o R. O módulo avançará para o estado do próximo módulo (Div15) quando for reconhecido o primeiro símbolo de R<G1>.

4.2.3 Divisor por 15

Como o resultado da hora de chegada irá depender do fuso horário, sendo que cada 15 graus representa 1 hora, então é necessário realizar essa divisão no valor da longitude. Logo, este módulo é responsável pelo resultado em unário dessa divisão.

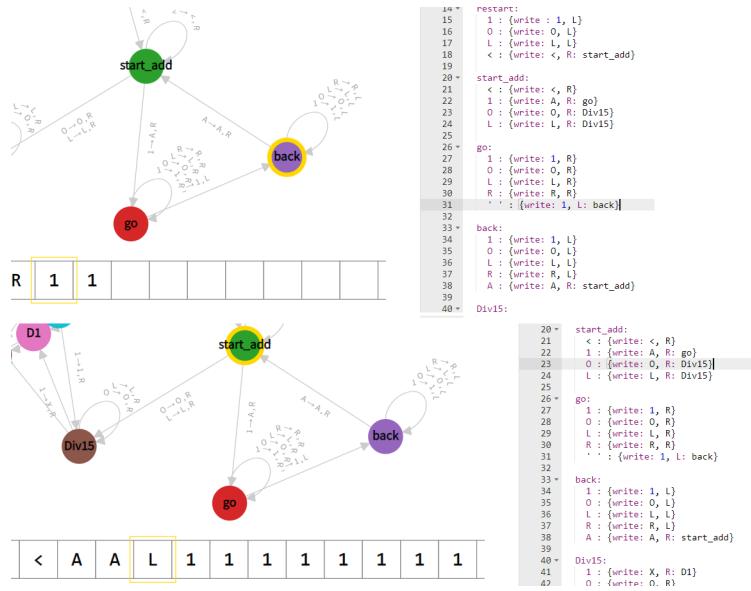


Figura 6 – Finalização da funcionalidade

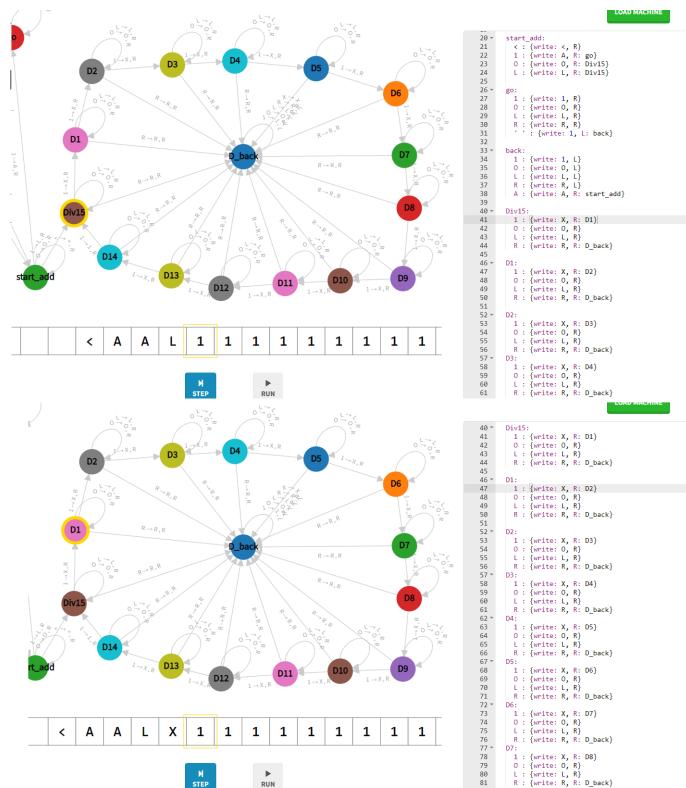


Figura 7 – Início da divisão por 15

Iniciando pelo estado Div15, em resumo cada estado substitui o 1 por um X para realizar a contagem de mais uma unidade e quando 15 unidades forem contadas o 1 não será substituído.

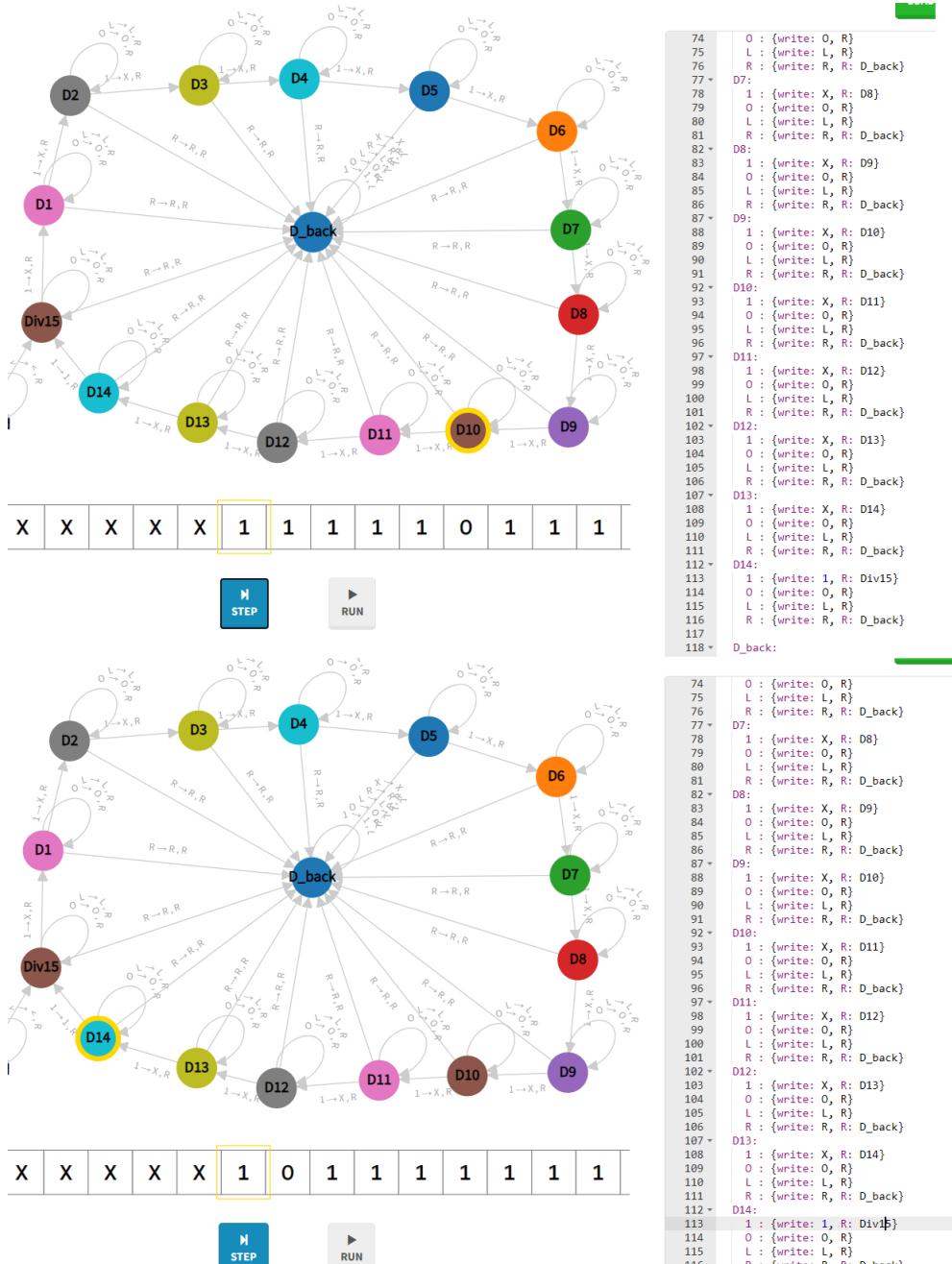


Figura 8 – Continuação da divisão por 15 em $R < G1 >$

Esse procedimento também já é realizado para $R < G2 >$ e ocorre do mesmo modo. O final dessa contagem para divisão ocorre quando o cabeçote chega na célula de R, nesse momento ocorre a transição para o estado que retorna o cabeçote para o início de $R < G1 >$.

É em F_choice que é definido qual será o próximo módulo, pois nesse momento o cabeçote está na célula de $R < G1 >$ que indica se a cidade está a leste ou oeste.

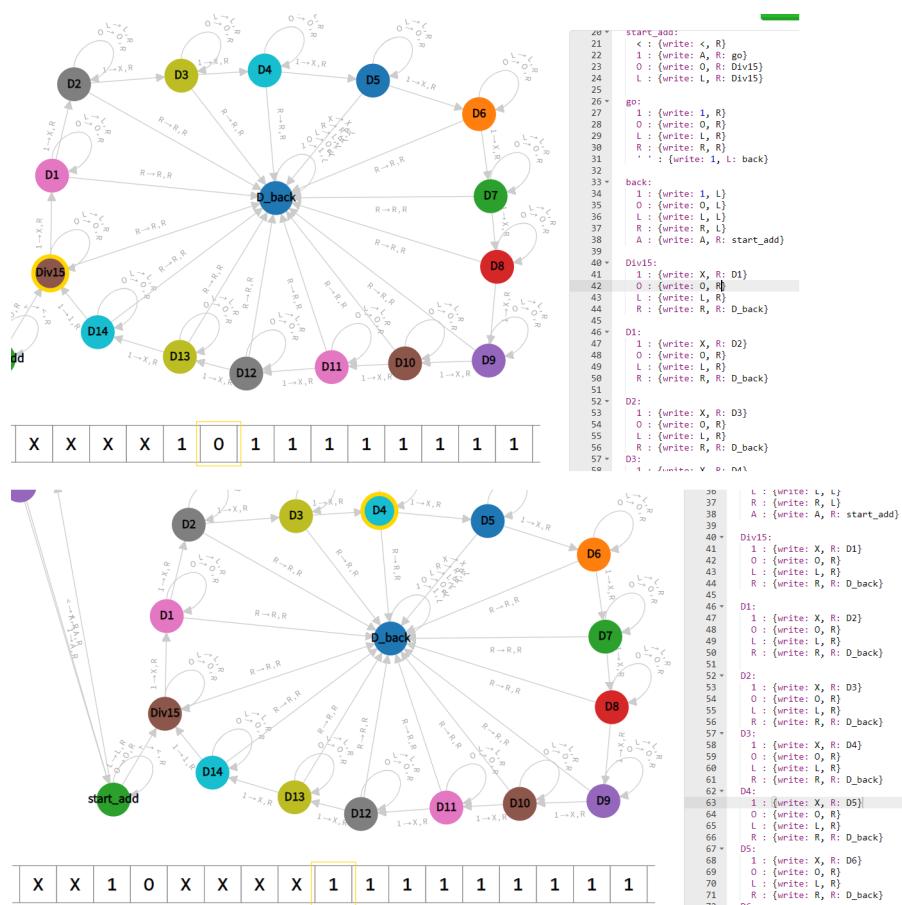


Figura 9 – Avanço da divisão para $R < G2 >$

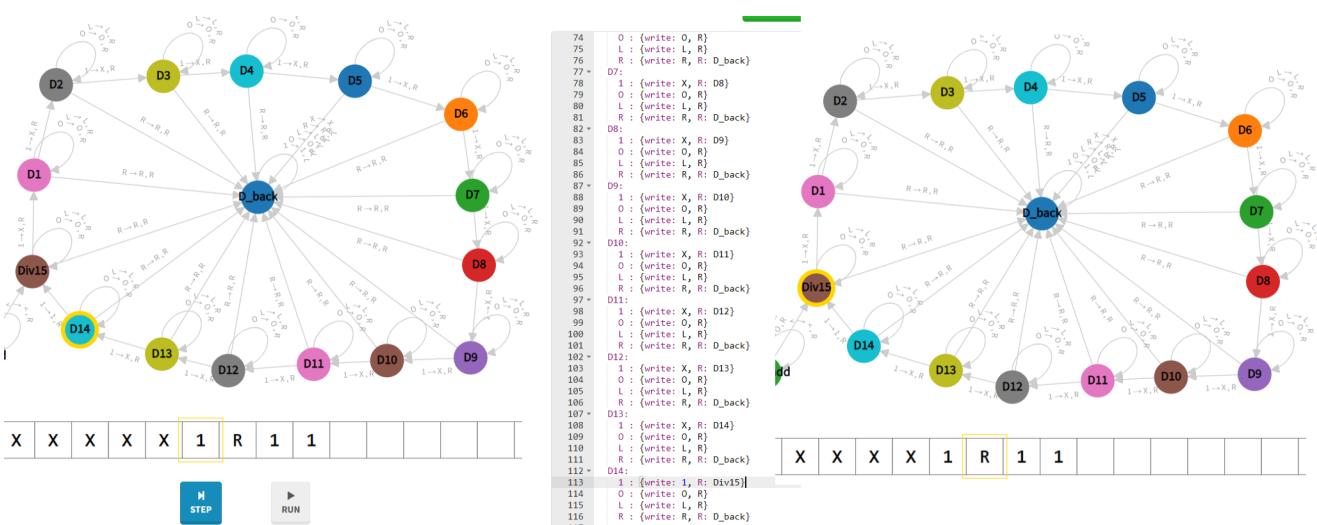
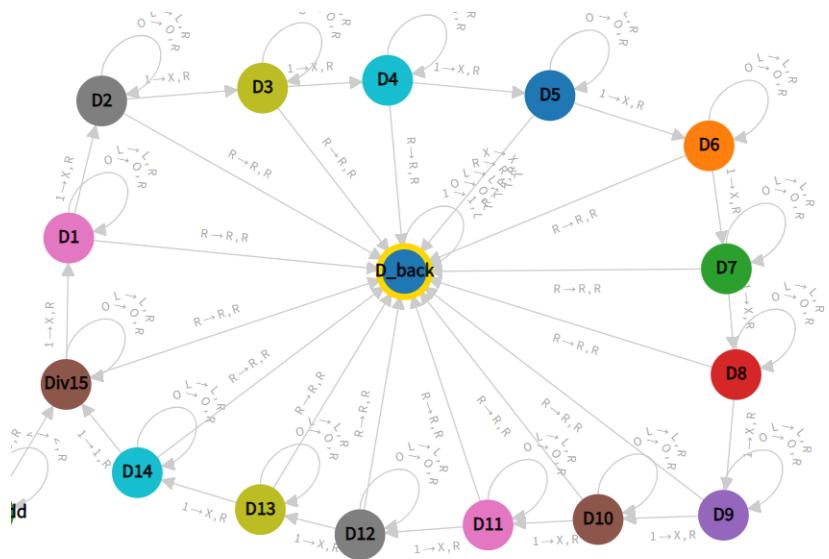
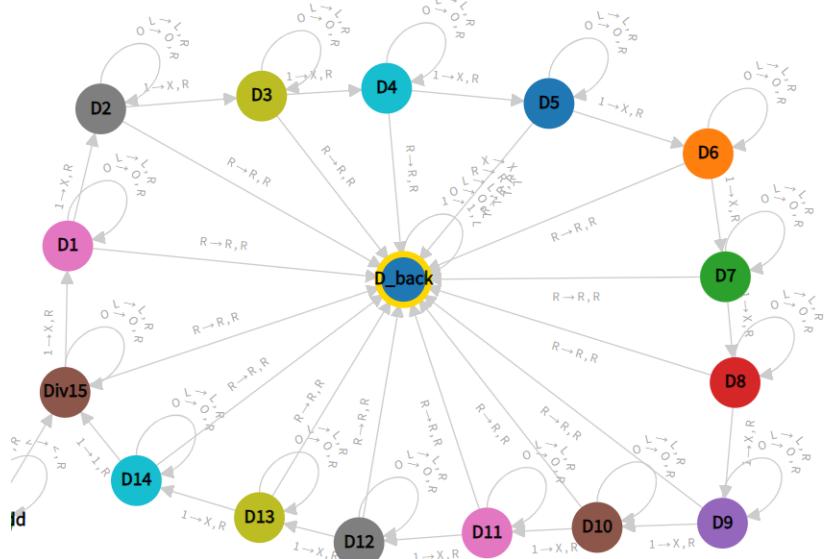


Figura 10 – Divisão por 15 próxima de suas etapas finais



X	X	X	1	R	1	1									
---	---	---	---	---	---	---	--	--	--	--	--	--	--	--	--

STEP RUN



X	X	X	X	X	X	1	0	X	X	X	X	X	X		
---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--

```

74    O : {write: O, R}
75    L : {write: L, R}
76    R : {write: R, R: D_back}
77 ▾ D7:
78    1 : {write: X, R: D8}
79    O : {write: O, R}
80    L : {write: L, R}
81    R : {write: R, R: D_back}
82 ▾ D8:
83    1 : {write: X, R: D9}
84    O : {write: O, R}
85    L : {write: L, R}
86    R : {write: R, R: D_back}
87 ▾ D9:
88    1 : {write: X, R: D10}
89    O : {write: O, R}
90    L : {write: L, R}
91    R : {write: R, R: D_back}
92 ▾ D10:
93    1 : {write: X, R: D11}
94    O : {write: O, R}
95    L : {write: L, R}
96    R : {write: R, R: D_back}
97 ▾ D11:
98    1 : {write: X, R: D12}
99    O : {write: O, R}
100   L : {write: L, R}
101   R : {write: R, R: D_back}
102 ▾ D12:
103   1 : {write: X, R: D13}
104   O : {write: O, R}
105   L : {write: L, R}
106   R : {write: R, R: D_back}
107 ▾ D13:
108   1 : {write: X, R: D14}
109   O : {write: O, R}
110   L : {write: L, R}
111   R : {write: R, R: D_back}
112 ▾ D14:
113   1 : {write: 1, R: Div15}
114   O : {write: O, R}
115   L : {write: L, R}
116   R : {write: R, R: D_back}
117   D_back:
118   1 : {write: 1, L}
119   O : {write: O, L}
120   L : {write: L, L}
121   R : {write: R, L}
122   Y : {write: X, I}
123

```

```

86   R : {write: R, R: D_back}
87 ▾ D9:
88   1 : {write: X, R: D10}
89   O : {write: O, R}
90   L : {write: L, R}
91   R : {write: R, R: D_back}
92 ▾ D10:
93   1 : {write: X, R: D11}
94   O : {write: O, R}
95   L : {write: L, R}
96   R : {write: R, R: D_back}
97 ▾ D11:
98   1 : {write: X, R: D12}
99   O : {write: O, R}
100  L : {write: L, R}
101  R : {write: R, R: D_back}
102 ▾ D12:
103  1 : {write: X, R: D13}
104  O : {write: O, R}
105  L : {write: L, R}
106  R : {write: R, R: D_back}
107 ▾ D13:
108  1 : {write: X, R: D14}
109  O : {write: O, R}
110  L : {write: L, R}
111  R : {write: R, R: D_back}
112 ▾ D14:
113  1 : {write: 1, R: Div15}
114  O : {write: O, R}
115  L : {write: L, R}
116  R : {write: R, R: D_back}
117   D_back:
118  1 : {write: 1, L}
119  O : {write: O, L}
120  L : {write: L, L}
121  R : {write: R, L}
122  X : {write: X, I}
123

```

Figura 11 – Retornando para o começo de $R < G1 >$

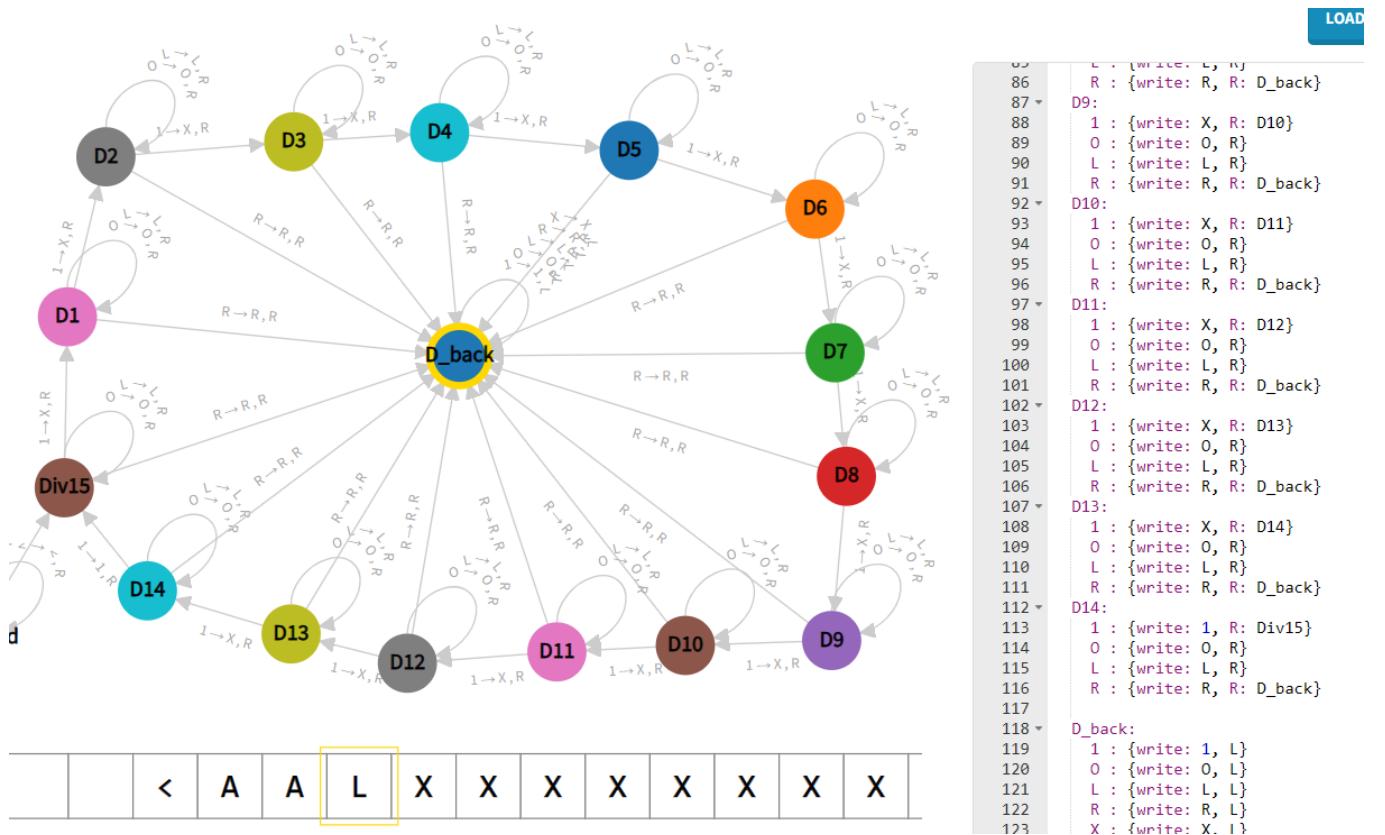


Figura 12 – Finalização da divisão

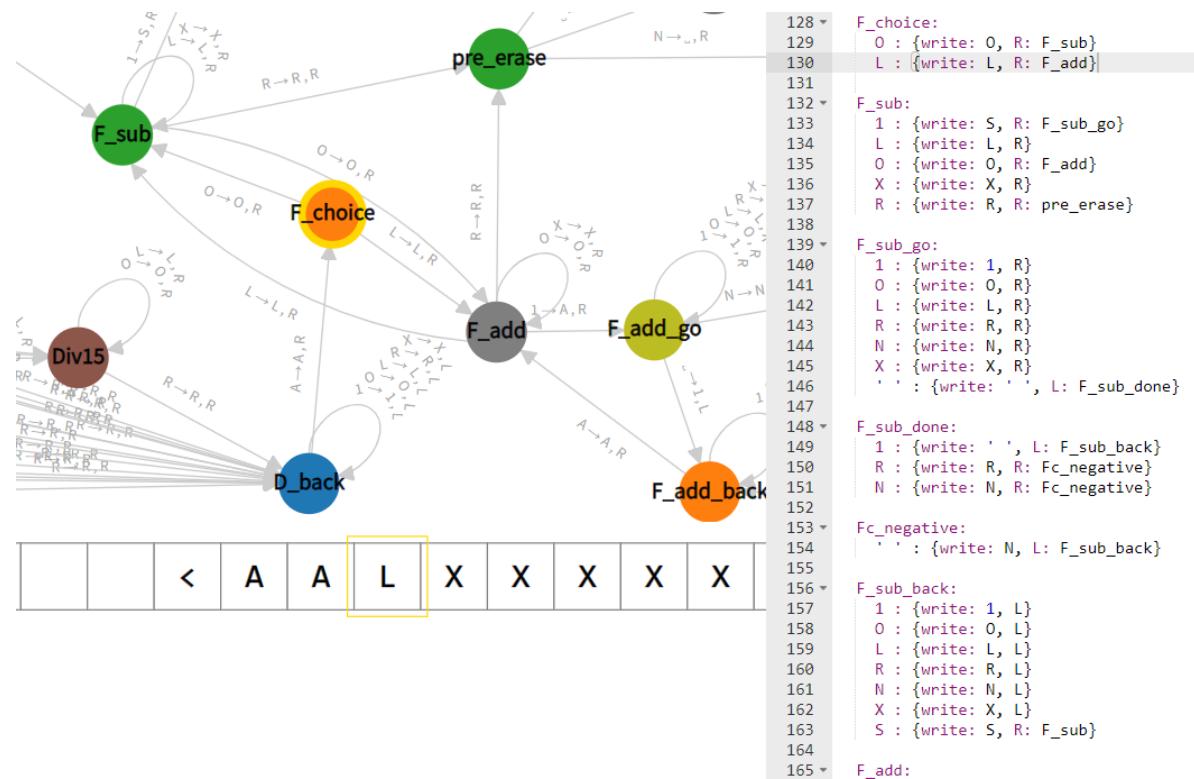


Figura 13 – Decisão da próxima operação

Como a operação necessária para calcular o horário de chegada irá depender da posição entre as duas cidades então é essa a transição que possibilita a MT1 decidir qual será o cálculo correto.

4.2.4 Adição

O módulo de adição se inicia no estado **F_add**, que é diretamente chamado quando a cidade de origem está a leste. O **F_add** avança por todos os X escritos no módulo anterior até encontrar um 1, nesse momento o 1 é sobreescrito pelo A e o **F_add_go** avança pela fita até escrever o 1 na primeira célula vazia após o final da palavra. Assim a adição dessa unidade é efetivamente realizada e o **F_add_back** é responsável por voltar até o ponto onde o computo foi realizado.

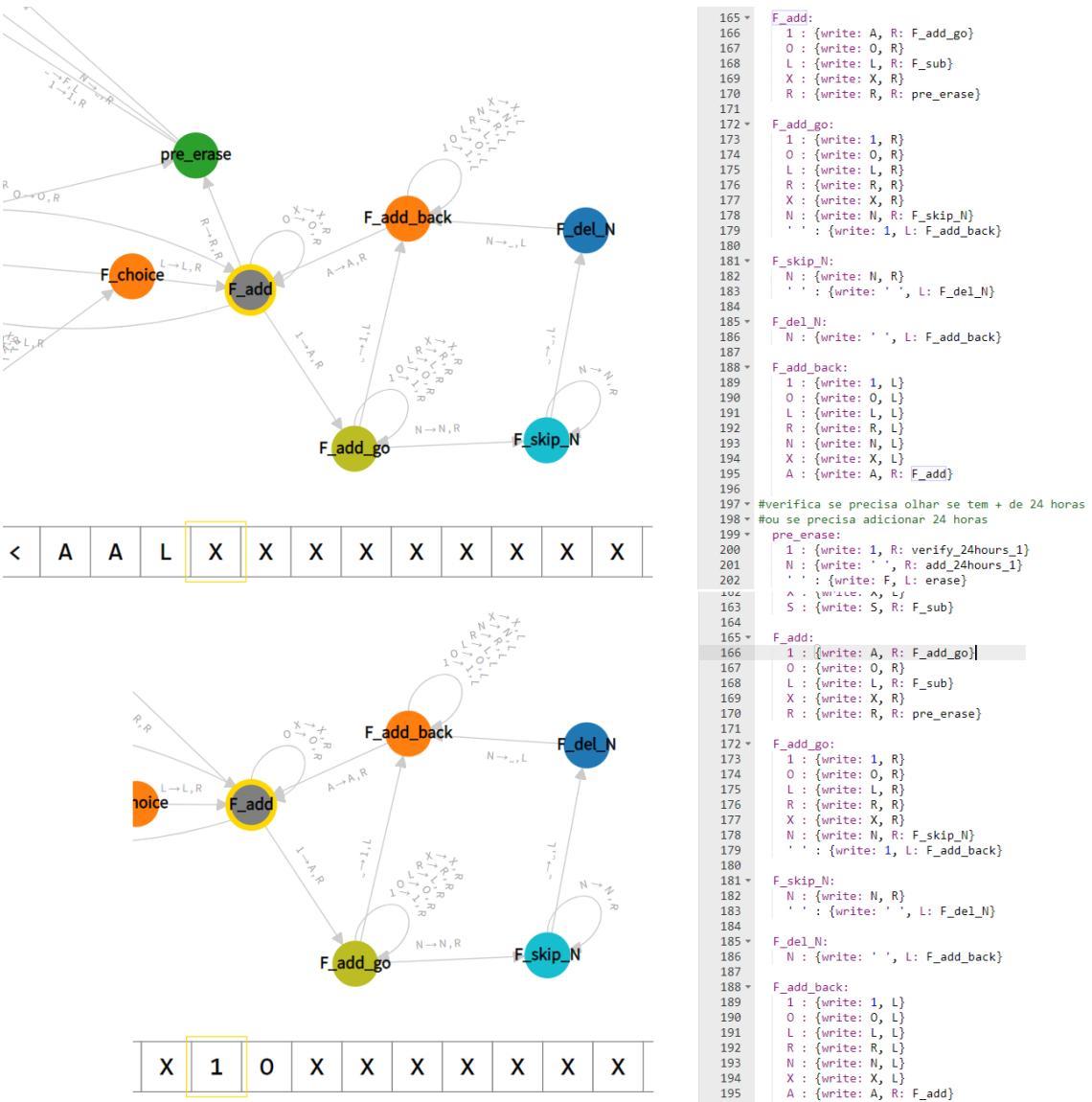
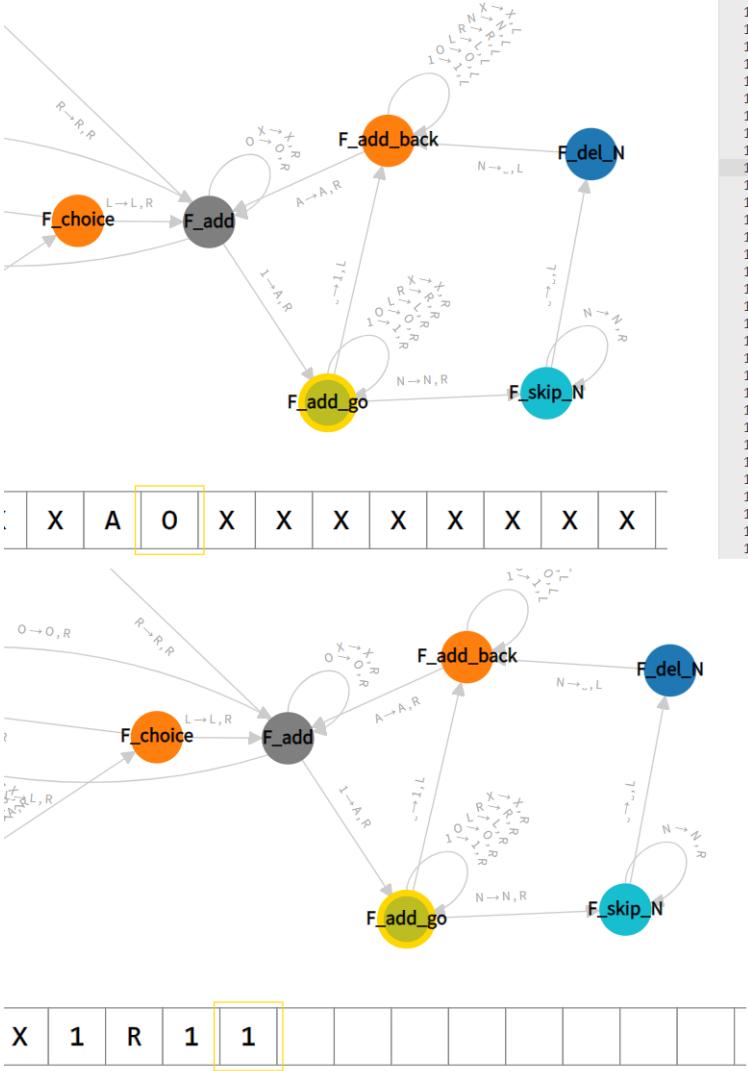


Figura 14 – Início do módulo da adição

Esse procedimento se repete até que a primeira célula de R<G2> seja identificada. Nesse momento é necessário realizar uma nova decisão de qual será a próxima operação



```

164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196

F_add:
  1 : {write: A, R: F_add_go}
  0 : {write: O, R}
  L : {write: L, R: F_sub}
  X : {write: X, R}
  R : {write: R, R: pre_erase}

F_add_go:
  1 : {write: 1, R}
  0 : {write: O, R}
  L : {write: L, R}
  R : {write: R, R}
  X : {write: X, R}
  N : {write: N, R: F_skip_N}
  ' ' : {write: 1, L: F_add_back}

F_skip_N:
  N : {write: N, R}
  ' ' : {write: ' ', L: F_del_N}

F_del_N:
  N : {write: ' ', L: F_add_back}

F_add_back:
  1 : {write: 1, L}
  0 : {write: O, L}
  L : {write: L, L}
  R : {write: R, L}
  N : {write: N, L}
  X : {write: X, L}
  A : {write: A, R: F_add}

169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196

  X : {write: X, R}
  R : {write: R, R: pre_erase}

F_add_go:
  1 : {write: 1, R}
  0 : {write: O, R}
  L : {write: L, R}
  R : {write: R, R}
  X : {write: X, R}
  N : {write: N, R: F_skip_N}
  ' ' : {write: 1, L: F_add_back}

F_skip_N:
  N : {write: N, R}
  ' ' : {write: ' ', L: F_del_N}

F_del_N:
  N : {write: ' ', L: F_add_back}

F_add_back:
  1 : {write: 1, L}
  0 : {write: O, L}
  L : {write: L, L}
  R : {write: R, L}
  N : {write: N, L}
  X : {write: X, L}
  A : {write: A, R: F_add}

197 #verifica se precisa olhar se tem + de 24 horas
198 #ou se precisa adicionar 24 horas

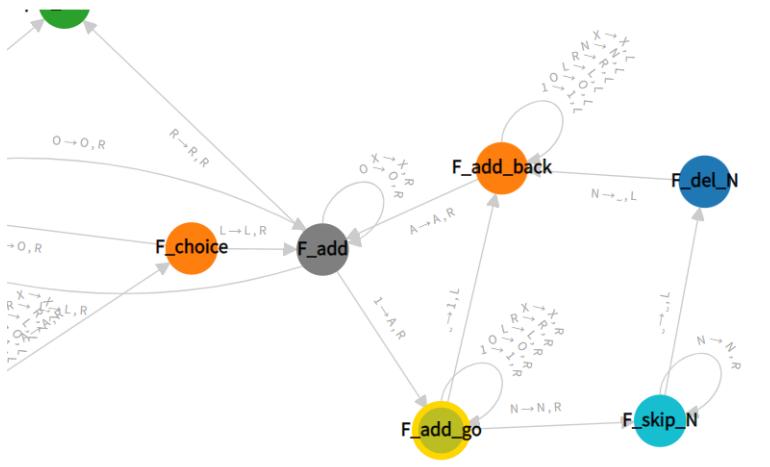
```

Figura 15 – Transições intermediárias

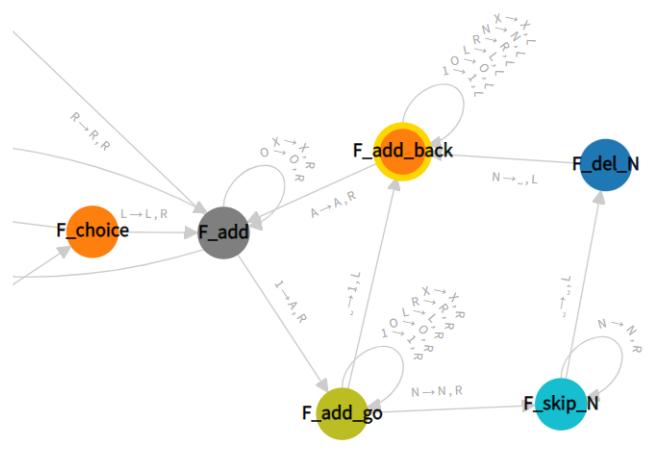
a ser realizada. Caso a cidade de destino esteja a leste então a MT1 irá para o primeiro estado do módulo de subtração (F_sub). Se a cidade de destino estiver a oeste então a operação de adição é realizada novamente, nesse caso este módulo finaliza quando o R é identificado no estado F_add, assim a MT1 avança para o primeiro estado do fluxo de verificação e correção para o formato 24h (pre_erase).

4.2.5 Subtração

A lógica envolvida nas funções de transição que constituem o módulo de subtração são muito semelhantes ao do módulo de adição, a principal diferença, além do nome utilizado para designá-las, é que em vez de acrescentar um 1, cada nova repetição irá substituir um 1 existente pelo símbolo em branco, além disso o símbolo de N auxilia no controle da MT1 caso a operação resulte em um valor negativo.



1	R	1	1								
---	---	---	---	--	--	--	--	--	--	--	--



(1	R	1	1	1						
---	---	---	---	---	---	--	--	--	--	--	--

```

164
165 F_add:
166   1 : {write: A, R: F_add_go}
167   0 : {write: O, R}
168   L : {write: L, R: F_sub}
169   X : {write: X, R}
170   R : {write: R, R: pre_erase}
171
172 F_add_go:
173   1 : {write: 1, R}
174   O : {write: O, R}
175   L : {write: L, R}
176   R : {write: R, R}
177   X : {write: X, R}
178   N : {write: N, R: F_skip_N}
179   ' ' : {write: 1, L: F_add_back}
180
181 F_skip_N:
182   N : {write: N, R}
183   ' ' : {write: ' ', L: F_del_N}
184
185 F_del_N:
186   ' N : {write: ' ', L: F_add_back}
187
188 F_add_back:
189   1 : {write: 1, L}
190   O : {write: O, L}
191   L : {write: L, L}
192   R : {write: R, L}
193   N : {write: N, L}
194   X : {write: X, L}
195   A : {write: A, R: F_add}
196
197 #verifica se precisa olhar se tem + de 24 horas

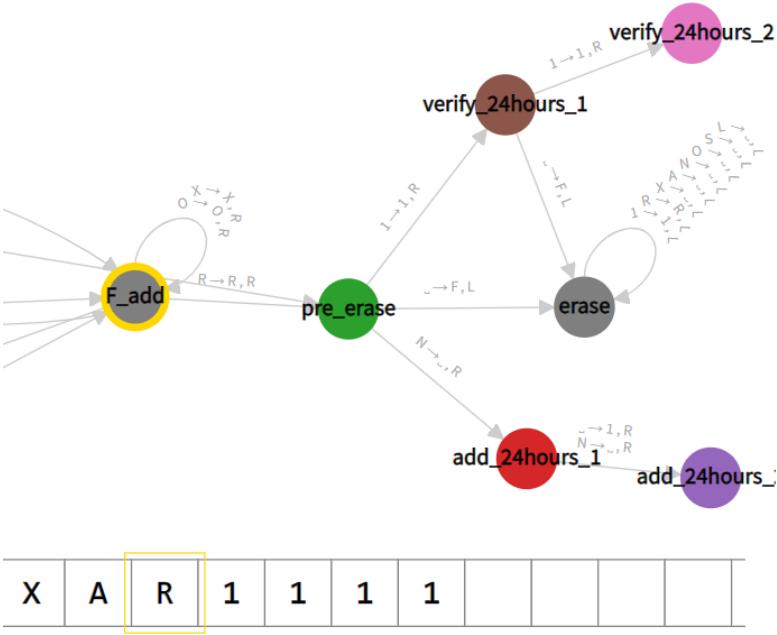
```

```

188 F_add_back:
189   1 : {write: 1, L}
190   O : {write: O, L}
191   L : {write: L, L}
192   R : {write: R, L}
193   N : {write: N, L}
194   X : {write: X, L}
195   A : {write: A, R: F_add}
196
197 #verifica se precisa olhar se tem + de 24 horas
198 #ou se precisa adicionar 24 horas
199 pre_erase:
200   1 : {write: 1, R: verify_24hours_1}
201   N : {write: ' ', R: add_24hours_1}
202   ' ' : {write: F, L: erase}
203
204 #para caso esteja negativo e necessita add 24 horas
205 add_24hours_1:
206   N : {write: ' ', R: add_24hours_2}
207   ' ' : {write: 1, R: add_24hours_2}
208 add_24hours_2:
209   #N: {write: ' ', R: add_24hours_3}
210   '# ' : {write: 1, R: add_24hours_3}
211
212 #verifica as 24h
213 verify_24hours_1:
214   1 : {write: 1, R: verify_24hours_2}
215   ' ' : {write: F, L: erase}
216 verify_24hours_2:
217   '#1: {write: 1, R: verify_24hours_3}
218   '# ' : {write: F, L: erase}
219
220 erase:

```

Figura 16 – Adição realizada

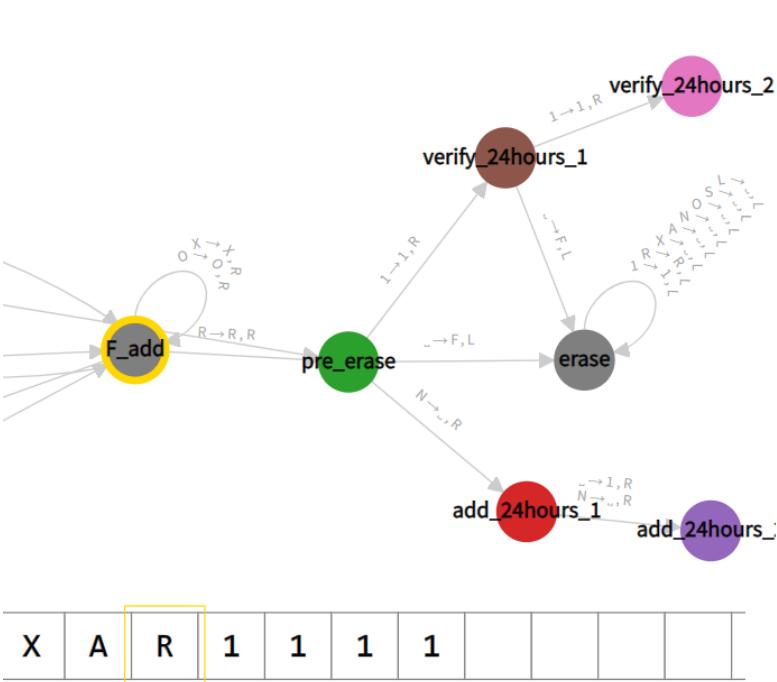


```

165 *
166   F_add:
167     1 : {write: A, R: F_add_go}
168     0 : {write: O, R}
169     L : {write: L, R: F_sub}
170     X : {write: X, R}
171     R : {write: R, R: pre_erase}
172 *
173   F_add_go:
174     1 : {write: 1, R}
175     0 : {write: O, R}
176     L : {write: L, R}
177     R : {write: R, R}
178     X : {write: X, R}
179     N : {write: N, R: F_skip_N}
180     '' : {write: 1, L: F_add_back}
181 *
182   F_skip_N:
183     N : {write: N, R}
184     '' : {write: '', L: F_del_N}
185 *
186   F_del_N:
187     N : {write: '', L: F_add_back}
188 *
189   F_add_back:
190     1 : {write: 1, L}
191     0 : {write: O, L}
192     L : {write: L, L}
193     R : {write: R, L}
194     N : {write: N, L}
195     X : {write: X, L}
196     A : {write: A, R: F_add}
197 *
198   #verifica se precisa olhar se tem + de 24 horas
199   #ou se precisa adicionar 24 horas
200   pre_erase:
201     1 : {write: 1, R: verify_24hours_1}
202     N : {write: '', R: add_24hours_1}
203     '' : {write: F, L: erase}
204

```

Figura 17 – Finalização da adição



```

165 *
166   F_add:
167     1 : {write: A, R: F_add_go}
168     0 : {write: O, R}
169     L : {write: L, R: F_sub}
170     X : {write: X, R}
171     R : {write: R, R: pre_erase}
172 *
173   F_add_go:
174     1 : {write: 1, R}
175     0 : {write: O, R}
176     L : {write: L, R}
177     R : {write: R, R}
178     X : {write: X, R}
179     N : {write: N, R: F_skip_N}
180     '' : {write: 1, L: F_add_back}
181 *
182   F_skip_N:
183     N : {write: N, R}
184     '' : {write: '', L: F_del_N}
185 *
186   F_del_N:
187     N : {write: '', L: F_add_back}
188 *
189   F_add_back:
190     1 : {write: 1, L}
191     0 : {write: O, L}
192     L : {write: L, L}
193     R : {write: R, L}
194     N : {write: N, L}
195     X : {write: X, L}
196     A : {write: A, R: F_add}
197 *
198   #verifica se precisa olhar se tem + de 24 horas
199   #ou se precisa adicionar 24 horas
200   pre_erase:
201     1 : {write: 1, R: verify_24hours_1}
202     N : {write: '', R: add_24hours_1}
203     '' : {write: F, L: erase}
204

```

Figura 18 – Módulo de subtração

4.2.6 Verificação e correção para o formato 24h

Este módulo tem início no estado pre_erase, nesse momento há 3 casos que podem ocorrer, tal como indicado a seguir, mas ao final de todos eles a MT1 avançará para o primeiro estado do módulo de finalização (_erase).

1. Caso o resultado seja positivo então será feita a verificação da quantidade de 1's inseridos por meio dos estados verify_24hours. Se ele possui menos de 23 1's então basta avançar para a finalização. No entanto, se no estado verify_24hours_23 o cabeçote fizer a leitura do 1 então é necessário subtrair os 1's excedentes, avançando primeiro para o estado auxiliar go_to_end_of_erase e em seguida para a sequência de estados subtract_24hours_1 ao subtract_24hours_24 que de fato fazem a remoção;
2. Caso o resultado tenha sido negativo então é necessário adicionar 1's, esse procedimento é realizado entre os estados add_24hours_1 até add_24hours_23;
3. Caso o símbolo do branco seja identificado então avança diretamente para a finalização.

A imagem a seguir busca ilustrar de modo simplificado as possibilidades de verificação e correção descritas anteriormente, devido o espaço disponível para elaborar o diagrama apenas alguns estados dos casos anteriores são exibidos.

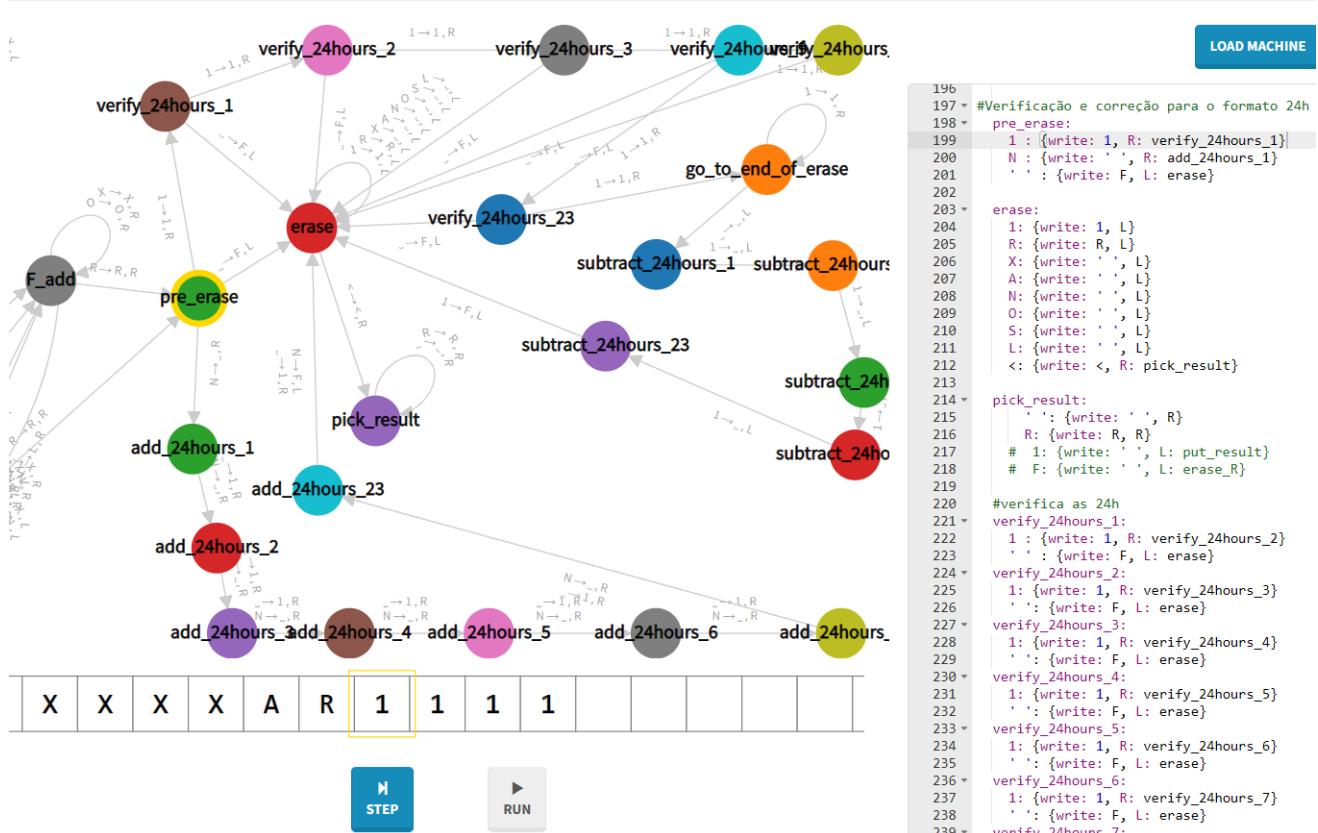


Figura 19 – Visão parcial do módulo de verificação e correção

Como o exemplo utilizado resultou em um horário válido então no estado verify_24hours_4 a MT1 avançará para o _erase.

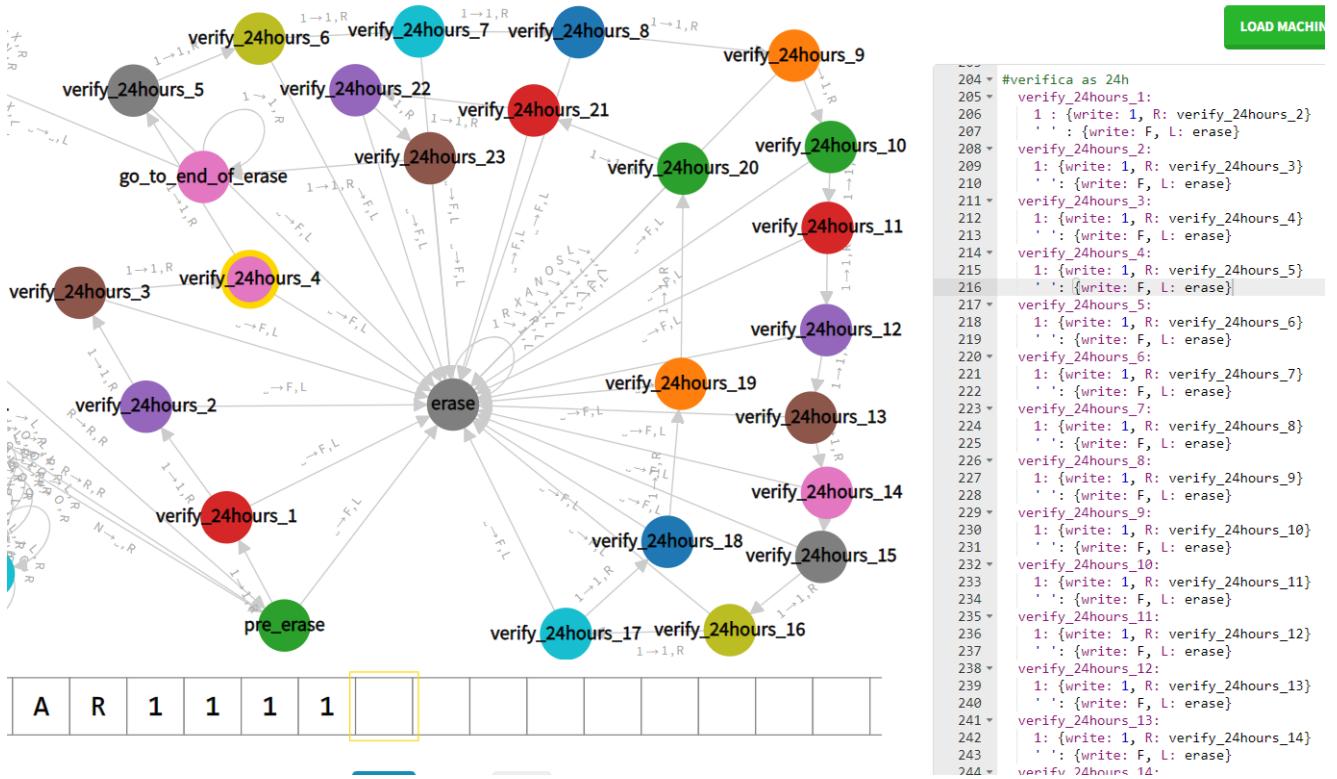


Figura 20 – Verificação do horário válido

4.2.7 Finalização

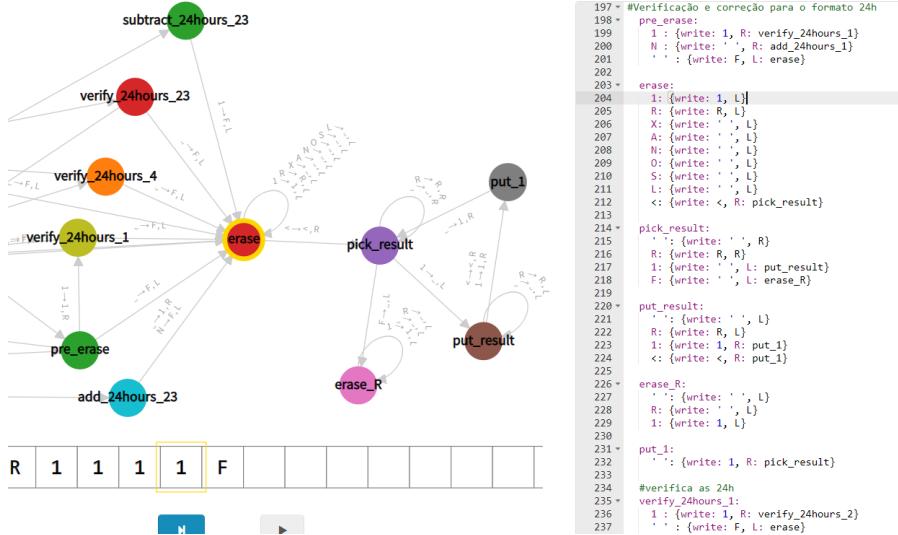


Figura 21 – Iniciando o fluxo de finalização

Tal como indicado anteriormente o fluxo de finalização ocorre após o cálculo da hora válida, e trata-se de um módulo auxiliar que é responsável por manipular os símbolos da fita para que ela possua apenas o resultado no formato unário. Assim, é no estado `_erase` que todos os símbolos residuais do computo são substituídos pelo símbolo em branco.

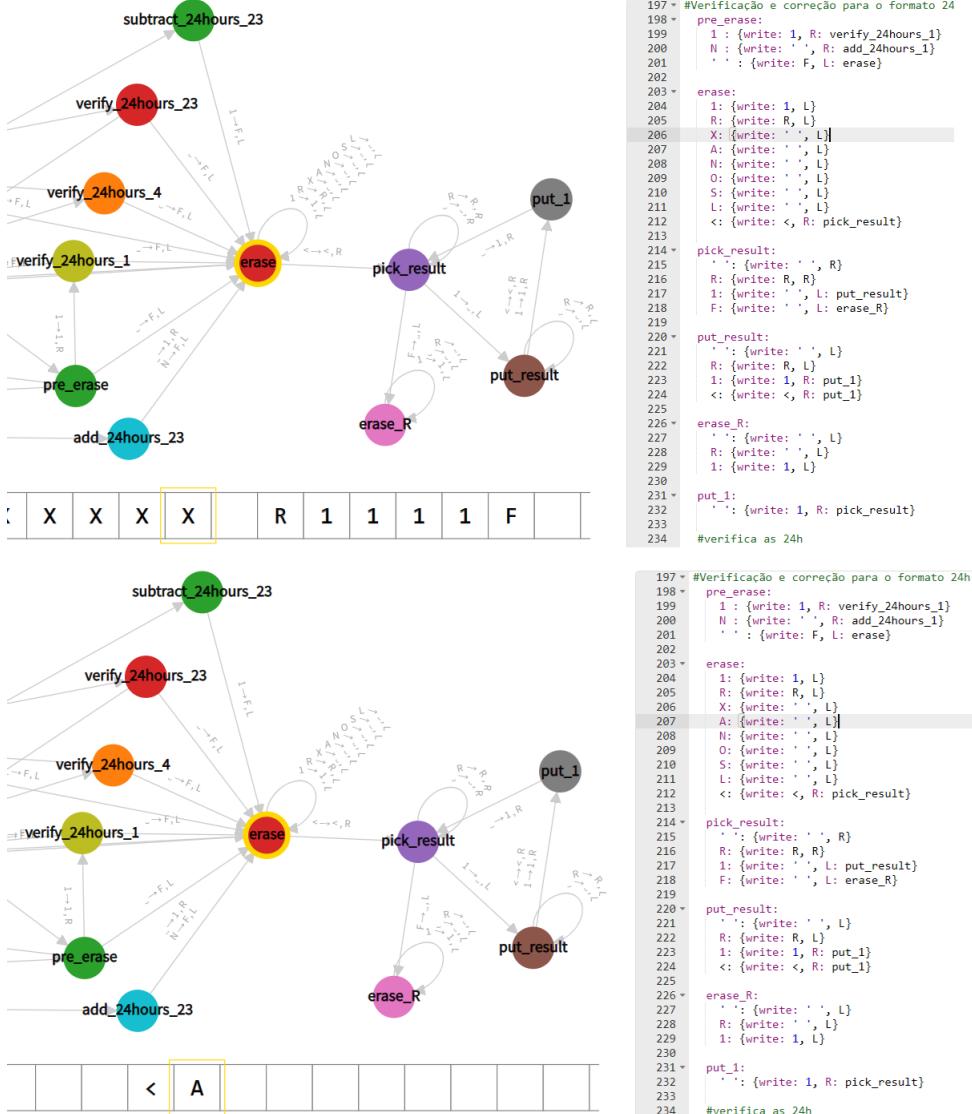


Figura 22 – Remoção dos símbolos residuais

Essa etapa é finalizada quando todos os símbolos anteriores ao R foram substituídos e o cabeçote está na primeira célula.

Feito isso os estados `pick_result`, `put_result`, `put_1` e `erase_R` são responsáveis por deslocar o resultado que está após o R para o início da fita, e remover os símbolos auxiliares R e F, concluindo assim a transdução da MT1.

O estado final é atingido em `erase_R` quando o cabeçote retorna para o símbolo delimitador a esquerda da fita.

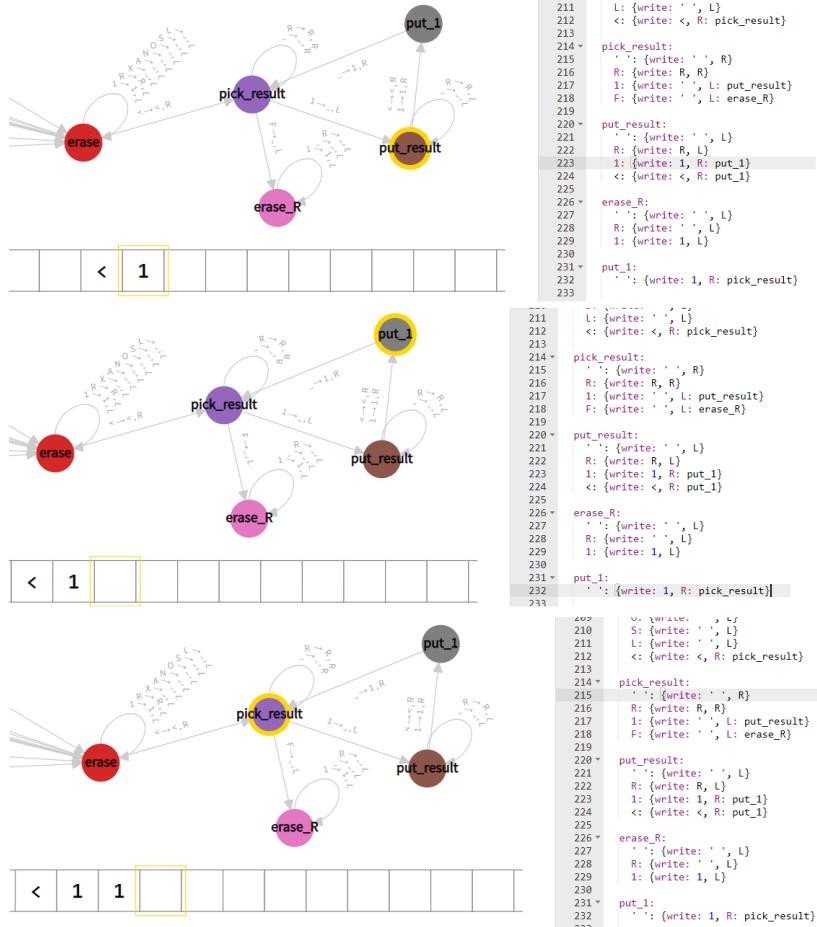


Figura 23 – Deslocamento do resultado

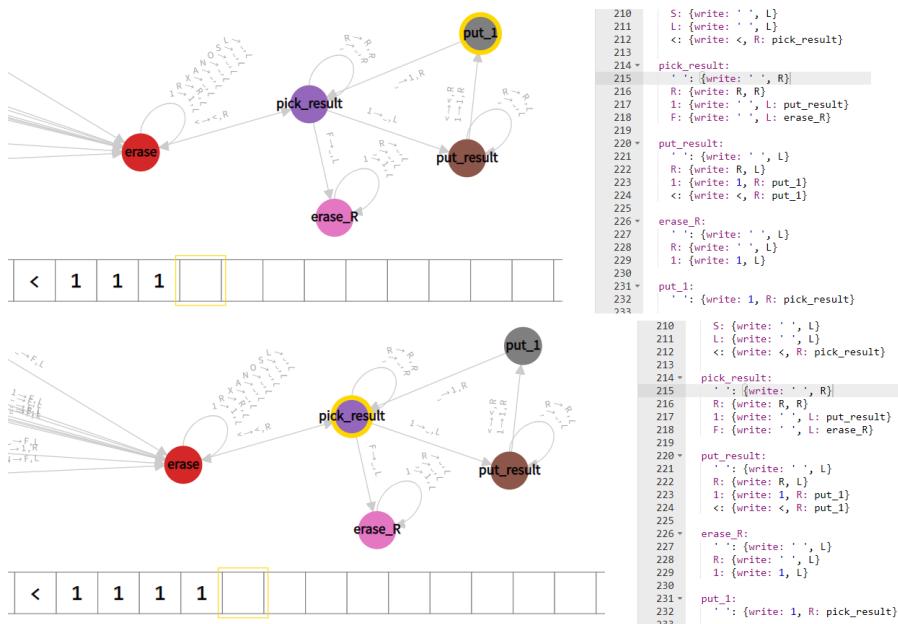


Figura 24 – Finalização do deslocamento

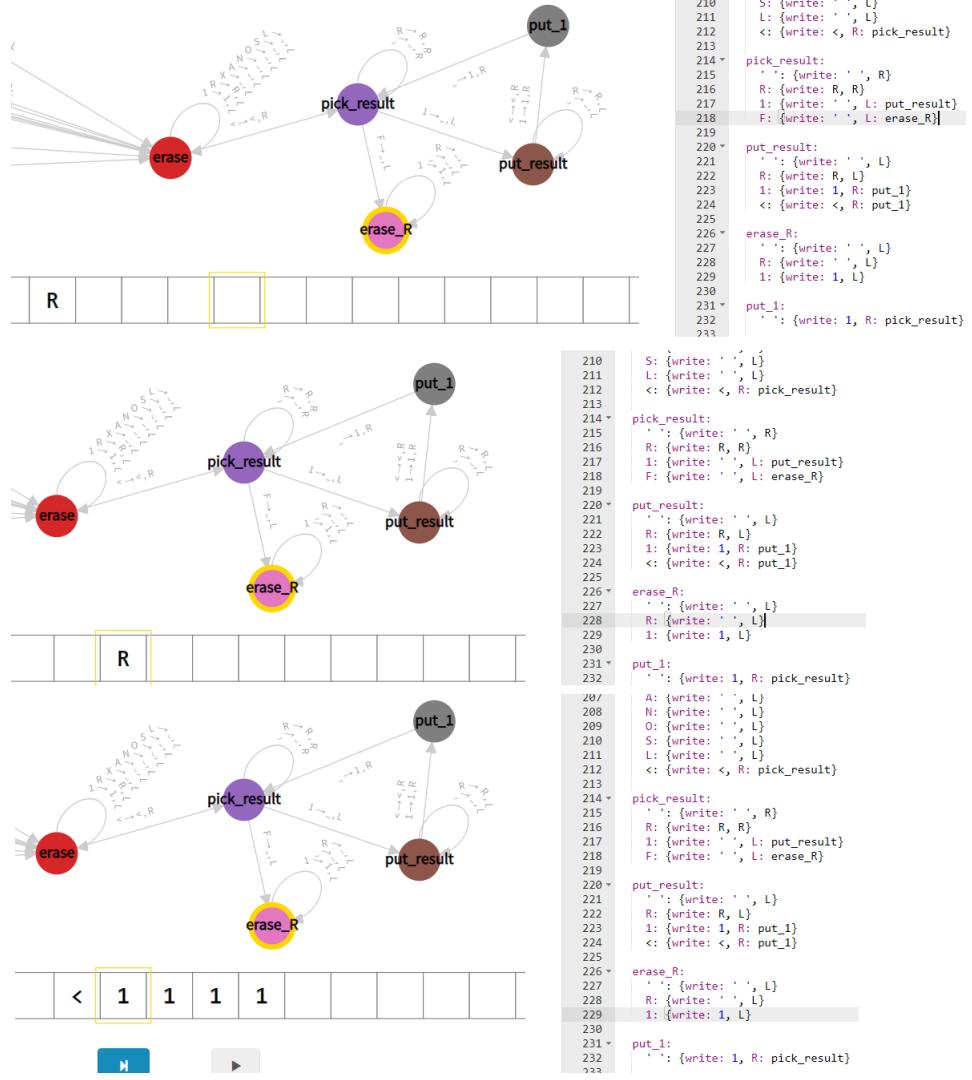


Figura 25 – Remoção dos últimos símbolos auxiliares e retorno do cabeçote

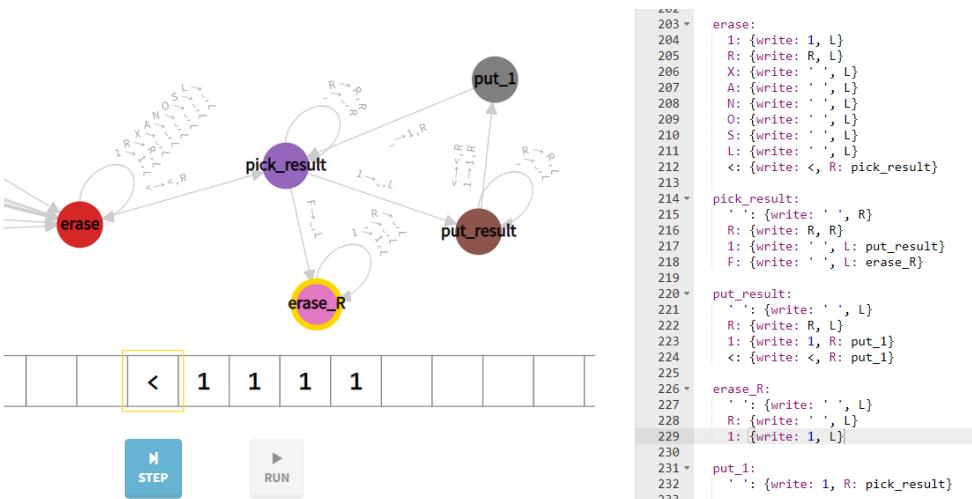


Figura 26 – Finalização da MT1

4.3 Prova da complexidade

Para as demonstrações abaixo, assuma sempre n como o $|R < w >|$, k como $|R < T >|$

4.3.1 Inicialização

Como ele percorre a entrada inteira e coloca o R ao final, são executados n passos e para voltar ao início da fita, n passos são utilizados. no total há 2n passos. Portanto, o módulo de inicialização é **O(n)**, por definição.

4.3.2 Somar hora de partida e duração

O módulo coloca $R < T >$ após o caractere R ao fim da entrada e substitui cada 1 de $R < T >$ por A na entrada. Sabemos que entre o primeiro símbolo da entrada e R (R incluso) há n-1 caracteres e como o módulo vai além do R, há n passos para incluir o primeiro 1. O procedimento descrito é repetido enquanto sobrarem 1's em $R < T >$ e como a cada iteração 1 caractere é "eliminado" (não contará na próxima iteração) e é adicionado um passo (posição da fita após o último 1 incluído na última iteração), sempre são executados n passos até restarem caracteres de $R < T >$. Sendo assim, o módulo executa $k \cdot n$ passos e como sabemos, pela especificação do problema, que nesse caso $k < n$, por definição a complexidade desse módulo é **O(n)**

4.3.3 Divisor por 15 e decisão

Já que $R < T >$ não será mais percorrida, portanto, são realizados $n - k$ durante a divisão, no momento de retorno para $R < G1 >$ são realizados mais $n - k$ passos e na decisão mais 1 passo é realizado. Portanto no total temos $2(n - k) + 1$ passos e como trivialmente $(n - k) < n$, esse módulo é $O(n)$.

4.3.4 Adição

Entre o encontro do primeiro 1 e sua adição depois do R, são realizados $n - k$ passos, no retorno ao ponto de computo só são realizados mais $n - k$ passos e até encontrar o próximo 1 são realizados mais 15 passos, já que há um intervalo de 15 caracteres entre cada par de 1's neste momento da fita. Sendo assim, são realizados $2(n - k) + 15$ passos na primeira execução do módulo. Nas execuções seguintes, teremos a mesma quantidade de passos, porém subtraídos 15 passos da execução anterior, já que os 1's anteriores não terão mais utilidade na execução atual. Portanto, para cada execução i, teremos $2(n - k) + 15 - 15(i - 1)$ passos, sendo $i =$ posição do 1 encontrado após o início de $R < G1 >$. Esta função é limitada superiormente por n. Portanto, este módulo é por definição **O(n)**

4.3.5 Adição

A subtração tem execução análoga à adição, portanto também tem complexidade **O(n)**

4.4 24 horas

Temos 2 opções:

Caso em que não é preciso adicionar ou subtrair 24 horas:

Será feita a verificação em 24 passos, portanto, complexidade **O(1)**

Caso em que é preciso adicionar ou subtrair 24 horas: Esse é o pior caso, já que além da verificação, a máquina irá percorrer 24 passos adicionando/subtraindo na fita, assim serão realizados 48 passos ao todo no pior caso, portanto complexidade **O(1)**

4.4.1 Apagar

Quando o módulo inicia, o programa está no final do resposta após o R, sendo assim ele irá voltar ao R e começara apagar tudo o que tiver antes, até chegar ao início da fita. Como ele irá percorrer w , isso consome n passos e somando os passos necessários para ir do fim da resposta até R, que serão menos do que n passos, já que foi feita a divisão por 15 anteriormente, teremos uma complexidade **O(1)**

4.4.2 Colocar resposta

Este módulo encerra a máquina, irá partir do início até a resposta até R, percorrendo n passos e em seguida, a cada iteração, sempre que encontrar 1 após R, apaga e coloca depois do último digito transferido para a resposta final. Sendo assim, em toda iteração há um passo a menos antes de R e um a mais após R, o que gera n passos para cada dígito da resposta e então temos $l \cdot n$ passos, sendo l o tamanho da resposta. Como sabemos que a resposta é menor que n , $n < l \cdot n < n^2$, portanto este módulo é **O(n)**

4.4.3 Complexidade Final

Temos alguns módulos $O(n)$ e outros $O(1)$, sendo que a soma de suas execuções é acima de n passos e abaixo de n^2 passos, já que não há n execuções necessárias para cada um dos dígitos da representação, portanto no pior caso a complexidade da máquina é $O(n)$

4.5 Exemplos de execução com passo a passo

4.5.1 Exemplo 1

Forma inicial da fita, posição do cabeçote marcado pela cor vermelha: $<1^{19}L1^{15}O1^{75}$

Depois de sair de put_R: $<1^{19}L1^{15}O1^{74}\textcolor{red}{1}R$

Depois de sair de restart: $<\textcolor{red}{1}1^{18}L1^{15}O1^{75}R$

Depois de sair de start_add: $<\textcolor{red}{A}1^{18}L1^{15}O1^{75}R$

Depois de sair de go: $<A1^{18}L1^{15}O1^{75}\textcolor{red}{R}1$

Depois de sair de back: $<A\textcolor{red}{1}1^{17}L1^{15}O1^{75}R1$

Repete "start_add", "go" e "back" até a fita estar como abaixo no estado "start_add":
 $<A^{19}\textcolor{red}{L}1^{15}O1^{75}R1^{19}$

Logo após, depois de sair de start_add: $<A^{19}\textcolor{red}{L}1^{14}O1^{75}R1^{19}$

Depois de sair de Div15: $<A^{19}LX\textcolor{red}{1}1^{13}O1^{75}R1^{19}$

Depois de sair de D1: $<A^{19}LXX\textcolor{red}{1}1^{12}O1^{75}R1^{19}$

Passando por D2, D3, D4 ... D13, que tem funcionamentos similares: $<A^{19}LX^{14}\textcolor{red}{1}O1^{75}R1^{19}$

Depois de sair de D14: <A¹⁹LX¹⁴1O¹75R1¹⁹

Depois de sair de Div15: <A¹⁹LX¹⁴1OX¹174R1¹⁹

Passando por D2, D3, D4 ... D14: <A¹⁹LX¹⁴1OX¹⁴11159R1¹⁹

Repetindo de Div15, D2, D3, D4 ... D14 mais quatro vezes:

<A¹⁹LX¹⁴1OX¹⁴1X¹⁴1X¹⁴1X¹⁴1R¹1¹⁸

Depois de sair de D_back: <A¹⁹L¹X¹⁴1OX¹⁴1X¹⁴1X¹⁴1X¹⁴1X¹⁴1R1¹⁹

Depois de sair de F_choice: <A¹⁹L¹X¹³1OX¹⁴1X¹⁴1X¹⁴1X¹⁴1X¹⁴1R1¹⁹

Depois de sair de F_add: <A¹⁹LX¹⁴A¹O¹X¹⁴1X¹⁴1X¹⁴1X¹⁴1R1¹⁹

Depois de sair de F_add_go: <A¹⁹LX¹⁴AOX¹⁴1X¹⁴1X¹⁴1X¹⁴1X¹⁴1R1¹⁸1¹

Depois de sair de F_add_back: <A¹⁹LX¹⁴A¹O¹X¹⁴1X¹⁴1X¹⁴1X¹⁴1X¹⁴1R1²⁰

Depois de sair de F_add: <A¹⁹LX¹⁴AOX¹⁴A¹X¹³1X¹⁴1X¹⁴1X¹⁴1R1²⁰

Depois de sair de F_add_go: <A¹⁹LX¹⁴AOX¹⁴AX¹⁴1X¹⁴1X¹⁴1X¹⁴1R1¹⁹1¹

Depois de sair de F_add_back: <A¹⁹LX¹⁴AOX¹⁴A¹X¹³1X¹⁴1X¹⁴1X¹⁴1R1²¹

Repete a passagem por “F_add”, “F_add_go”, “F_add_back” mais 4 vezes, resultando na fita abaixo:

<A¹⁹LX¹⁴AOX¹⁴AX¹⁴AX¹⁴AX¹⁴A¹R1²⁵

Depois de sair de F_add: <A¹⁹LX¹⁴AOX¹⁴AX¹⁴AX¹⁴AX¹⁴AX¹⁴AR¹1²⁴

Depois de sair de pre_erase: <A¹⁹LX¹⁴AOX¹⁴AX¹⁴AX¹⁴AX¹⁴AX¹⁴AR1¹1²³

Depois de sair de verify_24hours_1: <A¹⁹LX¹⁴AOX¹⁴AX¹⁴AX¹⁴AX¹⁴AX¹⁴AR11¹1²²

Passando por “verify_24hours_2”, “verify_24hours_3”, ... , “verify_24hours_23”: <A¹⁹LX¹⁴AOX¹⁴AX¹⁴AX¹⁴AX¹⁴AR1²⁵U

Depois de sair de go_to_end_of_erase: <A¹⁹LX¹⁴AOX¹⁴AX¹⁴AX¹⁴AX¹⁴AX¹⁴AR1²⁴1¹

Depois de sair de subtract_24hours_1: <A¹⁹LX¹⁴AOX¹⁴AX¹⁴AX¹⁴AX¹⁴AX¹⁴AR1²³1¹

Depois de sair de Passando por “subtract_24hours_2”, “subtract_24hours_3”, ... , “subtract_24hours_23”:

<A¹⁹LX¹⁴AOX¹⁴AX¹⁴AX¹⁴AX¹⁴AR1¹

Depois de sair de subtract_24hours_24: <A¹⁹LX¹⁴AOX¹⁴AX¹⁴AX¹⁴AX¹⁴AX¹⁴AR1¹f

Depois de sair de erase: <U¹U¹¹⁰R1F

Depois de sair de pick_result, omitindo os espaços em branco: <...R¹UF

Depois de sair de put_result: <U¹...RUF

Depois de sair de put_1: <1U¹...RUF

Depois de sair de pick_result: <1...R¹U

Depois de sair de erase_R e finalizando a MT: <1

4.5.2 Exemplo 2

Forma inicial da fita, posição do cabeçote marcado pela cor vermelha:

$<1^9L1^{30}L1^{90}$

Depois de sair de put_R: $<1^9L1^{30}L1^{89}1R$

Depois de sair de restart: $<1^8L1^{30}L1^{90}R$

Depois de sair de start_add: $<A1^8L1^{30}L1^{90}R$

Depois de sair de go:

$<A1^8L1^{30}L1^{90}R1$

Depois de sair de back:

$<A11^7L1^{30}L1^{90}R1$

Repete “start_add”, “go” e “back” até a fita estar como abaixo no estado “start_add”:
 $<A^9L1^{30}L1^{90}R1^9$

Logo após, depois de sair de start_add: $<A^9L11^{29}L1^{90}R1^9$

Depois de sair de Div15: $<A^9LX11^{28}L1^{90}R1^9$

Depois de sair de D1: $<A^9LXX11^{27}L1^{90}R1^9$

Passando por D2, D3, D4 ... D13, que tem funcionamentos similares:

$<A^9LX^{14}11^{15}L1^{90}R1^9$

Depois de sair de D14: $<A^9LX^{14}111^{14}L1^{90}R1^9$

Depois de sair de Div15: $<A^9LX^{14}1X11^{13}L1^{90}R1^9$

Passando por D2, D3, D4 ... D14: $<A^9LX^{14}1X^{14}1L1^{90}R1^9$

Repetindo de Div15, D2, D3, D4 ... D14 mais seis vezes:

$<A^9LX^{14}1X^{14}1LX^{14}1X^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^8$

Depois de sair de D_back: $<A^9LX^{14}1X^{14}1LX^{14}1X^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^9$

Depois de sair de F_choice: $<A^9LXX^{13}1X^{14}1LX^{14}1X^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^9$

Depois de sair de F_add: $<A^9LX^{14}AX^{14}1X^{14}1LX^{14}1X^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^9$

Depois de sair de F_add_go: $<A^9LX^{14}AX^{14}1LX^{14}1X^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^8$

Depois de sair de F_add_back: $<A^9LX^{14}AX^{14}1X^{13}1LX^{14}1X^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^{10}$

Depois de sair de F_add: $<A^9LX^{14}AX^{14}AX^{14}1X^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^{10}$

Depois de sair de F_add_go: $<A^9LX^{14}AX^{14}ALX^{14}1X^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^9$

Depois de sair de F_add_back: $<A^9LX^{14}AX^{14}ALX^{14}1X^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^{11}$

Depois de sair de F_add: $<A^9LX^{14}AX^{14}ALX^{14}1X^{13}1X^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^{11}$

Depois de sair de F_sub: $<A^9LX^{14}AX^{14}ALX^{14}SX^{13}1X^{14}1X^{14}1X^{14}1X^{14}1R1^{11}$

Depois de sair de F_sub_go: $<A^9LX^{14}AX^{14}ALX^{14}SX^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^{10}$

Depois de sair de F_sub_done: $<A^9LX^{14}AX^{14}ALX^{14}SX^{14}1X^{14}1X^{14}1X^{14}1X^{14}1R1^9$

Depois de sair de F_sub_back: $<A^9LX^{14}AX^{14}ALX^{14}SX^{13}1X^{14}1X^{14}1X^{14}1X^{14}1R1^{10}$

Passando por F_sub, F_sub_go, F_sub_done, F_sub_back mais 5 vezes:
 $<A^9LX^{14}AX^{14}ALX^{14}SX^{14}SX^{14}SX^{14}SX^{14}S1R1^5$

Depois de sair de F_sub: < A⁹LX¹⁴AX¹⁴ALX¹⁴SX¹⁴SX¹⁴SX¹⁴SX¹⁴SR1¹
 Depois de sair de pre_erase: < A⁹LX¹⁴AX¹⁴ALX¹⁴SX¹⁴SX¹⁴SX¹⁴SX¹⁴SR1¹¹
 Depois de sair de verify_24hours_1: < A⁹LX¹⁴AX¹⁴ALX¹⁴SX¹⁴SX¹⁴SX¹⁴SX¹⁴SR1¹
 Passando por “verify_24hours_2”, “verify_24hours_3” e “verify_24hours_4”: < A⁹LX¹⁴AX¹⁴ALX¹⁴SX¹⁴SX¹⁴SX¹⁴SX¹⁴SR1⁵
 Depois de sair de verify_24hours_5: < A⁹LX¹⁴AX¹⁴ALX¹⁴SX¹⁴SX¹⁴SX¹⁴SX¹⁴SR1⁴
 Depois de sair de erase: < U U¹³⁰R11111F
 Depois de sair de pick_result: < U¹³¹ R U1111F
 Depois de sair de put_result: < U U¹³⁰RU1111F
 Depois de sair de put_1: < 1 U U¹²⁹RU1111F
 Depois de sair de pick_result: < 1 U¹³⁰ R U U111F
 Depois de sair de put_result: < 1 U U¹²⁹RUU111F
 Depois de sair de put_1: < 11 U U¹²⁸RUU111F
 Passando por “pick_result”, “put_result” e “put_1” mais 3 vezes: < 11111 U U¹²⁵RUUUUUUF
 Depois de sair de pick_result: < 11111 U¹²⁶RUUUU U
 Depois de sair de erase_R e finalizando a MT: < 11111

5 Solução para MT2

5.1 Definição formal

A MT2 que estamos desenvolvendo representa uma variação da Máquina de Turing Padrão. Portanto, podemos aproveitar a definição geral estabelecida na MT1. No entanto, a distinção crucial ocorre devido à incorporação de múltiplas fitas em uma única máquina. Assim, sua definição formal é expressa através de uma óctupla:

$$MT2 = (E, \Sigma, \Gamma, <, v, \delta, i, F)$$

Onde:

- δ é uma função de $E \times \Gamma^k$ para $E \times (\Gamma^k \times \{D, E, I\})^k$.
- Uma configuração instantânea tem a forma:

$$[e, x_1 a_1 y_1, \dots, x_k a_k y_k]$$

- A linguagem aceita e o conjunto de toda palavra $w \subseteq \Sigma^*$ tal que

$$[i, < w, < v, \dots, < u] \vdash^* [e, x_1 a_1 y_1, x_2 a_2 y_2, \dots, x_k a_k y_k]$$

onde $e \in F$ e $\delta(e, a_1, a_2, \dots, a_k)$ é indefinido.

(VIEIRA; BARBOSA, 2010)

Especificamente em nossa solução temos:

- E é o conjunto finito de estados, identificados por números, contendo apenas 1 estado final $F = \{11\}$ e 1 estado inicial $i = \{0\}$.
- $\Sigma \subseteq \Gamma$ é o alfabeto da palavra w de entrada, assim $\Sigma = \{1, L, O\}$
- Γ^k é o alfabeto das k fitas que serão definidas no próximo tópico.

5.2 Definição das múltiplas fitas

Para nossa solução necessitaremos de 5 fitas, definidas de tal forma:

1. Tem como entrada o **tempo de duração do voo** e **horário do local de partida** concatenados. Além disso é apenas nessa fita onde faremos modificações em seu conteúdo, por conseguinte, ela conterá o resultado e as demais serão apenas leitura. O alfabeto da **Fita 1** é definido como: $\Gamma^1 = \{<, v, 1, N, L, O\}$
2. Tem como entrada o **fuso do local de partida** (origem). Seu alfabeto é: $\Gamma^2 = \{<, v, 1, L, O\}$
3. Sua entrada é o **fuso do local de chegada** (destino) e seu alfabeto é igual ao da **Fita 2** e portanto: $\Gamma^3 = \Gamma^2$
4. Essa fita irá servir para realizar a **divisão por 15** nos nossos cálculos em busca do horário de chegada, desta forma ela é preenchida, de maneira que tende ao infinito, onde a cada 14 espaços vazios v há um espaço preenchido com o caractere 1. Seu alfabeto se dá de tal maneira: $\Gamma^4 = \{<, v, 1\}$
5. Por fim essa fita nos será útil em **ajustar os horários dentro do intervalo de 00 à 23** horas e seu alfabeto é equivalente ao da **Fita 4**, por conseguinte: $\Gamma^5 = \Gamma^4$

5.3 Implementação

O software utilizado para implementar a **MT1** foi descartado na **MT2** pois o mesmo não possui suporte para máquinas de múltiplas fitas, nos levando a utilizar a plataforma de diagramas *excalidraw*.

A figura disponível nos anexos⁹ mostra o diagrama completo da implementação da MT2. Visando tornar mais didático buscaremos explicar de forma modular o seu funcionamento.

1. Fluxo onde a longitude da cidade de origem é à Oeste:

Para o tratamento desse fluxo, visível na Figura 27, utilizamos apenas os estados: 0, 1, 7 e 3 onde:

- **Estado 0:** É o estado inicial e fará com que a **Fita 1** seja percorrida até o final e os demais cabeçotes permaneçam imóveis. Ao chegar até ele o seu cabeçote deve retornar a esquerda, para se manter no último caractere preenchido na fita, e caso o caractere inicial da **Fita 2** seja O transicione para o **Estado 1**.

⁹ Disponível no Anexo II, pg 42.

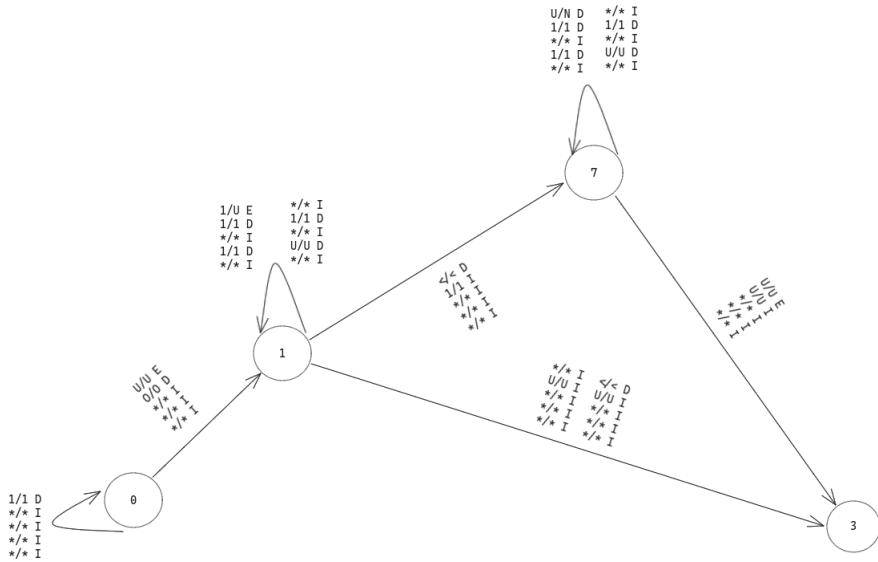


Figura 27 – Diagrama Parcial: Longitude da cidade de origem à Oeste

- **Estado 1:** Como nossas **Fita 1**, **Fita 2** e **Fita 4** vêm preenchidas conforme definição, daremos inicio a subtração: a cada 15 caracteres 1's percorridos na **Fita 2** substituíssse um caractere 1 na **Fita 1** por um caractere vazio v , caso não haja mais caracteres na **Fita 2** independentemente da posição dos outros cabeçotes transacionaremos para o **Estado 3** e manteremos os cabeçotes imóveis, com exceção do caso de a primeira fita estar exatamente em seu início (caractere $<$), neste precisaremos posicionar seu cabeçote a direita.
A outra possibilidade de transição desse estado diz respeito a especificidade que temos na subtração, ao subtrairmos um número de outro podemos nos deparar com a situação onde minuendo¹⁰ pode ser menor que o subtraendo¹¹ gerando assim números negativos, de modo que, na ocorrência dessa situação: alcançasse o inicio da **Fita 1** e a **Fita 2** ainda tem caracteres preenchidos, fazendo com que precisemos transicionar para o **Estado 7** colocando o cabeçote da **Fita 1** na posição a direita e mantendo os demais imóveis.
- **Estado 3:** Na nossa MT2 como dito anteriormente só há um estado final e ele é o **Estado 11** entretanto como nessa seção estamos tratando de forma modular consideraremos **Estado 3** como um estado final desse fluxo.
- **Estado 7:** Aqui introduziremos o conceito adotado para tratarmos com os números negativos: faremos uso do caractere N como um sinônimo do número -1 , no unário a quantidade de 1's representa o valor em decimal, nesse cenário a quantidade de N 's representa o valor decimal de um número negativo. Exemplo: $|N|N|N| = -3$. Dito isto, a cada quinze caracteres 1's percorridos à direita na **Fita dois**¹² substituiremos o caractere v por N . Ao alcançarmos o fim da **Fita 2** transacionaremos para o **Estado 3**.

¹⁰ É o número do qual outro número (o subtraendo) é subtraído. Em uma expressão de subtração $a - b$, a é o minuendo.

¹¹ É o número que é subtraído do minuendo. Em uma expressão de subtração $a - b$, b é o subtraendo.

¹² Sendo isso controlado de forma que encontramos um caractere 1 na **Fita 4**.

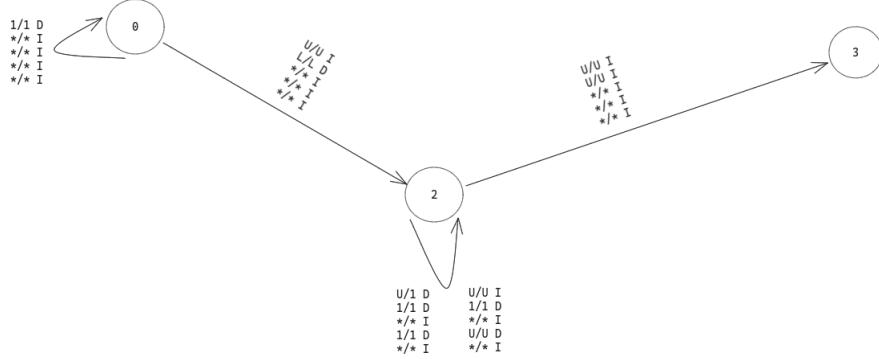


Figura 28 – Diagrama Parcial: Longitude da cidade de origem à Leste

Ao concluir esse fluxo a **Fita 1** que é nossa fita de resposta estará preenchida com o resultado da seguinte equação¹³: $\{(T + P) - Fp\}$.

2. Fluxo onde a longitude da cidade de origem é à Leste:

Para solucionar esse cenário utilizamos apenas três estados: **Estado 0**, **Estado 1** e **Estado 3**, além disso por se tratar de uma soma não há possibilidade de gerar números negativos, tornando-o consideravelmente mais simples. Na Figura 28 pode-se ver o diagrama referente a esse fluxo, e a seguir a descrição do seu funcionamento a partir dos seus estados.

- **Estado 0:** nesse fluxo esse estado se comportará igualmente como no fluxo anterior (n 1) percorrendo toda a **Fita 1** sem movimentar os demais cabeçotes, entretanto adicionamos aqui a transição para o **Estado 2** uma vez que o primeiro espaço da **Fita 2** esteja preenchido com o caractere *L* que nos indica que é necessário executar a soma.
- **Estado 2:** nesse estado realizaremos a operação de soma, logo, a cada quinze caracteres 1 contidos na **Fita 4** adicionasse um caractere 1 na **Fita 1**. Ao chegar no fim da **Fita 2** transacionaremos para o **Estado 3**.
- **Estado 3:** assim como no fluxo anterior esse estado também será considerado um estado final, sem que esqueçamos que isso só é válido por estarmos fazendo a análise modular.

Ao finalizarmos esse fluxo teremos na **Fita 1** o resultado da operação: $\{T + P + Fp\}$.

3. Fluxo onde a longitude da cidade de destino é à Oeste:

Esse fluxo dá continuação a qualquer um dos fluxos anteriores, portanto nossa **Fita 1** está preenchida com o seu respectivo resultado. Seguindo nossa equação principal¹² ao nos depararmos com a longitude da cidade de destino à Oeste deve realizar-se uma operação de soma. Na **Figura 30** vê-se o diagrama do fluxo e daremos continuidade a descrição do seu funcionamento a partir dos seus estados.

- **Estado 3:** aqui consideraremos esse estado como nosso estado inicial e ele possui duas transições possíveis: nossa **Fita 1** está preenchida com valores positivos ou negativos? Diferenciamos isso checando se o seu cabeçote está posicionado em

¹³ Instâncias definidas em 4.1

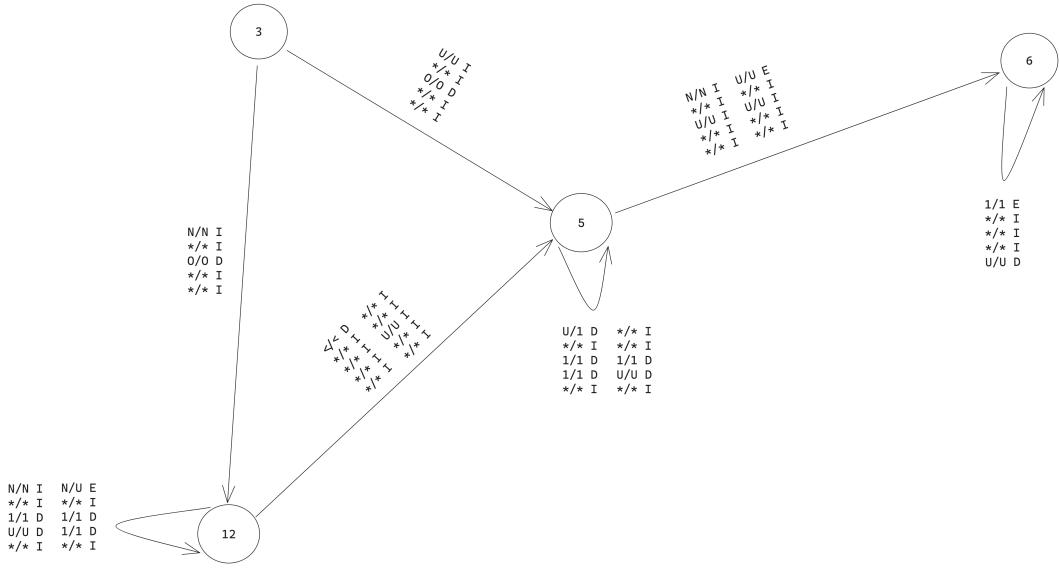


Figura 29 – Diagrama Parcial: Longitude da cidade de destino à Oeste

uma célula preenchida por v ou N , caso esteja posicionada em v transiciona diretamente para o **Estado 5**, no outro caso possível transiciona para o **Estado 12**. Ambas transições movem o cabeçote da **Fita 3** para direita.

- **Estado 12:** esse será o responsável por realizar a soma de um valor negativo com um valor positivo, desta maneira manteremos assim como nos fluxos anteriores nossas modificações na **Fita 1** apenas a cada quinze caracteres 1's contidos na fita de leitura, nesse caso **Fita 3**, logo substituí-se um caractere N por um v e move-se o cabeçote a esquerda. Caso chegue no caractere delimitador $<$ da **Fita 1** ou no caractere final da **Fita 3** transiciona-se para o **Estado 5**. No caso do caractere $<$ move-se o cabeçote da sua fita à direita.
- **Estado 5:** esse estado recebe o resultado da parte da soma que foi realizada com negativos assim como recebe diretamente a fita positiva a se realizar a soma, como foi descrito no **Estado 3**, caso o cabeçote da **Fita 1** esteja posicionado num caractere v indicando que a fita já está de forma positiva realizaremos a operação de soma da mesma maneira que realizamos no fluxo anterior no **Estado 2**: a cada quinze 1's na fita de leitura, substitui-se o caractere v da **Fita 1** por 1 e ao chegar no fim dela move-se o cabeçote a esquerda e transiciona-se para o **Estado 6**. Entretanto, caso chegue-se a esse estado com a fita ainda negativa¹⁴ transiciona-se diretamente para o **Estado 6**.
- **Estado 6:** para esse fluxo consideraremos o **Estado 6** como estado final, analogamente fizemos nos fluxos anteriores.

Ao encerrarmos esse fluxo temos agora o resultado desta equação: $\{T + P \pm Fp + Fc\}$.

4. Fluxo onde a longitude da cidade de destino é à Leste:

Esse fluxo tem a mesma função do fluxo anterior, *Fluxo onde a longitude da cidade de destino é à Oeste*, com a diferença que como a longitude da cidade de destino é à

¹⁴ Isso acontece caso o valor contido na **Fita 1** fosse menor que o valor a ser somado. Exemplo: **Fita 1**: $| < |N|N|$, **Fita 3**: $| < |1|$.

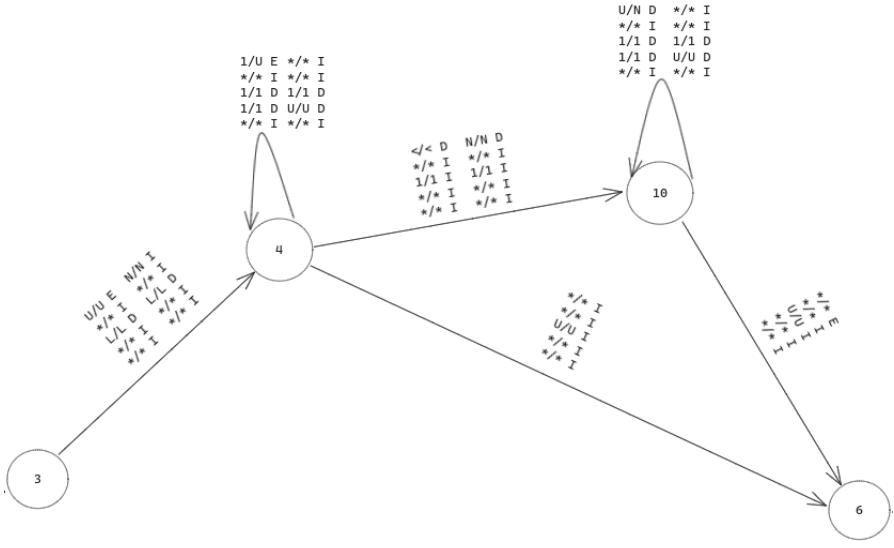


Figura 30 – Diagrama Parcial: Longitude da cidade de destino à Leste.

Leste precisamos realizar a operação de subtração. Tanto o diagrama (Figura) como a explicação dos seus estados está a seguir:

- **Estado 3:** assim como no fluxo anterior esse será considerado nosso estado inicial, contudo, ao reconhecer o caractere L na **Fita 3** possui apenas uma transição que vai para o **Estado 4** mudando apenas a forma como o cabeçote da **Fita 1** se comporta, caso esteja posicionado em um v move-se à esquerda e caso seja N se mantém imóvel.
- **Estado 4:** esse estado fará a subtração, no caso da **Fita 1** estar preenchida com caracteres negativos transiciona-se diretamente para o **Estado 10** movendo seu cabeçote à direita¹⁵. No caso de a **Fita 1** estar positiva¹⁶ faremos a subtração de forma similar aos outros fluxos¹⁷. Após percorremos o cabeçote à esquerda e alcançarmos o seu caractere inicial mas ainda terem caracteres na **Fita 3** transaciona-se para o **Estado 10** movendo seu cabeçote à direita¹⁸. Caso percorramos a **Fita 3** até o seu final, independente da posição do cabeçote da **Fita 1**, transiocionaremos ao **Estado 6** sem mover nenhum cabeçote.
- **Estado 10:** esse será o estado responsável por realizar a subtração com números negativos, funcionando de forma equivalente ao **Estado 7** no nosso primeiro fluxo *Fluxo onde a longitude da cidade de origem é à Oeste*¹⁹. todavia percorremos neste fluxo a **Fita 3** no luga da **Fita 2** e transicionamos para o **Estado 6** ao invés do **Estado 3** ao terminarmos de percorrer a nossa fita de leitura.
- **Estado 6:** para esse fluxo assim como no anterior consideraremos esse estado como estado final.

Chegando ao fim desse fluxo temos o resultado dessa equação: $\{T + P \pm Fp - Fc\}$.

¹⁵ Refere-se ao cabeçote da **Fita 1**.

¹⁶ Seu cabeçote estar posicionado no caractere 1

¹⁷ Ao percorrer à direita quinze 1's presentes na **Fita 3** substitui-se um caractere v por 1 na **Fita 1**. e move seu cabeçote à esquerda.

¹⁸ Refere-se ao cabeçote da **Fita 1**.

¹⁹ Disponível no item 1 da seção 5.3, pg. 28.

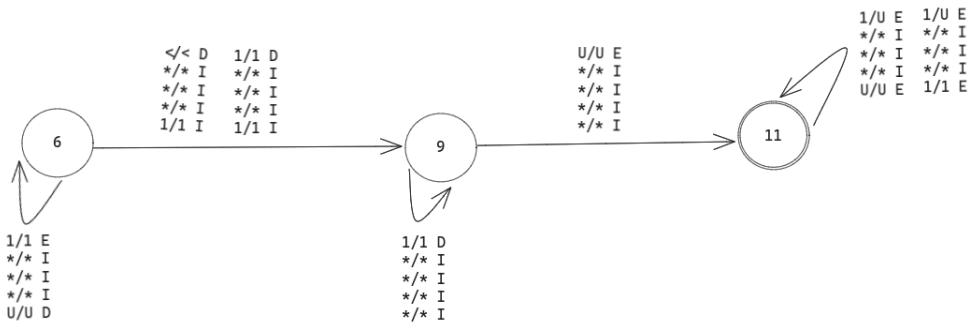


Figura 31 – Diagrama Parcial: Correção de valores maiores que 23.

5. Fluxo de verificação e correção para o formato 24h:

Ao olhar para nossa equação de solução pode-se pensar que nosso trabalho chegou ao fim, todavia, ao lidarmos com horas é necessário que as mesmas estejam no formato adequado²⁰, por isso não podemos ter nossa fita resultado preenchidos com mais de vinte e três 1's, por isso foi criado esse fluxo que irá verificar a fita resultado (**Fita 1**) e corrigi-la caso necessário. Vê-se seu diagrama (Figura 31) e descrição na sequência.

- **Estado 6:** o estado final dos fluxos que finalizam os cálculos na fita de resposta é justamente o nosso estado inicial nesse fluxo de verificação e correção e sua funcionalidade é verificar se há mais de vinte e três células preenchidas na nossa fita de resposta. Inicialmente percorre-se a **Fita 1** à esquerda enquanto há v 's na **Fita 5**, uma vez que alcancemos a célula vinte e quatro, marcada com o caractere 1, na fita auxiliar²¹ tanto caso alcancemos o inicio ou não da **Fita 1** deve-se transicionar para o **Estado 9** movendo seu cabeçote à direita e mantendo os demais imóveis.
- **Estado 9:** a função desse estado é apenas posicionar corretamente o cabeçote da **Fita 1** para a realização da correção, logo ele percorre toda fita sem alterá-la e por fim transiciona à esquerda ao achar seu caractere v .
- Estado 11: esse é o estado final e como podem ver ele é de fato o estado final da nossa MT2 também, ele é responsável pela correção e funciona de modo que ao se deparar com o 1 da **Fita 5** que indica que há mais de 23 dígitos, ele passa a percorrer a esquerda removendo vinte e quatro 1's da nossa fita resposta, ao substituir 1 por v fazendo com que dessa forma a resposta esteja correta em horas. Como não há transição definida para o delimitador < e a fita percorre à esquerda a máquina "pára"²².

Ao alcançar o **Estado 11** é finalizado o trabalho da máquina, tendo em sua fita de resposta o horário de chegada do voo de acordo com nossa equação base. Embora esse seja um caminho da finalização também há mais duas possibilidades a serem discutidas nos próximos fluxos.

6. Fluxo de verificação e correção de resultado negativo:

²⁰ O formato de horas adotado foi o de 24h, sendo assim indo de 00h à 23h.

²¹ Refere-se a **Fita 5**.

²² Pára com acento apenas para dar ênfase que se trata do uso do verbo parar, com a licença da nossa língua portuguesa.

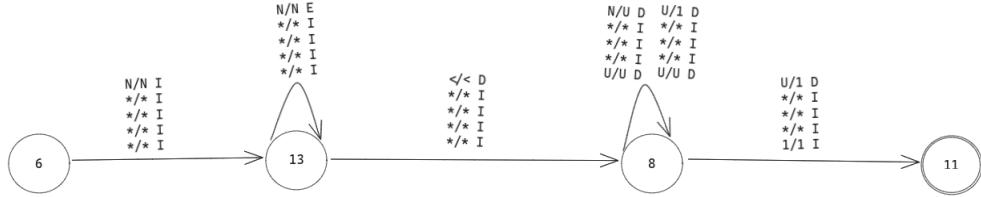


Figura 32 – Diagrama Parcial: Correção de valores negativos.

Como é sabido, não existem horas negativas e além disso definimos que a resposta do nosso problema seria dada em unário, então é duplamente equivocado a fita de resposta estar preenchida por N 's. De tal maneira que foi criado esse fluxo que verifique se estamos nesse caso e se sim ele seja corrigido. Como os demais fluxos vamos descrever o funcionamento de seus estados e seu diagrama está presente na Figura 32.

- **Estado 6:** assim como no fluxo anterior esse estado será nosso estado inicial, contudo, ao verificar a existência do caractere N na primeira fita transiciona-se para o **Estado 13** sem movimentar nenhum cabeçote. Essa é sua única transação.
- **Estado 13:** esse estado é um estado auxiliar e percorrerá toda a **Fita 1** à esquerda até que encontre seu caractere $<$ e transicione para o **Estado 8** movendo seu cabeçote à direita.
- **Estado 8:** nesse estado eliminaremos os caracteres N 's cada hora negativa deve-se somar ao valor 24 o valor negativo de horas para se obter a hora do dia correta. Para realizarmos essa operação percorreremos ambas fitas à direita fazendo com que enquanto haja caracteres v na **Fita 5** os valores negativos sejam apagados e os valores v 's sejam substituídos por 1, ao chegar no marcador de 24 caracteres da **Fita 5** faremos nossa última substituição na fita de resposta e moveremos seu cabeçote à direita ao transicionar para o **Estado 11**.
- **Estado 11:** esse será nosso estado final e apenas isso.

Vale notar que ao executar esse fluxo teremos o horário do da chegada do voo como um dia anterior ao da sua partida, devido a diferença de fuso, além disso apenas nesse fluxo nossa fita de resposta terá caracteres v 's antes do valor das horas em unário.

7. Fluxo onde a fita de resposta contém menos de 23h e não há valores negativos

Esse é o melhor caso, não é necessário nenhuma correção sendo apenas um fluxo de verificação, logo só é seguido se não atende os parâmetros dos dois últimos fluxos descritos: Seu diagrama está presente na Figura 33 e sua breve descrição logo abaixo.

- **Estado 6:** não diferente dos dois anteriores esse é o nosso estado inicial, enquanto não encontrarmos o caractere 1 na **Fita 5** indicando que há 24 dígitos percorreremos sem alterações a **Fita 1**. Caso alcancemos o início da fita de resposta e ainda não ter indicação de 24 caracteres dada através da **Fita 5** transiciona-se para o **Estado 11** sem mover nenhum cabeçote.

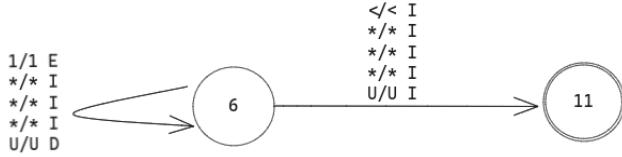


Figura 33 – Diagrama Parcial: Fita de horas positivas e menores que 23.

Esse é o nosso menor fluxo, tendo apenas um estado além do final e é o último a ser descrito neste trabalho acadêmico. Essa e as demais descrições de fluxos serão utilizadas para desenvolvemos nossos exemplos de uso da nossa MT2.

5.4 Prova da complexidade

Para a análise de complexidade da **MT2**, não será analisado o fluxo 7, pois estamos em busca do pior caso e o fluxo citado não o integra e portanto é irrelevante no cálculo. Também omitiremos o caso onde o fluxo e o 3 são executados na mesma computação da máquina, pois sua complexidade é análoga à da computação com os fluxos analisada abaixo.

Considere que (e_1, e_2, e_3, e_4) sejam as respectivas entradas para as fitas de 1 a 4.

Considere ainda que quando for citado o tamanho da entrada, a não ser quando seja explicitamente dito que seja a entrada de uma fita específica, a entrada referida será a soma das entradas de cada fita (i.e igual à entrada da **MT1**) e que $n = \text{tamanho da entrada}$.

5.4.1 Fluxo 1

Para este fluxo, observe que o pior caso é quando houver resultante da computação negativo após a execução do fluxo.

Inicialmente $|e_1|$ passos são realizados. Como $|e_1| = |n| - |e_2| - |e_3| - |e_4|$, temos complexidade inicial $= |n| - |e_2| - |e_3| - |e_4|$, logo, **O(n)**, já que $n < n - k < n^2$

A seguir, no pior caso são realizados $|e_2|$, no estado 1, sendo $|e_2| = n - |e_1| - |e_3| - |e_4|$ com o adendo que será realizado um passo a mais, para levar e_1 de ' $<$ ' para a primeira posição da entrada, pois é o caso em que $|e_2|/15 > |e_1|$. Como a apenas 1 passo a mais do que o do estado anterior, é **O(n)**

Como estamos no pior caso, partimos para o passo 7, onde serão percorridos o restante dos caracteres na fita 2.

Esse fluxo termina na transição para estado 3, portanto, no pior caso teremos $|e_1| + |e_2| - 2(|e_3| + |e_4|) + 1$ passos, gerando complexidade **O(n)**

5.4.2 Fluxo 4

No pior caso, a cidade de origem estará a oeste e portanto iremos para o estado, onde serão executados $|e_3|$ passos e como $|e_3| = |n| - |e_1| - |e_2| - |e_4|$, temos complexidade **O(n)** neste caso.

Em resumo, este trabalho contribuiu para o conhecimento de aplicações de máquinas de Turing em problemas práticos e teóricos, proporcionando soluções para a desafiadora tarefa de calcular a hora de chegada em viagens com múltiplos fusos horários.

Referências

BRANDAO, P. R. Alan turing: da necessidade do cálculo, a máquina de turing até à computação. In: . n 12. Revista de Ciências da Computação, 2017. Disponível em: <<https://repositorioaberto.uab.pt/bitstream/10400.2/7683/1/8-Texto%20Artigo-14-1-10-20180311.pdf>>. Acesso em: 22 nov. 2023. Citado na página 1.

GIBRA', E. *A geografia namorada da matemática estava enamorada do design, que quis namorar a educação: o projeto Local - Greenwich Mean Time.* [S.l.]: Sinais, vol 1, n 3, 2008. Citado na página 3.

REDAÇÃO, NATIONAL GEOGRAPHIC BRASIL. *Quem foi Alan Turing, pioneiro no desenvolvimento da inteligência artificial e da computação moderna.* 2023. [Online]. Disponível em: <<https://www.nationalgeographicbrasil.com/ciencia/2023/06/quem-foi-alan-turing-pioneiro-no-desenvolvimento-da-inteligencia-artificial-e-da-computacao-moderna>>. Acesso em: 22 nov. 2023. Nenhuma citação no texto.

VIEIRA, N. J.; BARBOSA, I. G. *Introdução aos fundamentos da computação, Capítulo 4: Maquinas de Turing.* [S.l.]: UFMG, 2010. Citado 2 vezes nas páginas 3 e 27.

WIKIPEDIA. *Kurt Gödel.* 2004. [Online]. Disponível em: <https://pt.wikipedia.org/wiki/Kurt_G%C3%B3del>. Acesso em: 22 nov. 2023. Nenhuma citação no texto.

WIKIPEDIA. *Máquina de Turing.* 2005. [Online]. Disponível em: <https://pt.wikipedia.org/wiki/M%C3%A1quina_de_Turing>. Acesso em: 22 nov. 2023. Nenhuma citação no texto.

WIKIPEDIA. *Bertrand Russel.* 2006. [Online]. Disponível em: <https://pt.wikipedia.org/wiki/Bertrand_Russell>. Acesso em: 22 nov. 2023. Nenhuma citação no texto.

WIKIPEDIA. *David Hilbert.* 2007. [Online]. Disponível em: <https://pt.wikipedia.org/wiki/David_Hilbert>. Acesso em: 22 nov. 2023. Nenhuma citação no texto.

WIKIPEDIA. *Paul Bernays.* 2011. [Online]. Disponível em: <https://es.wikipedia.org/wiki/Paul_Bernays>. Acesso em: 22 nov. 2023. Nenhuma citação no texto.

WIKIPEDIA. *Leopold Kronecker.* 2012. [Online]. Disponível em: <https://en.wikipedia.org/wiki/Leopold_Kronecker>. Acesso em: 22 nov. 2023. Nenhuma citação no texto.

ANEXO I: Descrição do Problema

Projeto Final - Viagem ao Mundo

Proposto pelo professor Felipe Fernandes na disciplina Teoria da Computação da Universidade Federal da Bahia

A companhia aérea Turing Airlines precisa de um sistema computacional, implementado com Máquinas de Turing, para gerenciar os horários de partida e chegada de seus voos com base nos fusos horários do planeta. Os fusos horários são definidos levando em consideração a distância longitudinal, em graus, relativa ao meridiano de Greenwich. Considere que o globo terrestre possui 24 fusos horários, igualmente espaçados, a cada 15 graus de longitude. A cada longitude a leste de Greenwich soma-se uma hora; a cada longitude a oeste subtrai-se uma hora. Por exemplo, Pequim está no GMT+8, isto é, 120° Leste de Greenwich e, portanto, com 8 horas a mais que esse meridiano. Nova York está no GMT-4, isto é, 60° Oeste de Greenwich e, portanto, com 4 horas a menos que este meridiano.

Projete uma máquina de Turing que recebe a longitude em graus da cidade de origem, a longitude em graus da cidade de destino, o horário em que o voo partiu (referente à cidade de origem), e a duração do voo em horas. O sistema deve calcular o horário em que o voo chegou na cidade de destino. Assuma que as longitudes são múltiplas de 15 e as horas são fechadas (sem considerar minutos).

A tarefa consiste em projetar duas Máquinas de Turing para o problema dado, respeitando os seguintes requisitos:

1. MT1: Máquina de Turing padrão (apenas uma fita e determinística).
2. MT2: Variação de Máquina de Turing padrão.

Para MT1 e MT2, é necessário responder à seguinte pergunta: quantos passos a máquina executa no pior caso em função do tamanho da entrada? Calcule formalmente, com clareza de detalhes, e expresse em termos de notação assintótica.

ANEXO II: DIAGRAMA COMPLETO MT2

Acesso online disponível em: <https://excalidraw.com/room=cc02>

