Linguagens Formais e Autômatos

Prof. Alexandre Donizeti Alves

Aplicação de autômatos finitos no jogo Super Mario World

Andre Kenji Sato, Leonardo Severgnine Maioli, Marco Antonio Figueira Barbosa

11 de abril de 2024
Universidade Federal do ABC (UFABC)
Av. dos Estados, 5001 - Bairro Bangu - Santo André - CEP: 09280-560

{a.sato, l.severgnine, figueira.marco}@aluno.ufabc.edu.br

Sumário

1. Introdução

Os autômatos finitos são modelos matemáticos utilizados para representar sistemas computacionais simples, com uma quantidade limitada de memória, mas que podem ser utilizados para modelar muitas tarefas úteis.

Comumente, os autônomos finitos são representados por máquinas de estados devido a facilidade de compreender sua representação. Ao observar uma máquina de estados, é fácil identificar o estado inicial do autômato, as transições entre os estados e os possíveis estados finais. No entanto, apesar dessa simplicidade oferecida pela representação por máquinas de estados, uma definição formal é de suma importância para solucionar quaisquer incertezas sobre o que é permitido ou não em um autônomo finito, trazendo maior precisão na sua representação.

A definição formal trazida por [SIPSER 2005] diz que um autômato finito é uma 5-upla $(Q, \sum, \delta, q0, F)$, onde:

1. Q é um conjunto finito denominado os estados do autômato,
2. $\sum$ é um conjunto finito denominado o alfabeto,
3. δ: $Q \times \sum \rightarrow Q$ é a função de transição,
4. $q0 \in Q$ é o estado inicial, e
5. $F \subseteq Q$ é o conjunto de estados de aceitação.

Juntamente com essa definição formal, também é utilizado o conceito de entrada como sendo uma espécie de fita de elementos pertencentes ao alfabeto, guardando portanto as informações a serem processadas pelo autômato. Dessa forma, o funcionamento de um autômato finito é determinado pela entrada que ele recebe e pelo estado em que ele se encontra. O autônomo começa em um estado inicial e, a cada passo, consome um símbolo da entrada e muda de estado de acordo com a função de transição. Se após consumir toda a entrada o autômato estiver em um estado de aceitação, a entrada é aceita e caso contrário, a entrada é rejeitada.

Os autômatos finitos possuem uma vasta área de aplicação dentro do campo de ciência da computação e diversos outros. Algumas das aplicações mais comuns ocorrem em projetos de linguagem de programação, no processamento de linguagem natural, em análise de texto, em projetos de circuitos digitais, em protocolos de comunicação… Além dessas aplicações, os autômatos finitos também são muito utilizados na área de jogos em atividades como modelar o comportamento de NPCs (Personagens não jogáveis), controlar os estados de um jogo, modelar conversas entre jogadores e personagens e muitas outras.

Nesse contexto, o objetivo deste trabalho é aplicar os conceitos que estão sendo vistos na disciplina de “Linguagens Formais e Autômatos” na prática. Para isso, foi selecionado um típico problema real que é a modelagem dos estados de um jogo por meio de um autômato finito. Com base nesse problema escolhido, o grupo construiu uma aplicação em *Python* via Google Colab para simular os possíveis estados do personagem Mario em uma fase do tradicional jogo de plataforma “Super Mario World”, assim como as transições que ocorrem entre esses estados de acordo com os comandos do jogador.

2. Descrição do Problema

Para alcançar o objetivo deste trabalho, foi selecionado um típico problema real que é a modelagem dos estados de um jogo por meio de um autômato finito. O grupo escolheu o tradicional jogo de plataforma “Super Mario World”, no qual é possível simular as ações do personagem Mario e as

possíveis transições entre os estados com um autômato finito, de acordo com os comandos do jogador. O interesse pela modelagem das ações do Mario com um autômato finito, se dá pelo objetivo final de poder avaliar se uma sequência de entrada com comandos de um jogador pode fazer o Mario passar ou não de fase.

Dessa maneira, para a explorar esse problema foi escolhida a fase 1-1 do jogo como ponto de partida para se pensar todas as ações a serem consideradas e quais os estados seriam implementados no autômato, assim como as transições entre esses. Como a intenção do autômato é planejar como prosseguir na fase, tendo em vista que existem diversas maneiras de concluir essa tarefa, a equipe criou um fluxograma da fase, que ilustra todos os estados e todas as transições com suas respectivas equivalências na legenda do fluxograma. Dentro do fluxograma, evitou-se a adição de transições de laço para não poluí-lo ainda mais. A figura 1 traz uma parte desse fluxograma que serviu como esboço para a criação do autômato e que também é muito útil para simplificar o entendimento do problema escolhido.

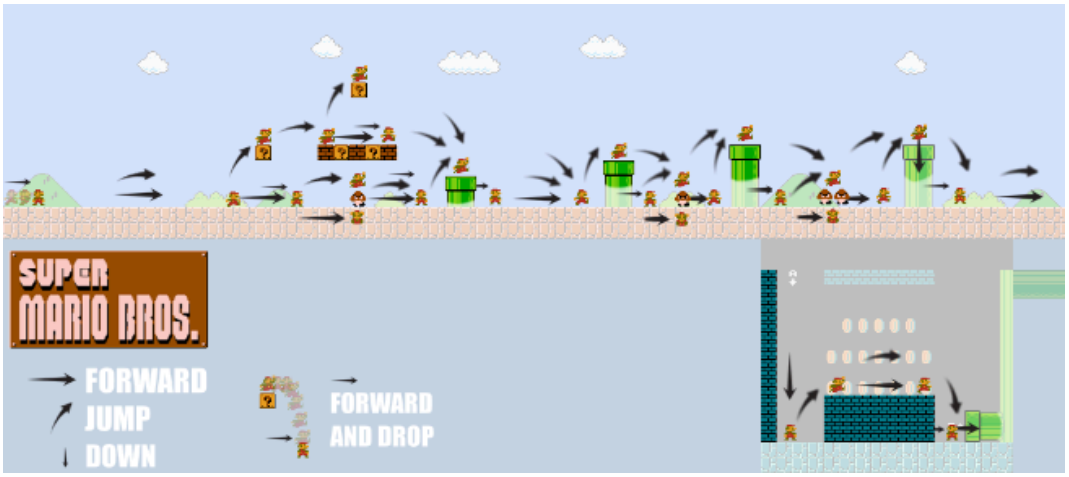

Figura 1. Parte do fluxograma utilizado como esboço do autômato finito com a legenda para as possíveis transições (setas) entre os estados (Marios). O fluxograma firstLevel.png pode ser acessado na integra em: https://drive.google.com/drive/folders/1DCsLhXEeLuSLRdVZ25uAWOqSOn5xvtQE?usp=sharing

Partindo das ideias e definições obtidas com o fluxograma, para que o problema fosse modelado usando autômatos finitos, algumas etapas cruciais foram necessárias. Inicialmente, o grupo definiu quais as possíveis ações do personagem Mario seriam consideradas para esse projeto, e consequentemente, quais os comandos do jogador iriam compor o alfabeto a ser aplicado no autônomo. A tabela 1 traz as ações que foram consideradas e os respectivos comandos do jogador que levam a essas ações.

Os comandos foram denotados por meio de números de um teclado, pois torna-os mais intuitivos, uma vez que 5 é considerada a posição central num teclado numérico. Desta forma, se 5 é neutro, então 6 é adiante, 4 é atrás, 2 é embaixo e 8 é acima. Com essa mesma ideia, 9 está adiante de 8 e acima de 5, logo simboliza adiante e acima.

Ações do Mario	Comandos do jogador (teclas)
Ir para frente	6
Pular para frente	9
Agachar	2

Tabela 1. Lista de ações do Mario e os respectivos comandos do jogador que geram essas ações

É importante destacar que as ações escolhidas para o Mario visam apenas a sua progressão na fase, de forma que o Mario alcançar o final da fase é o mais importante, independente dele realizar outras tarefas como pegar muitas moedas. Dessa maneira, muitas das ações do Mario existentes no jogo, não foram consideradas para o escopo deste trabalho com o intuito de simplificar o autômato resultante e não prejudicar a visualização e entendimento de todos os estados e transições. Com apenas três ações escolhidas, o alfabeto foi reduzido, facilitando a criação de um autônomo finito determinístico. Além disso, essas três ações são suficientes para o Mario poder completar a fase escolhida, mesmo não sendo da forma mais adequada possível.

Alguns exemplos de ações removidas do escopo do projeto foram:

- **Movimentos para trás:** Todas as ações que envolviam movimentos para trás foram removidas porque toda vez que o jogador andasse para trás, ele teria que eventualmente andar para frente ou pular para frente como uma forma de compensar o retrocesso, pois andar para trás não ajuda na progressão da fase. Portanto, assumindo que W seja uma cadeia qualquer aceitável em um autômato com movimentação em todas as direções, seria possível observar que nesta mesma cadeia teria-se que andar para frente toda vez que se anda para trás. Com isso, no contexto desse projeto é mais prático simplesmente apenas mover-se para frente. Além disso, uma vez que a câmera do jogo avança, ela não recua, logo, mesmo que o jogador tente voltar, em muitos casos isso se torna impossível.
- **Saltos na vertical:** Os diferentes tipos de saltos na vertical também não foram considerados, pois são movimentos que não auxiliam na progressão do Mario na fase.
- **Ações de "pegar um item":** O ato de pegar um item como um cogumelo, moeda ou uma flor não foi considerado devido ao fato de que quando o Mario obtém itens específicos, ele ganha novas ações antes não disponíveis, como lançar bolas de fogo. Isso aumentaria o nível de complexidade do problema pois o autômato se tornaria variável ao longo do jogo.
- **Ações oriundas de pegar um item:** Ações como lançar bolas de fogo que são disponibilizadas apenas após pegar algum item não foram consideradas, pois necessitaria a implemetação de um autômato variável.

Apesar de também parecer inútil no contexto de progressão do Mario na fase, o comando agachar foi implementado pois este oferece um atalho interessante dentro da fase, pois permite que o Mario acesse uma parte subterrânea a partir de um cano específico, cortando caminho no processo.

Com as ações e comandos definidos, o passo seguinte foi definir os estados do autômato, assim como as indicações do estado inicial e os estados de aceitação. Os estados do autômato foram criados para que pudessem simbolizar os diferentes Marios em diferentes posições da fase com 96 estados no total.

Os estados criados foram denotados da seguinte forma:

- Todos os estados são representados por uma ou duas letras seguidas de um número de um número entre 0 a 95.
- O estado inicial é ***q0*** e o estado final é ***gg90***.
- Os estados denotados por ***q*** indicam que o Mario está no piso principal da fase.
- Os estados denotados por ***p*** indicam que o Mario está um nível acima do piso principal, por ter subido em um bloco ou em uma escada.
- Os estados denotados por ***o*** indicam que o Mario está dois níveis acima do piso principal.
- Os estados denotados por ***u*** indicam que o Mario está na parte subterrânea da fase.
- Os estados denotados por ***s*** simbolizam estados em que o Mario acabou de derrotar um monstro (***s*** corresponde a "stomp", pois o Mario pisa nos seus oponentes para derrotá-los).
- Os estados denotados por ***d*** representam estados em que o Mario foi derrotado por um monstro ou caiu num buraco (***d*** corresponde a "*defeated*", pois Mario foi derrotado). Se uma sequência de comandos levar para um estado denotado por ***d***, o automato é rejeitado, pois o Mario morreu.

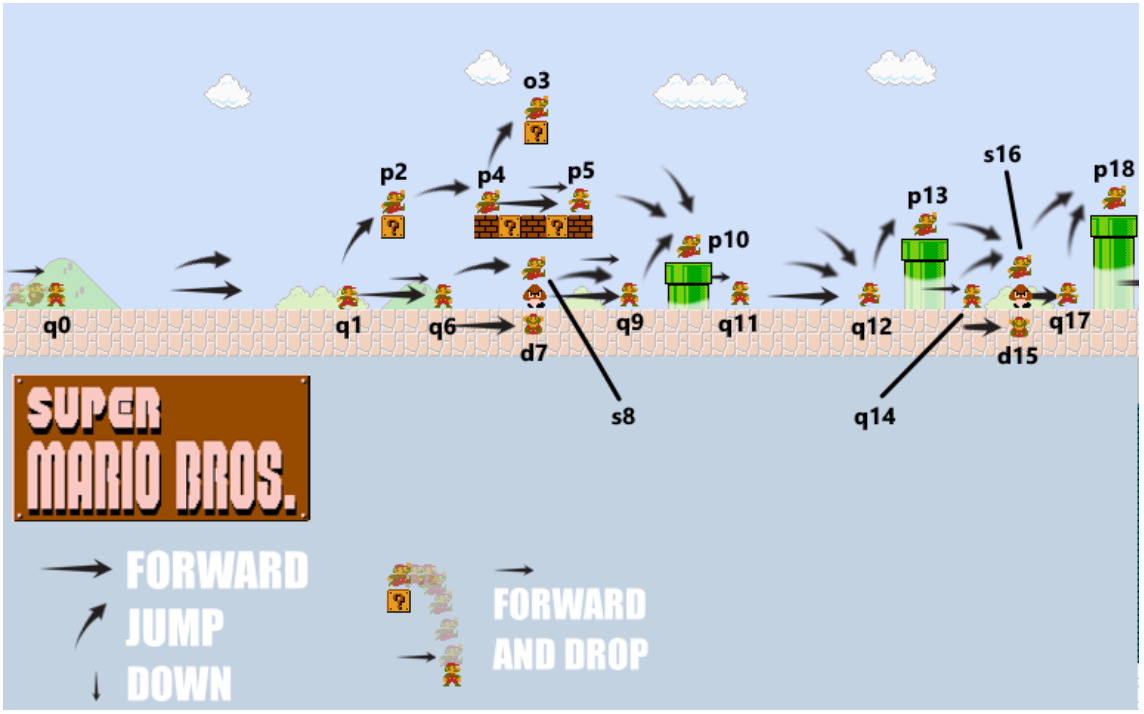

Figura 2. Parte do fluxograma com os nomes dos estados correspondentes. O fluxograma *levelStates.png* pode ser acessado na integra em: https://drive.google.com/drive/folders/1DCsLhXEeLuSLRdVZ25uAWOqSQn5xvtQE?usp=sharing

Concluída as etapas anteriores, para que o autômato finito utilizado possa ser definido formalmente, resta apenas a elaboração das transições entre cada estado. A Tabela 2 foi então elaborada para indicar todas as transições existentes.

δ	6	9	2	δ	6	9	2	δ	6	9	2	δ	6	9	2
q0	q1	q1	q0	**q70**	q70	p71	q70	**p67**	q69	q70	p67	**s21**	q22	p23	s21
q1	q6	p2	q1	**q74**	q75	p76	q74	**p71**	d72	p73	p71	**d20**	d20	d20	d20
q6	d7	s8	q6	**q75**	q75	p76	q75	**p73**	q74	q75	p73	**d51**	d51	d51	d51
q9	q9	p10	q9	**q77**	q80	p78	q77	**p76**	q77	p78	p76	**s30**	o32	o32	s30
q11	q12	q12	q11	**q80**	d82	s81	q80	**p78**	p79	p79	p78	**d31**	d31	d31	d31
q12	q12	p13	q12	**q83**	q83	p84	q83	**p79**	s81	s83	p79	**d52**	d52	d52	d52
q14	d15	s16	q14	**q88**	q89	gg90	q88	**p84**	p85	p86	p84	**s39**	q42	p41	s39
q17	q17	p18	q17	**q89**	q89	gg90	q89	**p85**	p85	p86	p85	**d40**	d40	d40	d40
q19	d20	s21	q19	**p2**	q6	p4	p2	**p86**	p86	o87	p86	**s44**	q43	p46	s44
q22	q22	p23	q22	**p4**	p5	o3	p4	**o3**	p5	p10	o3	**d45**	d45	d45	d45
q24	q25	q25	q24	**p5**	q9	p10	p5	**o29**	d31	s30	o29	**s49**	q53	p48	s49

q25	d51	q26	q25	**p10**	q11	q12	p10	**o32**	q36	o33	o32	**d50**	d50	d50	d50
q26	q27	q27	q26	**p13**	q14	s16	p13	**o33**	o34	o34	o33	**s61**	d64	s63	s61
q27	q35	p28	q27	**p18**	q19	s21	p18	**o34**	s39	p41	o34	**d62**	d62	d62	d62
q35	d52	q36	q35	**p23**	q24	q24	u91	**o47**	p48	q53	o47	**s63**	q65	p59	s63
q36	q38	p37	q36	**p28**	q35	o29	p28	**o55**	o56	o56	o55	**d64**	d64	d64	d64
q38	d40	s39	q38	**p37**	q38	p41	p37	**o56**	s63	o57	o56	**s81**	q83	p84	s81
q42	d45	p43	q42	**p41**	q42	p43	p41	**o57**	o58	o58	o57	**d82**	d82	d82	d82
q43	d50	s49	q43	**p43**	s44	o47	p43	**o58**	q65	p66	o58	**d72**	d72	d72	d72
q53	q60	p54	q53	**p46**	s49	p48	p46	**o87**	q88	gg90	o87	**u91**	u91	u93	u91
q60	d62	s61	q60	**p48**	q53	p54	p48	**s8**	q9	q9	s8	**u93**	u94	u94	u93
q65	q65	p66	q65	**p54**	q60	o55	p54	**d7**	d7	d7	d7	**u94**	u92	u92	u94
q68	q68	p67	q68	**p59**	q65	q65	q59	**s16**	q17	p18	s16	**u92**	p76	u92	u92
q69	q70	q70	q69	**p66**	q68	p67	p66	**d15**	d15	d15	d15	**gg90**	gg90	gg90	gg90

Tabela 2. Tabela de transição entre os estados

Dessa maneira, a definição formal do autômato finito determinístico para a simulação dos estados do personagem Mario é a 5-upla $(Q, \sum, \delta, q0, F)$, onde:

- Q = {q0, q1, q6, q9, q11, q12, q14, q17, q19, q22, q24, q25, q26, q27, q35, q36, q38, q42, q43, q53, q60, q65, q68, q69, q70, q74, q75, q77, q80, q83, q88, q89, p2, p4, p5, p10, p13, p18, p23, p28, p37, p41, p43, p46, p48, p54, p59, p66, p67, p71, p73, p76, p78, p79, p84, p85, p86, o3, o29, o32, o33, o34, o47, o55, o56, o57, o58, o87, s8, d7, s16, d15, s21, d20, d51, s30, d31, d52, s39, d40, s44, d45, s49, d50, s61, d62, s63, d64, s81, d82, d72, u91, u93, u94, u92, gg90}
- $\sum = \{2, 6, 9\}$
- δ é definida pela Tabela 2.
- $q0 = q0$
- $F = \{gg90\}$

Os dados obtidos com essa definição formal do autômato foram então utilizados em uma aplicação python implementada no google colab. Mais detalhes específicos da aplicação implementada podem ser observados na seção seguinte ou no próprio arquivo do Google Colab. A Figura 3 traz a representação de parte do autômato criado para a fase escolhida, gerado a partir do programa JFLAP.

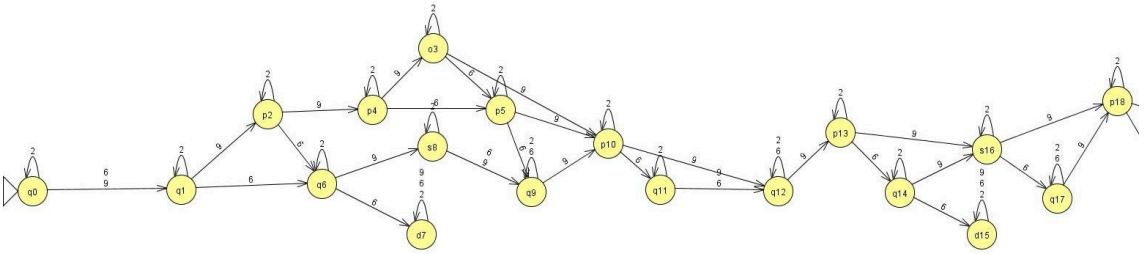

Figura 3. Autômato parcial gerado pelo programa JFLAP. O automato *firstLevel.jpg* pode ser acessado na integra em:
https://drive.google.com/drive/folders/1DCsLhXEeLuSLRdVZ25uAWOqSQn5xvtQE?usp=sharing

3. Descrição da Aplicação

A partir do problema descrito na seção anterior e da definição formal do autômato finito determinístico para a resolução desse problema, foi criada uma aplicação em *Python* para simular os possíveis estados do personagem Mario, assim como as transições que ocorrem entre esses estados de acordo com os comandos do jogador. Essa aplicação criada permite que uma sequência de comandos de entrada seja avaliada, podendo ser válida caso forneça ações que levam o Mario a alcançar a bandeira e passar de fase ou inválida se as ações levam o Mario à morte (estados denotados por ***d***) ou gere indeterminações.

A primeira etapa realizada na implementação foi a importação da biblioteca chamada "*automathon*" que é utilizada para a simulação e visualização de autômatos finitos. Essa biblioteca oferece uma variedade de funcionalidades para criar, manipular, minimizar e verificar autômatos.

Com base na definição formal estabelecida para o autônomo na seção 2, nota-se que o autômato finito para a simulação dos estados do personagem Mario possui uma natureza determinística. Por esse motivo, na sequência da implementação foi importada a classe DFA (inglês para AFD) da biblioteca "*automathon*", para que fosse possível criar e manipular autômatos finitos determinísticos. Dessa forma, o AFD foi implementado passando as definições do conjunto de estados, do alfabeto, da função de transição, do estado inicial e do conjunto dos estados de aceitação a partir do comando:

```
automata = DFA(q, sigma, delta, initial_state, F)
```

que cria um objeto chamado "automata" da classe DFA que pode então ser usado para realizar operações como verificar se uma entrada é aceita pelo autômato ou não, ou verificar a validade do próprio autômato em termos da estrutura definida para ele.

Após verificar a validade do autômato criado com o comando:

```
automata.is_valid()
```

a próxima etapa da implementação foi a visualização do AFD criado com o comando:

```
automata.view("afd01")
```

que é uma chamada para a visualização padrão do autômato em formato de uma máquina de estados.

Com esse autômato implementado e com base no fluxograma da figura 1, o grupo definiu algumas sequências de entrada para simular possíveis sequências de comandos utilizadas por um jogador e explorar situações que acontecem com o Mario na fase 1-1. Para testar essas sequências, utilizou-se do comando:

```
automata.accept("XXXXXXXX")
```

que retorna "*True*" caso a sequência seja aceita pelo autômato e "*False*" caso contrário. Entradas aceitas simbolizam sequências de comandos do jogador que levam o Mario a passar de fase. Entradas negadas

indicam que o Mario não passou de fase, seja porque foi derrotado, caiu em um buraco ou entrou em um estado de indeterminação durante o processamento da sequência de comandos. Os testes feitos se encontram no projeto feito no arquivo do Google Colab.

Algumas das situações testadas e suas respectivas sequências de comando foram:

- **Mario derrotado quando apenas se movimenta para frente**

  Para simular a situação em que o Mario se movimenta apenas para frente foi utilizada a sequência de comandos "666". Avaliando essa sequência de comandos, pode-se verificar que aplicando a função de transição sequencialmente, a sequência de estados do Mario são: $q0 \rightarrow q1 \rightarrow q6 \rightarrow d7$. Dessa forma, após a sequência de comandos aplicada, o Mario fica em um estado que não é de aceitação e o autômato rejeita a entrada.

- **Mario vence a fase pegando o atalho pelo cano**

  Para simular a situação em que o Mario consegue vencer a fase utilizando o atalho pelo cano, foi utilizada a sequência "669699999992966696696996 69". Aplicando a função de transição para essa entrada, a sequência de estados do Mario são: $q0 \rightarrow q1 \rightarrow q6 \rightarrow s8 \rightarrow q9 \rightarrow p10 \rightarrow q12 \rightarrow p13 \rightarrow s16 \rightarrow p18 \rightarrow s21 \rightarrow p23 \rightarrow u91$

  $\rightarrow u93 \rightarrow u94 \rightarrow u92 \rightarrow p76 \rightarrow p78 \rightarrow p79 \rightarrow s81 \rightarrow p84 \rightarrow p85 \rightarrow p86 \rightarrow o87 \rightarrow$

  $\rightarrow q88 \rightarrow q89 \rightarrow gg90$. Dessa forma, após a sequência de comandos aplicada, o Mario consegue pegar a bandeira e vencer a fase, encerrando no estado de aceitação "gg90". Consequentemente o autômato aceita a entrada.

O grupo também criou um vídeo com uma simulação realizada para acompanhar o autômato enquanto um jogador joga a primeira fase. O vídeo está disponível em: https://drive.google.com/file/d/1n3uGTQBLklvUXqfI8lP940mCvcNZWAgF/view?usp=drive_link

4. Considerações Finais

Devido a limitações de tempo e de conhecimento teórico, vários aspectos presentes no jogo "Super Mario World" não foram consideradas durante a elaboração do autômato, como efeitos oriundos da obtenção de itens, os quais possibilitam que o Mario execute ações extras, os quais necessitam da implementação de um autômato variável. Portanto, uma melhoria possível para o trabalho seria explorar o uso de um autômato variável para a fase, permitindo o Mario mudar de autômato durante o processamento da sequência de comandos, conforme ele ganha (ao tocar em um item) ou perde poderes (ao tocar em um inimigo).

Além disto, é possível complementar o trabalho feito adicionando outros estados presentes no jogo, mas que não foram considerados no trabalho, como chutar o casco do Koopa Troopa (tartaruga), ser atingido pelo casco, ser atingido por projéteis, entre outros.

5. Referências

1. [SIPSER 2005] SIPSER, M. - "Introdução à Teoria da Computação", Editora Cengage, Tradução da 2ª edição norte-americana, 2005.
2. Super Mario World Speedrunning Wiki. The great seal of Super MarioWorld, 20 set. 2022. Disponível em: https://smwspeedruns.com/Main_Page. Último acesso em: 06 abril 2024.
3. https://www.jflap.org/