



Instituto Tecnológico de Aeronáutica

Divisão de Ciência da Computação

PROJETO:
SUBTRATOR DE NÚMEROS INTEIROS

Disciplina:

Fundamentos de Autômato e Linguagens Formais - CT200

Professor:

Dr. Carlos Henrique Costa Ribeiro

Equipe de alunos:

Bernardo de Pádua dos Santos

Hilário dos Santos Neto

Tereza Lutércia Saraiva Lins

Maurício Pozzobon Martins

São José dos Campos, 24 de junho de 2008.

1. Introdução

1.1. *Motivação*

O estudo da computabilidade foi motivado por duas questões fundamentais: “o que é um algoritmo?” e “quais são as capacidades e limitações da computação algorítmica?” Uma resposta à primeira questão exige um modelo formal de computação. Aparentemente, combinar um computador e uma linguagem de programação de alto nível, formando um sistema computacional, resultaria no suporte ideal para o estudo da computabilidade. Mas qual computador? Que quantidade de memória ele deveria ter? Qual linguagem de programação? A simples seleção de uma máquina e linguagem qualquer, por melhores que estas sejam, implicaria em conseqüências inesperadas e indesejáveis para a segunda questão. A resposta sobre se um problema é algorítmicamente solucionável deveria ser independente do modelo computacional usado [1].

A caracterização geral da computação algorítmica tem sido bastante estudada por matemáticos e lógicos desde os anos 30. Máquinas abstratas desenvolvidas por Alan Turing fazem parte desses modelos de computação.

A máquina de Turing tem muitos atributos em comum com um computador. Ela processa entradas, escreve na memória e produz dados. Embora as instruções da máquina de Turing sejam primitivas em relação àsquelas processadas em um computador, não é difícil perceber que o processamento de um computador pode ser simulado por uma seqüência apropriada de máquinas de Turing. Por causa da simplicidade e similaridade com os modernos computadores, as máquinas de Turing são muito utilizadas no estudo da computação [1].

1.2. *Especificação do problema e objetivo*

O problema consiste em modelar uma máquina de Turing que subtraia dois números inteiros. Portanto, o objetivo deste projeto é implementar uma máquina de Turing para subtrair dois números inteiros x e y , de maneira que a entrada do sistema seja: x e y inteiros, separados por um símbolo específico, e a saída seja $x - y$.

A ferramenta JFLAP (<http://www.jflap.org/>) foi utilizada para a implementação.

1.3. *Escopo*

A máquina de Turing que realiza subtração de dois inteiros é um projeto acadêmico e tem a finalidade de demonstrar a aplicação dos conhecimentos adquiridos na disciplina Fundamentos de Autômato e Linguagens Formais.

Variações nos dados de entrada são tratadas pela maneira mais simples possível, ou seja, a máquina trava ou termina a leitura da entrada em um estado não final.

Para representar os números inteiros, escolhemos a representação unária. Embora não seja a muito eficiente do ponto de vista computacional e de armazenamento, ela simplifica consideravelmente o desenvolvimento da Máquina de Turing. Para representar inteiros positivos, utiliza-se o algarismo "1". O número de "1"s concatenados representa a magnitude do número. Por exemplo, o número 6 é representado pela cadeia "111111". O número 0 é representado pela cadeia vazia "". Números negativos são representados de forma análoga aos números positivos, mas utilizando-se o algarismo "0". Por exemplo, o número -4 é representado pela cadeia "0000".

Tabela 1: Exemplos de valores na representação unária utilizada

Representação	Valor (base 10)
111	3
11	2
1	1
	0
0	-1
00	-2
000	-3

Esta representação, embora simples, possibilita a representação de inteiros de qualquer ordem de grandeza, pois a fita da máquina de Turing é infinita. Entretanto, o número de transições necessárias à computação cresce linearmente com o tamanho dos números representados nas entradas.

1.4. Especificação de Requisitos

A máquina de Turing foi implementada seguindo um conjunto de requisitos arquitetados pelo grupo de desenvolvimento com o intuito de apresentar robustez e coerência. Tais requisitos são apresentados a seguir.

1) Aceitar como entrada dois números inteiros na forma unária especificada no item “Escopo”, separados pelo sinal de menos (“-”). Por exemplo, “11111-0000”.

2) Ter a cabeça de leitura posicionada no início do primeiro número antes do cálculo.

3) Ter a cabeça de leitura posicionada no início do resultado após o cálculo.

4) Escrever o resultado da operação na fita, na forma unária especificada anteriormente, sobrescrevendo as entradas. Por exemplo, tendo-se “111-11” inicialmente na fita teremos “1” na fita após o cálculo.

5) O número 0 será caracterizado pela ausência de 1’s e 0’s.

6) Se um ou ambos os termos da subtração contiverem mistura de 0’s e 1’s a cadeia de entrada será rejeitada.

7) Cadeias incompletas, como por exemplo -000, são aceitas, seriam equivalentes a (0-(-3)); 111- também é aceito, seria equivalente a (3-0). Cadeias que não apresentam o sinal de subtração não são aceitas. Cadeia com apenas o sinal de subtração também deve ser aceita, equivalente a (0-0).

Tabela 2: Exemplos de entradas e saídas esperadas para o subtrator

Entradas	Saídas
1111-11	11
11-000	11111
-111	000
000-	000
-	
00-1111	000000
000-00	0

1.5. Ordem de Apresentação do Projeto

Na Seção 1, *Introdução*, apresenta-se a Motivação, a Especificação do problema e objetivo, Escopo e Especificação de Requisitos.

Na Seção 2, *Desenvolvimento*, uma breve descrição da máquina de Turing e do projeto do subtrator são apresentados, em seguida verifica-se a Descrição Formal do projeto, Descrição das Transições e Execução passo-a-passo. Uma breve consideração sobre teste de limite de processamento também compõe esta seção. Tendo em vista os bons resultados alcançados algumas melhorias foram concebidas e descritas em seguida.

Na Seção 3, *Conclusão*, apresenta-se uma análise crítica do projeto, bem como se propõem melhorias.

Na Seção 4, seguem as Referências Bibliográficas.

2. Desenvolvimento

A máquina de Turing é uma máquina de estados finitos (*finite-state machine*) na qual a transição imprime um símbolo na fita. A cabeça de leitura da fita pode mover-se em qualquer direção, permitindo à máquina ler e manipular a entrada tantas vezes quanto for desejado. A estrutura da máquina de Turing é similar ao autômato finito, com a função de transição incorporando esses atributos adicionais [1].

Inicialmente, a fita contém apenas a cadeia de entrada e caracteres em branco (*blank*) por todo o resto. Se a máquina necessitar armazenar informação ela poderá escrever esta informação na fita. Para ler a informação que foi escrita, a máquina pode mover a cabeça de leitura de volta à respectiva posição. A máquina continua o processamento até que decide produzir uma saída. A saída “aceito” ou “rejeitado” é produzida pela entrada em um “estado de aceitação”, ou estado final, ou em um “estado de rejeição”, ou estado não final. Se a máquina não entrar nem em um estado de aceitação nem de rejeição ela seguirá infinitamente, nunca travando (*halting*) [2].

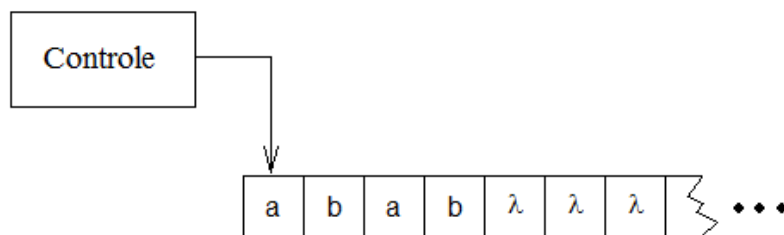


Figura 1 - Esquema da máquina de Turing [2].

Matematicamente, uma Máquina de Turing (modelo básico) é denotada por $M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$ onde:

- Q é o conjunto finito de estados;
- Γ é o alfabeto finito de símbolos da fita;
- $B \in \Gamma$ é o símbolo “em branco” (blank);
- $\Sigma \subset \Gamma - \{B\}$ é o conjunto de símbolos de entrada;
- δ é a função de transição de $Q \times \Gamma$ em $Q \times \Gamma \times \{L,R\}$ (pode ser uma função não definida para alguns argumentos);
- $q_0 \in Q$ é o estado inicial;
- $F \subset Q$ é o conjunto de estados finais.

2.1. Máquina de Turing Subtratora de Inteiros

Antes da operação de subtração da máquina de Turing, efetua-se uma validação da cadeia de entrada, verificando se há 0's e/ou 1's misturados. Após essa validação, aciona-se a máquina de Turing que realiza a subtração de dois inteiros.

O princípio de funcionamento da máquina é simples. A cada iteração, a máquina desloca a cabeça de leitura até o final da segunda cadeia de entrada, lê o último símbolo da cadeia e apaga-o. Se este símbolo for um "1", sabemos que a segunda cadeia é positiva, se for um "0", que é negativa.

Dessa maneira, a máquina desloca a cabeça de leitura até o início da primeira cadeia, lê seu primeiro símbolo e prossegue da seguinte forma:

- Se a primeira cadeia for negativa, e a segunda positiva, adiciona um "0" à primeira cadeia.
- Se a primeira e a segunda cadeias forem negativas, remove um "0" da primeira cadeia.
- Se a primeira cadeia for positiva e a segunda negativa, adiciona um "1" à primeira cadeia.
- Se a primeira e segunda cadeias forem positivas, remove um "1" da primeira cadeia.

A máquina prossegue com estas iterações até que a segunda cadeia se extinga. Aí, a máquina remove o sinal de "-", e retorna a cabeça de leitura ao início da primeira cadeia, que passa a ser o resultado da operação.

No intuito de respeitar o requisito de número 6 que evita que a máquina tente processar uma cadeia equivocada, com mistura de 1's e 0's no mesmo termo da subtração, acrescentou-se uma sub-rotina anterior ao subtrator, denominada *Verificador de Entrada*. Esta rotina varre a cadeia até o símbolo "-" e deste até o final da cadeia, verificando a repetição do mesmo símbolo que inicia cada termo. Qualquer mistura resulta em rejeição da cadeia.

2.2. Descrição Formal

Para a sub-rotina *Verificador de Entrada* temos os seguintes valores de especificação da máquina de Turing:

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$$

$$\Gamma = \{0, 1, -, B(\text{blank})\}$$

$$q_0 = q_1$$

$$F = \{q_7\}$$

A Figura 2 apresenta o Diagrama de Transição da sub-rotina *Verificador de Entrada*, desenvolvido em JFLAP.

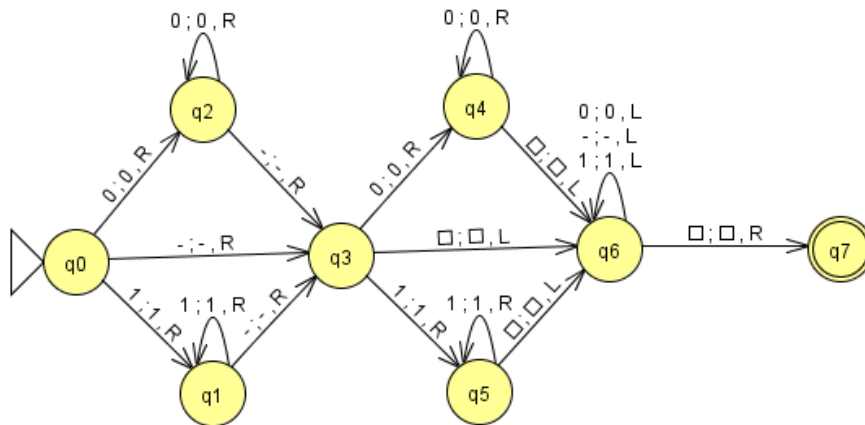


Figura 2 - Diagrama de Transição do Verificador de Entrada.

Para o subtrator desenvolvido no projeto temos os seguintes valores de especificação da máquina de Turing:

$$Q = \{q_1, q_2, q_3, q_4, \text{sub0}, \text{sub1}, \text{sub2}, \text{soma0}, \text{soma1}, \text{soma2}\}$$

$$\Gamma = \{0, 1, -, B(\text{blank})\}$$

$$q_0 = q_1$$

$$F = \{q_4\}$$

A Figura 3 apresenta o Diagrama de Transição do Subtrator, desenvolvido em JFLAP.

Figura 3 - Diagrama de Transição.

Para um melhor entendimento da máquina de Turing subtratora de inteiros a descrição das transições da mesma foi separada da rotina de verificação de entrada. A rotina, descrita a seguir, é muito mais simples que o subtrator e a união das descrições poderia prejudicar o entendimento da máquina principal.

$$1) \delta(q_0, 0) = (q_2, 0, R)$$

$$2) \delta(q_2, 0) = (q_2, 0, R)$$

3) $\delta(q_2, -) = (q_3, -, R)$

$$4) \delta(q_0, 1) = (q_1, 1, R)$$

5) $\delta(q_1, 1) = (q_1, 1, R)$

$$6) \delta(q1, -) = (q3, -, R)$$

No estado q1, lendo “-”, vai para o estado q3, mantém símbolo lido e vai para a direita.

Ainda uma outra possibilidade para q0, a seguir. Para o caso de a cadeia começar com sinal de subtração.

$$7) \delta(q0, -) = (q3, -, R)$$

No estado q0, lendo “-”, vai para q3, mantém símbolo na fita e caminha para a direita.

$$8) \delta(q3, 0) = (q4, 0, R)$$

No estado q3, lendo 0, ir para q4, manter o símbolo na fita e avançar para a direita.

$$9) \delta(q4, 0) = (q4, 0, R)$$

No estado q4, lendo 0, manter q4, manter 0 na fita e ir para a direita. Nesta etapa, caso haja mistura de símbolos no segundo termo da subtração a máquina trava fora do estado final, como desejado.

$$10) \delta(q4, \square) = (q6, \square, L)$$

No estado q4, lendo símbolo em branco, vai para o estado q6, não escreve nada e vai para a esquerda.

$$11) \delta(q3, 1) = (q5, 1, R)$$

Outra transição para o estado q3. No estado q3, lendo 1, ir para q5, manter o símbolo na fita e avançar para a direita.

$$12) \delta(q5, 1) = (q5, 1, R)$$

No estado q5, lendo 1, manter q5, manter 1 na fita e ir para a direita. Nesta etapa, caso haja mistura de símbolos a máquina trava fora do estado final, como desejado.

Ainda há uma outra transição saindo de q3, para o caso de não haver segundo termo na cadeia.

$$13) \delta(q3, \square) = (q6, \square, L)$$

No estado q3, lendo fita em branco, vai para q6, deixa fita em branco e move para a esquerda.

$$14) \delta(q5, \square) = (q6, \square, L)$$

No estado q5, lendo símbolo em branco, vai para o estado q6, não escreve nada e vai para a esquerda.

$$15) \delta(q6, 0 \mid 1 \mid -) = (q6, 0 \mid 1 \mid -, L)$$

No estado q6, lendo 0, 1 ou “-”, manter-se no estado q6, manter símbolo lido e ir para a esquerda. Esta transição permite que a cabeça de leitura volte ao início da cadeia.

$$16) \delta(q6, \square) = (q7, \square, R)$$

No estado q6, lendo símbolo em branco, vai para q7, manter símbolo em branco e ir para a direita. Agora a cabeça de leitura está posicionada no início da cadeia.

A seguir descreve-se cada transição da máquina de Turing subtratora propriamente dita, a fim de possibilitar um entendimento perfeito do processo. Apesar de possuir relativamente poucos estados, esta máquina de Turing mostrou-se bastante eficaz nos testes realizados.

Funções de Transição do Subtrator:

$$1) \delta(q1, - | 0 | 1) = (q1, - | 0 | 1, R)$$

No estado q1, lendo “-“, 0 ou 1, permanecer em q1, manter o símbolo na fita e avançar para a direita. Avança até o final do segundo termo.

$$2) \delta(q1, \square) = (q2, \square, L)$$

No estado q1, lendo fita em branco (*blank*), mudar para estado q2, manter em branco e ir para a esquerda.

$$3) \delta(q2, 0) = (soma0, \square, L)$$

No estado q2, lendo 0, vai para o estado soma0, apaga o símbolo da fita e vai para a esquerda. Já se sabe que o segundo termo é negativo.

$$4) \delta(soma0, - | 0 | 1) = (soma0, - | 0 | 1, L)$$

No estado soma0, lendo “-“, 0, ou 1, manter-se no estado soma0, manter o símbolo lido e caminhar para a esquerda. Volta ao início do primeiro termo. Agora é preciso verificar se o primeiro termo é negativo, o que resultará em uma subtração ($-a-(-b)$), ou se o primeiro termo é positivo, o que resultará em uma soma ($a-(-b)$).

$$5) \delta(soma0, \square) = (soma1, \square, R)$$

No estado soma0, lendo fita em branco, vai para estado soma1, manter fita em branco e avançar para a direita. Posicionou o leitor no início do primeiro termo.

$$6) \delta(soma1, 0) = (q1, \square, R)$$

No estado soma1, lê 0 na fita, vai para q1, apaga símbolo da fita e vai para a direita. Inicia a subtração, que é um tipo de eliminação de símbolos dos dois termos até que se chegue a um “resto” da operação. Ou então...

$$7) \delta(soma1, 1 | -) = (soma2, 1 | -, L)$$

No estado soma1, lendo 1 ou “-“ vai para estado soma2, repete o símbolo lido e vai para a esquerda.

$$8) \delta(soma2, \square) = (q1, 1, R)$$

No estado soma2, ao ler símbolo em branco vai para estado q1, escreve 1 na fita e vai para a direita. Inicia o processo de soma, caracterizado pelo acréscimo de símbolos à esquerda da cadeia original (permitido pelo JFLAP).

A Função de Transição número 3 ainda tem outras duas seqüências, a próxima é descrita a seguir.

$$9) \delta(q2, 1) = (sub0, \square, L)$$

No estado q2, ao ler 1 vai para estado sub0, apaga símbolo da fita e move a cabeça de leitura para a esquerda. Agora se sabe que o segundo termo é positivo.

$$10) \delta(\text{sub0}, - \mid 0 \mid 1) = (\text{sub0}, - \mid 0 \mid 1, L)$$

No estado sub0, lendo “-“, 0 ou 1, manter o símbolo lido e ir para a esquerda. Volta ao início do primeiro termo. Agora é preciso verificar se o primeiro termo é negativo, o que resultará em uma soma (-a-b), ou se o primeiro termo é positivo, o que resultará em uma subtração (a-b).

$$11) \delta(\text{sub0}, \square) = (\text{sub1}, \square, R)$$

No estado sub0, ao ler fita em branco, manter em branco e ir para a direita. Posiciona o leitor no início do primeiro termo.

$$12) \delta(\text{sub1}, 1) = (q1, \square, R)$$

No estado sub1, ao ler 1, vai para estado q1, apaga o símbolo da fita e caminha para a direita. Inicia-se a subtração, que é um tipo de eliminação dos termos da operação a fim de deixar na fita apenas o “resto”. Ou então...

$$13) \delta(\text{sub1}, - \mid 0) = (\text{sub2}, - \mid 0, L)$$

No estado sub1, lendo “-“ ou 0, vai para estado sub2, mantém o símbolo lido e move para a esquerda.

$$14) \delta(\text{sub2}, \square) = (q1, 0, R)$$

No estado sub2, ao ler fita em branco vai para estado q1, escreve 0 na fita e vai para a direita. Inicia o processo de soma, no qual se acrescenta símbolos à esquerda do início do primeiro termo (o que é permitido no JFLAP).

O terceiro desdobramento da Função de Transição número 3 é descrito a seguir.

$$15) \delta(q2, -) = (q3, \square, L)$$

No estado q2, ao ler “-“ vai para q3, apaga símbolo da fita e move para a esquerda. Significa que o segundo termo já foi eliminado, seja por soma ou subtração, restando apenas apagar o símbolo “-“ para que a fita contenha apenas o resultado.

$$16) \delta(q3, 0 \mid 1) = (q3, 0 \mid 1, L)$$

No estado q3, lendo 0 ou 1, mantém q3 e o símbolo lido, move a cabeça de leitura para a esquerda. Move a cabeça de leitura para o início da cadeia (que agora é o resultado).

$$17) \delta(q3, \square) = (q4, \square, R)$$

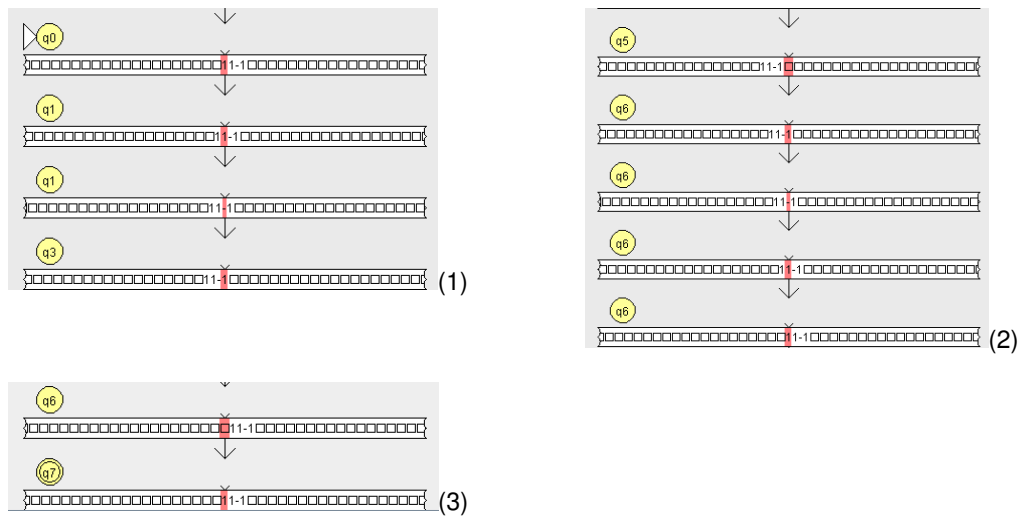
No estado q3, lendo fita em branco, vai para q4 (estado final), mantém fita em branco e move cursor de leitura para a direita, posicionando-o no início da cadeia (resultado).

2.4. Execução passo-a-passo

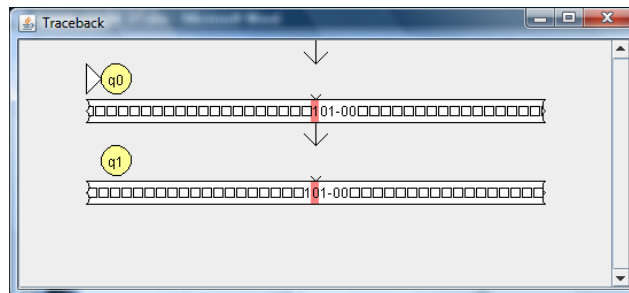
Seguindo o mesmo raciocínio da apresentação das Funções de Transição, para apresentar a execução passo-a-passo da máquina de Turing desenvolvida adotou-se a separação da sub-rotina de verificação da máquina de subtração de inteiros.

A seguir são apresentados três exemplos de aplicação da máquina de Turing *Verificador de Entrada* desenvolvida para este trabalho.

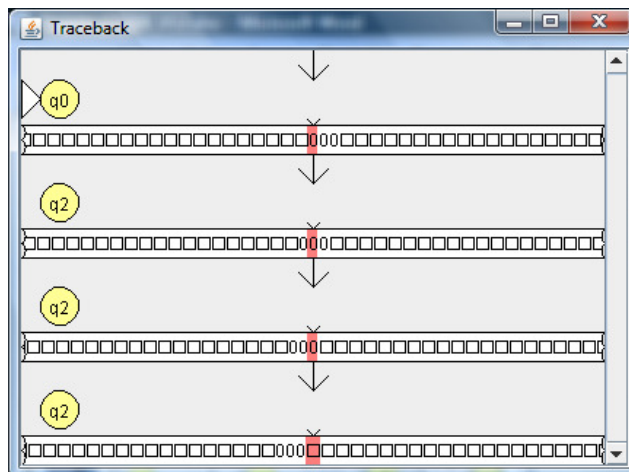
2.4.1. Exemplo 1. Conteúdo inicial da fita: 11-1. Cadeia aceita.



2.4.2. Exemplo 2. Conteúdo inicial da fita: 101-00. Entrada rejeitada por haver mistura de símbolos no primeiro termo.

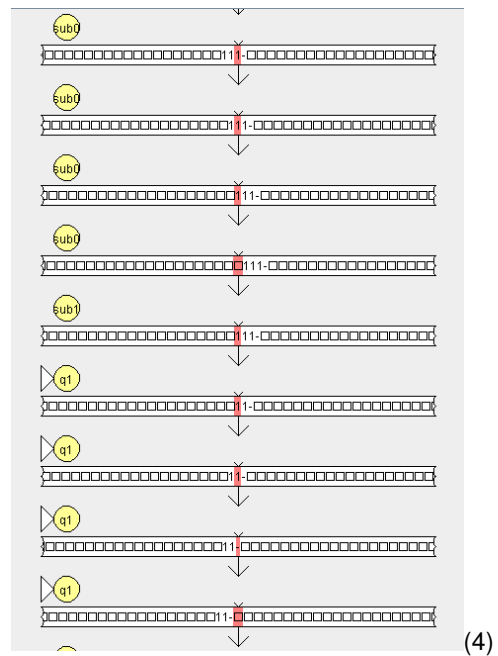
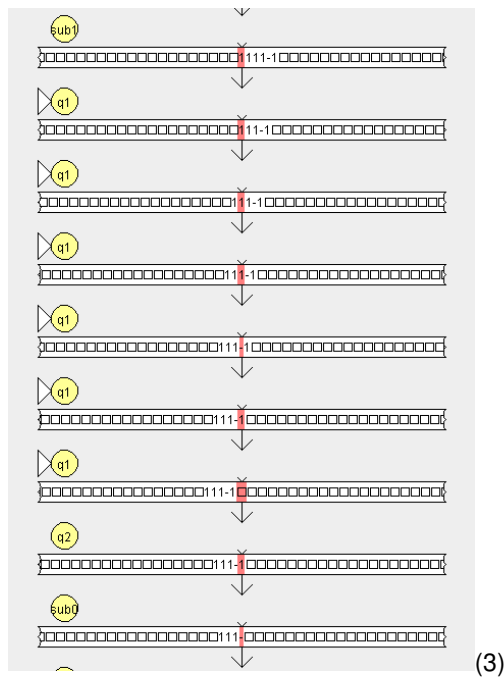
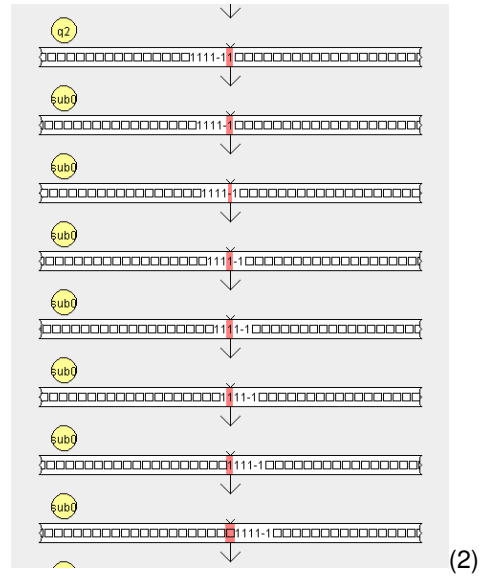
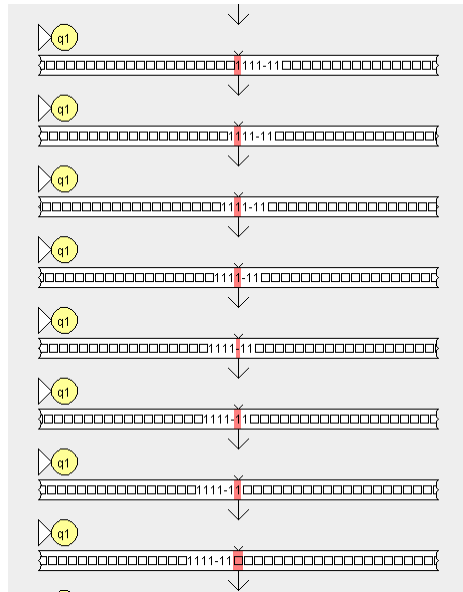


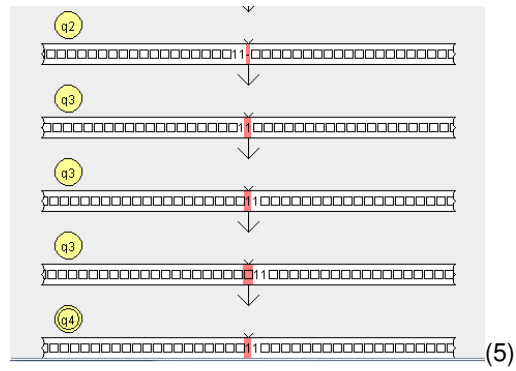
2.4.3. Exemplo 3. Conteúdo inicial da fita: 000. Entrada rejeitada por não conter o sinal de subtração.



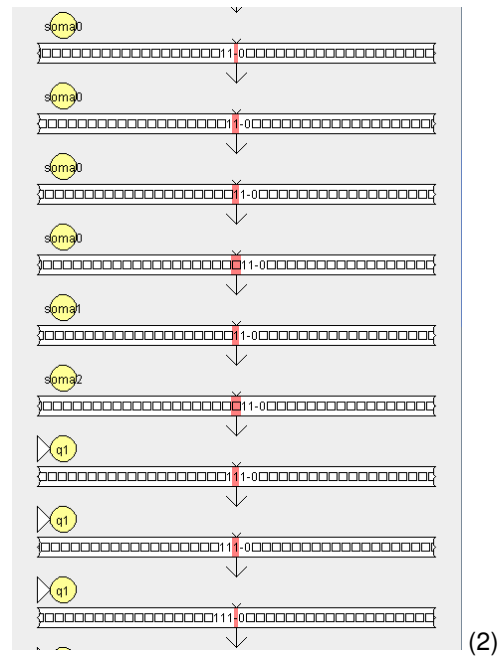
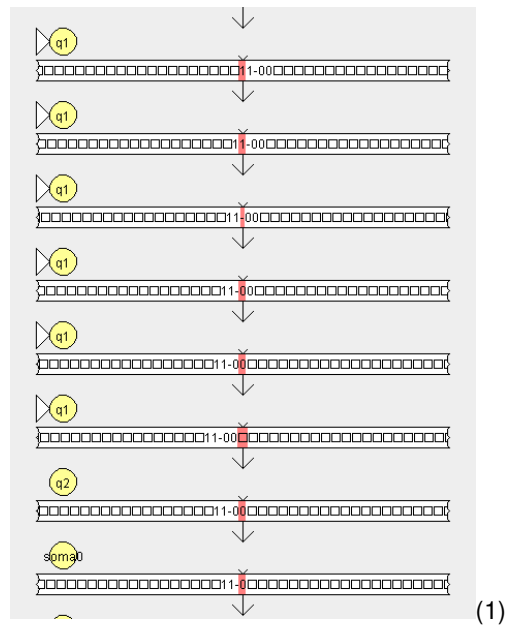
A seguir são apresentados três exemplos de aplicação da máquina de Turing subtratora de inteiros desenvolvida para este trabalho. Considera-se que a fita já passou pelo *Verificador de Entrada*.

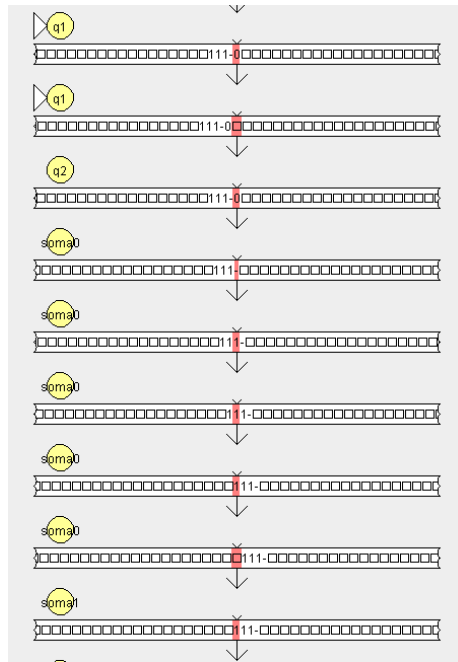
2.4.4. Exemplo 1. Conteúdo inicial da fita: 1111-11, equivalente à operação em números decimais 4-2; resultado esperado: 11, ou 2 em decimais.



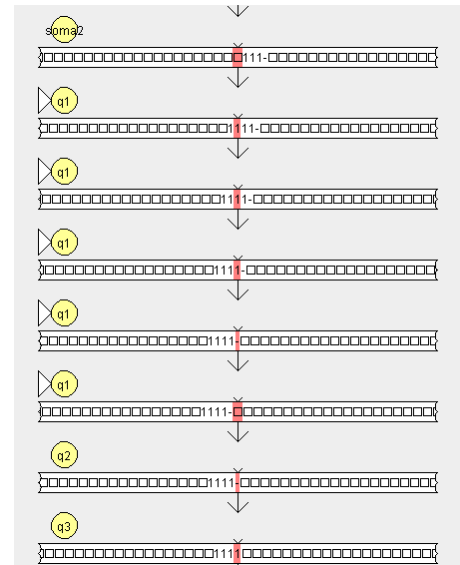


2.4.5. Exemplo 2. Conteúdo inicial da fita: 11-00, equivalente à operação em números decimais $2 - (-2)$; resultado esperado: 1111, ou 4 em decimais.

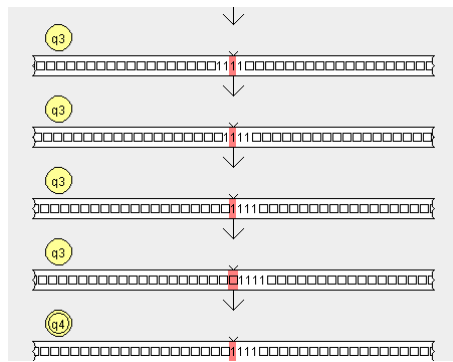




(3)

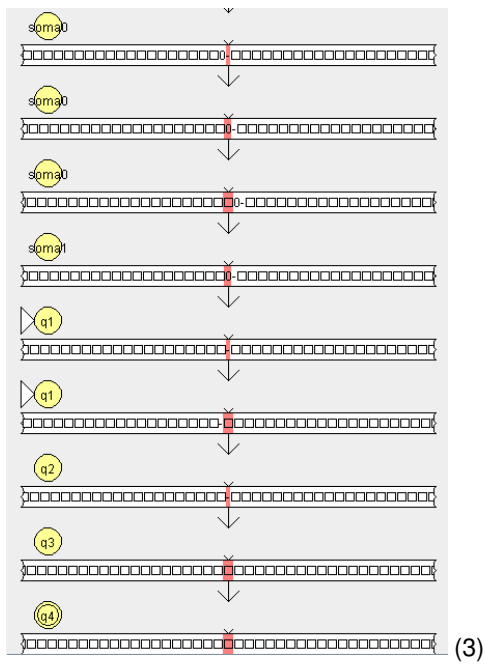
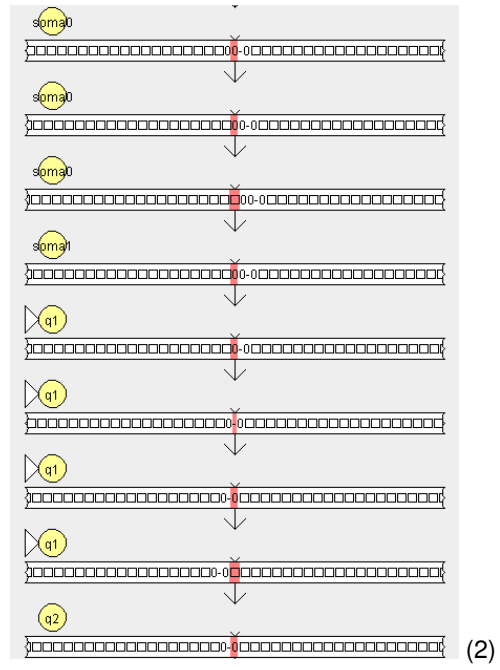
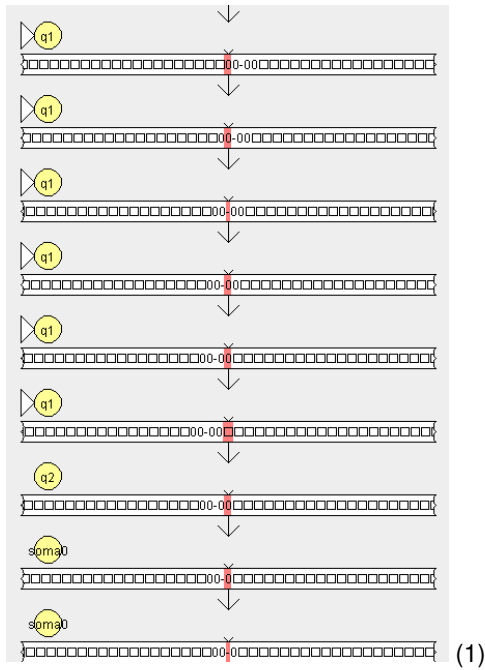


(4)

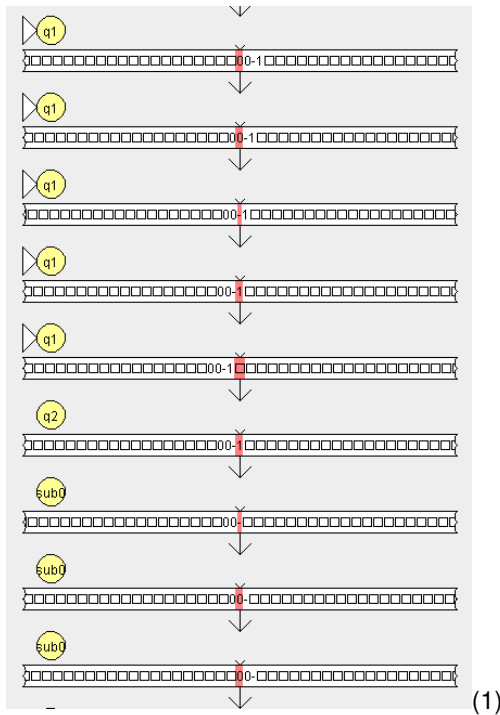


(5)

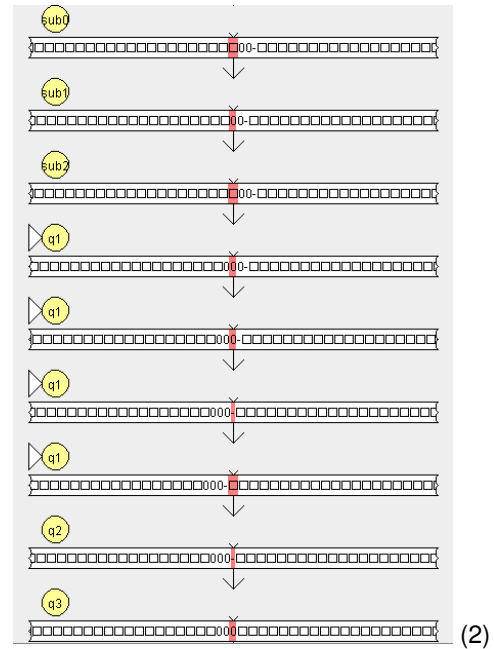
2.4.6. Exemplo 3. Conteúdo inicial da fita: 00-00, equivalente à operação em números decimais $-2-(-2)$; resultado esperado: fita em branco, ou 0 em decimais.



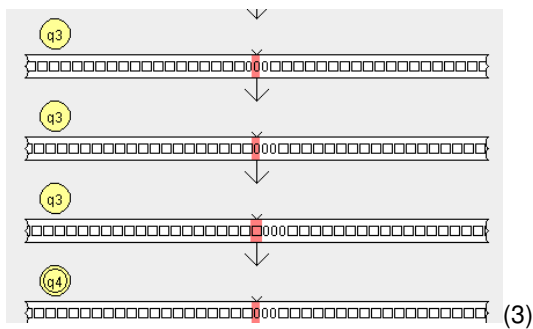
2.4.7. Exemplo 4. Conteúdo inicial da fita: 00-1, equivalente à operação em números decimais -2-1; resultado esperado: 000, ou -3 em decimais.



(1)



(2)



(3)

2.5. Limite de Processamento

Sobre o limite de processamento desta máquina subtratora, sabe-se que há influência de pelo menos os itens abaixo:

- a quantidade de memória disponível no computador;
- limitações do JFLAP;
- a complexidade da máquina de Turing, que acaba forçando o “consumo” tanto em memória quanto do aplicativo.

Foi proposto um teste para se verificar um limite para o subtrator implementado. O subtrator conseguiu tratar a operação $130 - (-130) = 260$. Acima disso o JFLAP acusou erro de processamento.

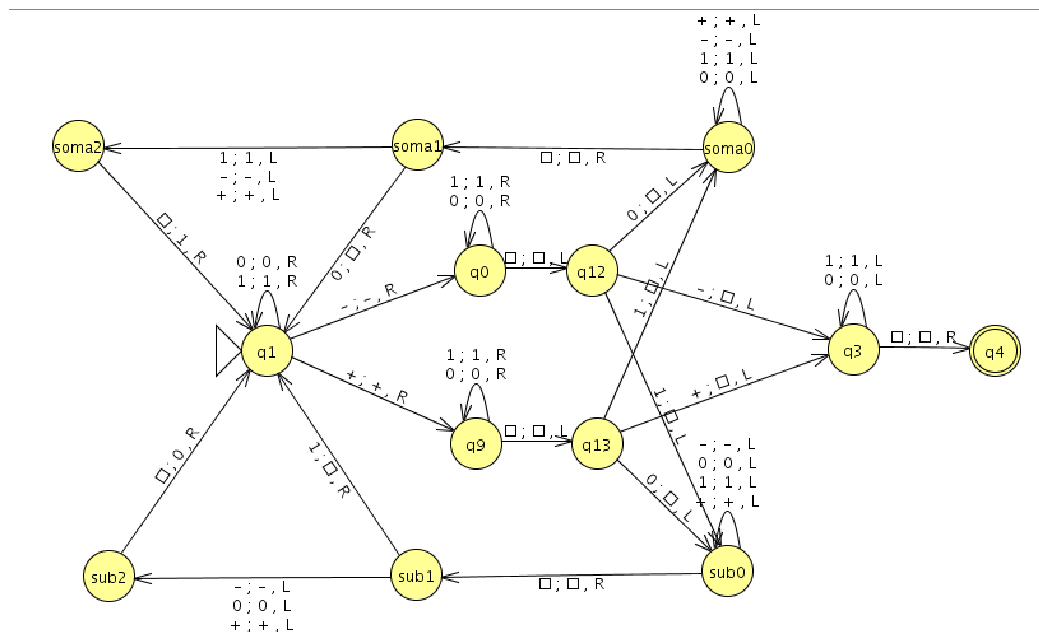
2.6. Melhorias Adicionais

Tendo em vista os bons resultados alcançados com o subtrator de dois inteiros, foram realizados alguns aprimoramentos na máquina, que fogem do escopo original do projeto, mas ilustram possibilidades interessantes. Por fugirem do escopo do projeto original,

apresentamos estes aprimoramentos de forma mais sucinta e sem o rigor de avaliação do subtrator.

2.6.1. Um Somador-Subtrator

Por suportar números negativos a máquina subtratora já possui a essência da lógica necessária para realizar somas, além das subtrações, visto que a soma $X + Y$ equivale à subtração $X - (-Y)$. Desta forma, foi possível, com a adição de apenas 3 estados em relação ao subtrator original, criar um somador/subtrator que, além da subtração com o uso do símbolo “-”, também suporta a soma com o uso do símbolo “+”. Este somador obedece a todas as especificações do subtrator. O Diagrama de Transição desta máquina é apresentada na Figura 4.



Além de gerar os mesmos resultados do subtrator já apresentado, esta máquina efetua soma como exemplificado na tabela a seguir. Não incluímos o verificador de entradas nesta máquina, mas o mesmo pode ser modificado a partir do utilizado no subtrator original de forma trivial.

Tabela 3: Entradas e saídas para o somador-subtrator com uso da soma.

Entradas	Saídas
1111+11	111111
11+000	0
+111	111
111+	111
+	
00+1111	11
000+00	00000

2.6.2. Um subtrator de n inteiros

Apresenta-se a seguir uma estratégia de implementação de um subtrator para n inteiros, aproveitando-se o subtrator desenvolvido como uma sub-rotina. Para esclarecer, alguns exemplos de cadeias de entrada para o subtrator de n inteiros seriam 000-111-11-0-11, 0-1111-1-11, 0-0-11-1 etc. Segue abaixo uma execução da MT.

A estratégia utilizada foi fazer as operações da direita pra esquerda. Por exemplo, considerando que a entrada seja 111-00-1-000-11, a seqüência de processamento seria esta:

1º 111-00-1-(000-11)

2º 111-00-[1-(00000)]

3º 111-[00-[111111]]

4º 111-[0000]

5º 1111111

Para cada passo, há uma estratégia de ação.

1 - Avança o ponteiro da fita até encontrar a última ocorrência de "-" (q1->q2)

2 - Procura-se o símbolo "-" anterior ao encontrado acima

3- Se for achado significa que existe outra subtração a ser realizada depois da subtração corrente e por isso é preciso acionar a sub-rotina *PrepararSub* (q2->prepararSub).

Em seguida, aciona-se o subtrator de dois inteiros, que nos dará a saída 111-000-1-(B...B)00000 com a posição do ponteiro representada pelo símbolo sublinhado.

4- Após isso, caso tenha um B à esquerda da cadeia resultante, deve-se puxar a cadeia de 0's (neste exemplo) resultantes para a esquerda, concatenando-a com o resto da cadeia. Ao final (antes da transição tirarBs->q2) deveremos ter 111-000-1-00000. A partir daí, o processo em q2 será repetido até se chegar na condição descrita no passo 4.

5- Se não achar o símbolo "-" significa que o processo está na última subtração a ser realizada e, portanto, podemos chamar o subtrator no próximo passo (q2->sub).

A sub-rotina *PrepararSub* tem a finalidade de, dada uma cadeia 111-00-1-000-11, com o ponteiro na posição descrita, dar a resposta 111-00-1-(BBBBBB)000-11. Em prepararSub coloca-se uma quantidade de símbolos em branco (B's) igual ao número de elementos da cadeia de entrada a partir da posição inicial do ponteiro (nesse caso 5 elementos(0,0,0,-,1,1)). Isso deve ser feito pelo fato de o subtrator de dois inteiros acrescentar 0's à esquerda em algumas operações o que poderia invalidar os elementos à esquerda da cadeia dada inicialmente. Por isso chama-se a rotina *tirarB'sDoMeio* que retira os B's(blanks) do meio da fita e concatena as cadeias novamente, antes de continuar as operações.

A Figura 5 apresenta o esquema de implementação descrito anteriormente, porém o grupo não chegou a implementar e testar esta máquina de Turing.

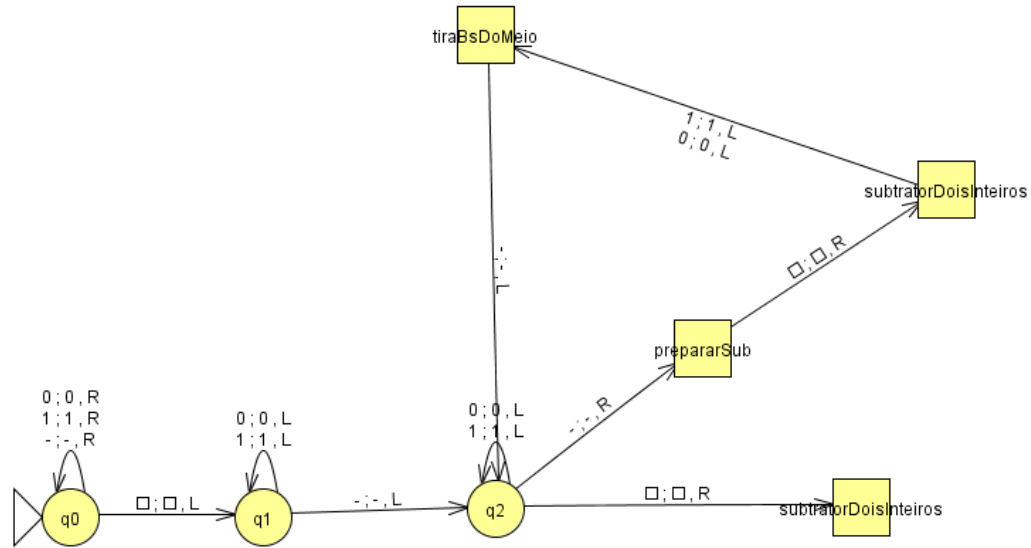


Figura 5 - Diagrama de Transição do Subtrator de n Inteiros.

2.7. Relação de Arquivos

Segue abaixo a relação de arquivos referentes às implementações apresentadas.

- 1) Verificador de entradas: verificador_entrada.jff
- 2) Subtrator de inteiros: subtrator.jff
- 3) Somador-Subtrator de inteiros: subtrator_somador.jff
- 4) Subtrator de n Inteiros: subtratorNinteiros.jff (apenas esquema representativo das sub-rotinas, não foi implementada de fato).

3. Conclusão

Este trabalho teve por objetivo implementar, no aplicativo JFLAP, uma máquina de Turing para subtrair dois números inteiros x e y , de maneira que a entrada do sistema fosse x e y inteiros, separados por um símbolo específico, e a saída fosse $x - y$. Tal máquina de Turing teve finalidade apenas acadêmica, porém trata-se de um assunto muito importante no estudo da computabilidade.

Foram apresentados alguns requisitos mínimos de robustez e coerência estipulados pelo grupo de trabalho.

Em seguida apresentou-se uma breve descrição da teoria da máquina de Turing e do subtrator desenvolvido, bem como o Diagrama de Transição e a evolução do processamento passo-a-passo de quatro exemplos bem sucedidos.

Por causa dos bons resultados alcançados terem estimulado o grupo, foram apresentadas tentativas de aprimoramento da máquina, apesar desse item não fazer parte do escopo do projeto.

Como análise crítica da máquina de Turing desenvolvida pode-se afirmar que foi necessário um processo de amadurecimento de idéias e estratégias até se chegar a esta estrutura de transições deveras compacta e eficiente (tendo em vista os resultados alcançados). Inicialmente, uma máquina bem mais complexa e com várias sub-rotinas foi considerada, pois a estratégia originalmente adotada era de se analisar o sinal das entradas

e condicionalmente direcionar estas entradas para um somador ou subtrator de naturais, implementados como subrotinas. A mudança de estratégia, visando criar uma única máquina que contemplava todos os possíveis casos de entrada de forma intrínseca, mostrou-se acertada, pois culminou numa máquina compacta e de fácil entendimento.

Algumas outras escolhas e características do modelo de Máquina de Turing adotado contribuíram para a eficiência da implementação. A possibilidade de escrever à esquerda da cadeia de entrada, devido ao fato da fita ser infinita em ambas as direções, foi uma delas. A representação de inteiros utilizando um símbolo diferente (o “0”) para números negativos e a cadeia vazia para o número 0 foi outra. Caso uma representação diferente fosse desejada, ela poderia ser facilmente convertida na que utilizamos, com a adição de etapas de conversão.

Destacamos também o fato de que não terem sido utilizados símbolos intermediários ou temporários no processamento da máquina. Assim a qualquer momento da iteração a cadeia representada na fita é uma entrada válida para a máquina. Isto possibilitou o processo de recursão que contribuiu para a elegância da implementação. Além de ter dispensado a necessidade de uma etapa de pré ou pós-processamento da cadeia, fora a simples remoção do sinal no fim da operação.

Recomenda-se o aprimoramento do subtrator e das máquinas de Turing desenvolvidas adicionalmente, incluindo o tratamento de números representados em binário para uma maior eficiência.

4. Referências Bibliográficas

[1] Sudkamp, T. A. Languages and Machines – An introduction to the Theory of Computer Science. 2nd Ed. Addison-Wesley. 1997.

[2] Sipser, M. Introduction to the Theory of Computation, 2nd Ed., Thomson Course Technology, 2006.