



Contexto

Produção de Poções

A produção de poções é o principal negócio de uma determinada família de bruxos e bruxas, no qual são produzidas e comercializadas diversas poções simples, complexas e variadas. Com a recente alta na demanda de produtos, vocês foram convocados para realizar o presente trabalho e auxiliar no controle de qualidade dos negócios da família de bruxos (enquanto aprendem ainda mais sobre a diferença de poder computacional entre máquinas de estados).

Para isso, vocês deverão construir e implementar diferentes autômatos para diferentes produções de poções, desde as mais simples, se importando apenas com os ingredientes, até às mais complexas, se atentando às propriedades de cada um dos mesmos.

Neste trabalho **você e seu grupo (definidos em aula da disciplina [nesta planilha](#))** deverão implementar um programa que seja capaz de **simular a fabricação de poções utilizando Máquinas AFD e de pilha**. Outras máquinas poderão ser implementadas seguindo seus próprios critérios como funcionalidades extras ao programa. Quanto mais detalhadas as máquinas obrigatórias e extras, e quanto mais evidente a diferença de poder computacional entre as mesmas, melhor poderá ser considerado seu trabalho. Tudo deve ser devidamente documentado e entregue em 3 arquivos (PDF da documentação, PDF dos slides e todo o código do programa dentro de um arquivo .zip) **no PVANet Moodle**. **É necessário que apenas um integrante de cada grupo faça a entrega, mas na documentação devem constar os nomes completos e números de matrícula de cada um**. Além disso, vocês podem implementar mais detalhes que não estejam especificados aqui, desde que devidamente documentados, e até mesmo outras opções de interface.

Observações:

- 1) A linguagem de programação pode ser escolhida pelo grupo e deve ser justificada na documentação. Tudo o que for usado de materiais prontos ou de terceiros, deve ser documentado e referenciado, mas o ideal é que o grupo faça toda a implementação.
- 2) Documento, de forma detalhada, como utilizar a interface do seu programa.

Especificação

Cada grupo deverá criar máquinas de estados de diferentes tipos para representar a criação de poções a partir de ingredientes e receitas. Cada tipo de autômato utilizado deve explorar

ao máximo seu poder computacional, evidenciando a diferença de complexidade entre cada um.

Para a realização do trabalho, deverão ser definidos os ingredientes utilizados nas poções, de forma que um alfabeto seja definido para a criação das receitas. Cada receita será representada a partir de uma linguagem (LR, LLC, etc).

Segue abaixo o funcionamento básico das máquinas:

1. Cada transição da máquina representa a adição de um novo ingrediente;
2. Cada estado é nomeado conforme o grupo definir, com exceção do estado inicial e final, que devem ter o nome **I** e **F**, respectivamente;
3. Cada transição leva a um estado que pode ou não exibir as características atuais da poção, alguma alteração visual ou aromática da mesma, por exemplo. Essas características devem ser exploradas de forma diferente por autômatos de diferentes tipos, evidenciando o poder computacional único de cada um.
4. No final da adição de todos os ingredientes deverá ser exibida uma mensagem de aceitação ou erro, seguida de uma pequena descrição da receita seguida.
5. Não podem haver transições inválidas, ou seja, a máquina **deve ser completa**, apresentando todas as transições possíveis de todos os estados.
6. No trabalho devem ser criados, no mínimo, 6 ingredientes e cada autômato deve ser capaz de ler pelo menos 3 desses ingredientes.

AFD:

Restrições predeterminadas:

- cada poção deverá ter pelo menos 3 ingredientes, portanto cada máquina deve possuir pelo menos 3 transições;
- todos os autômatos devem possuir um estado de erro;
- Cada máquina deve explorar ao máximo seu poder computacional e funcionalidades únicas, seja criativo ao definir como farão isso.

Entrada:

Dando um exemplo para as entradas, assuma que existem leituras de **p (pétalas)**, **a (água)** e **o (óleo)**. Lembre-se que a máquina deve ser completa. Seu programa deve então receber um arquivo de entrada no seguinte formato:

- Na primeira linha, constam os caracteres **Q:** , seguidos de um espaço, e depois são especificados os nomes de cada estado da máquina, separados por espaços. Os nomes dos estados são compostos apenas de caracteres de **a** a **z**, **A** a **Z**, e de **0** a **9**, com no máximo 7 caracteres, em qualquer ordem, com exceção do estado inicial(I) e final(F) que são especificados nas próximas 2 linhas e nesta ordem.
- Nas linhas seguintes, é definida a função de transição, sendo que em cada linha uma ou mais transições são definidas. Cada transição é composta pelo nome de dois estados separados por espaços e por uma seta à direita (**->**), depois uma barra vertical (**|**), e em seguida uma lista de caracteres, separados por espaço, que são os caracteres de entrada sobre os quais aquela transição ocorre.
- O fim da definição da função de transição ocorre numa linha contendo apenas **---**, encerrando assim a definição também da máquina. Fim do arquivo.

- As transições deverão ser utilizadas pelo usuário no momento da execução, mostrando para qual estado ela levou, e as possíveis mudanças na mistura provocadas pela adição de um novo ingrediente. Mais detalhes na seção de “Saída”.

Exemplo

A tabela abaixo demonstra um exemplo de como podem ser documentados os ingredientes, receitas e restrições. As imagens abaixo não estão completas para fins de simplificação.

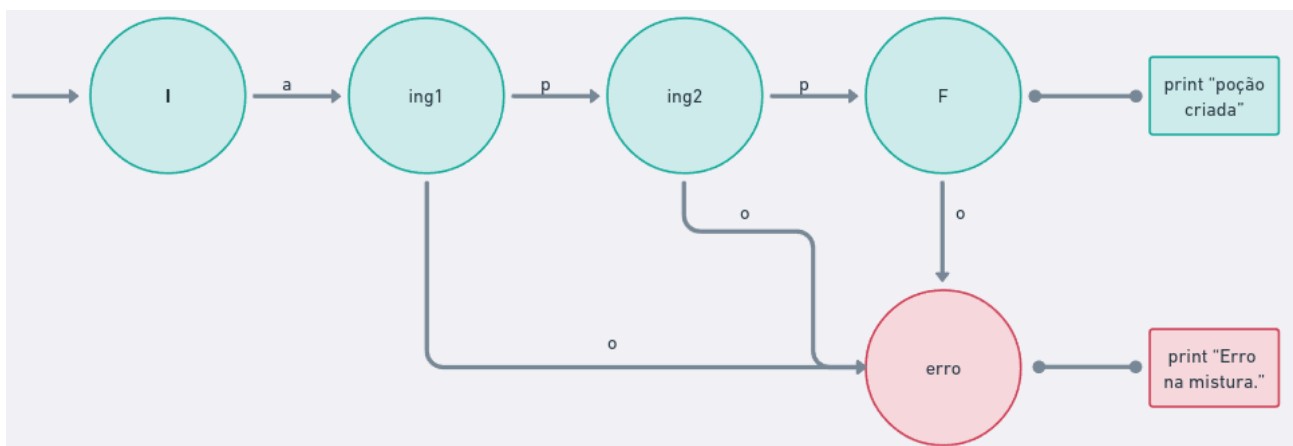
Ingrediente	Símbolo	Propriedades
água	a	—
pétalas	p	—
óleo	o	—

Restrições:

- Água e óleo não podem estar na mesma receita.

Receita:

- Poção de restauração comum: água e duas pétalas.



Exemplo de Entrada

Especificação do AFD da imagem anterior no formato exigido.

Arquivo:

```

Q: I ing1 ing2 ing3 erro F
I: I
F: F
I -> ing1      | a
ing1 -> ing2   | p
ing1 -> erro   | o

```

ing2 -> F		p
ing2 -> erro		o
F -> erro		o

Saída

A primeira saída deve ser do programa perguntando qual ingrediente será inserido na máquina:

Insira o símbolo do primeiro ingrediente da receita:

Após a inserção do primeiro ingrediente o programa deve perguntar se o usuário deseja inserir mais um ingrediente:

Deseja inserir mais um ingrediente(s/n)?

s

Qual ingrediente será inserido:

Caso o usuário opte por não inserir mais ingredientes e sequência de símbolos inseridos esteja de acordo com uma receita, a saída deve informar que uma poção foi criada.

Deseja inserir mais um ingrediente(s/n)?

n

Poção de restauração comum criada

Caso a sequência de caracteres não esteja de acordo com uma receita, a saída deve informar que houve um erro.

Deseja inserir mais um ingrediente(s/n)?

n

Erro na mistura

Observações:

- 1) Note que o usuário pode exceder o número de ingredientes aceitáveis das máquinas, logo o programa não impede uma inserção contínua de símbolos, o importante é reconhecer a entrada.
- 2) As exibições não precisam ser feitas como no exemplo, fica a critério do grupo a implementação das interações, desde que fique devidamente explícito o que deve ser feito a seguir;

Autômato de pilha

No caso do autômato de pilha a inserção de ingredientes deve provocar algum tipo de reação na mistura (acidez, coloração, mudança de temperatura, etc). Esta reação deve ser inserida na pilha e devidamente retirada com uma nova entrada. Ficam a critério do grupo as reações ou outras ideias que quiserem implementar. Vale ressaltar que para este autômato o formato da entrada fica a critério dos alunos, mas façam a definição de maneira semelhante à da entrada do AFD.

Extras

Turing, Moore e Mealy são opcionais e a temática é livre.

Estas máquinas poderão ser exploradas como os grupos quiserem, tendo em mente seu poder computacional relativo às máquinas obrigatórias.

Observações Gerais:

- Não é necessário utilizar sempre poções na parte extra. Vocês podem criar outro contexto em que são utilizados pratos de comida ou Megazords, por exemplo. Sendo assim, fica a seu critério como utilizar as máquinas para gerar algo ou outro contexto devidamente documentado.
- Faça a documentação bem organizada e com figuras ilustrando o que está acontecendo em suas máquinas.
- Uma interface bem feita será levada em consideração na correção final.
- Na apresentação é importante mostrar a execução de um exemplo em detalhes, discutindo cada ponto da implementação.

Dicas para montagem de interfaces pela linha de comando:

- <https://ascii.today/> - Digite e escolha um formato de título.
- <https://ozh.github.io/ascii-tables/> - Formatação de menus (várias formas em output style).