

# VISUALIZAÇÃO E ANIMAÇÃO DE AUTÓMATOS EM OCSIGEN FRAMEWORK

**RITA MACEDO**

**ARTUR MIGUEL DIAS**

**ANTÓNIO RAVARA**



# Introdução

- Linguagens Formais e Teoria de Autómatos:
  - Tema base nas Ciências da Computação
  - Natureza abstrata constitui uma dificuldade para muitos estudantes
  - Ferramentas interativas são importantes na aprendizagem
- Em disciplinas de Teoria da Computação:
  - Falta suporte de ferramentas e/ou programação
  - Bibliografia em inglês (material de aula pode ser em português)

# Objetivos

- Ferramenta interativa, disponível em browsers (para web) que permita:
  - Execução animada e gráfica dos algoritmos
  - Criação de exercícios pelos alunos
  - Avaliação automática e com feedback de exercícios
- Requisitos:
  - Plataforma extensível e genérica;
- Desenvolvida com uma ferramenta robusta, extensível e que permita o uso do mínimo de linguagens;
  - Prova de conceito: desenvolvimento só em OCaml

# Contribuições

- Primeira versão da ferramenta ainda em desenvolvimento
- Esta ferramenta pode ser acedida em <http://ctp.di.fct.unl.pt/FACTOR/OFLAT>

## Autómatos Animados

### **Ações**

Carregar Autómatos

Gerar Autómato

Testar aceitação de palavra

Avaliar natureza dos estados

# Contribuições

- Carregar Autómatos:

## Autómatos Animados

### Ações

#### Carregar Autómatos

Exemplos de AFD

[Example 1](#)

[Example 2](#)

Exemplos de AFN

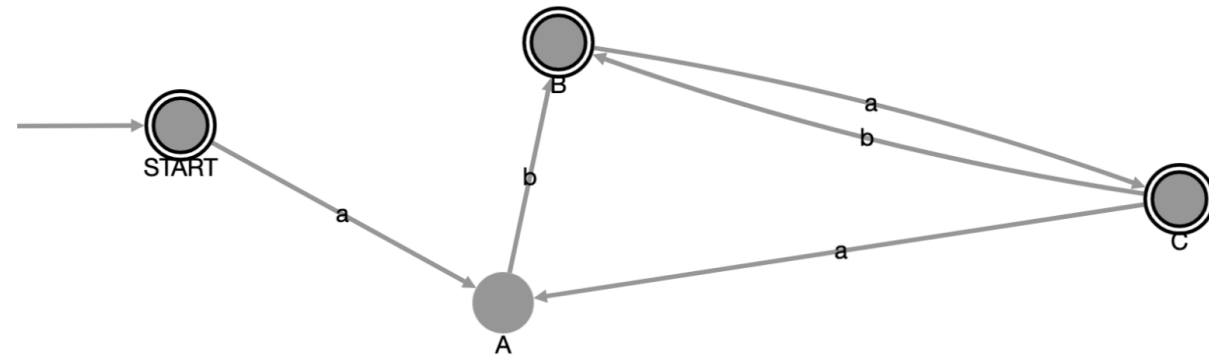
[Example 1](#)

[Example 2](#)

#### Gerar Autómatato

#### Testar aceitação de palavra

#### Avaliar natureza dos estados



# Contribuições

- Gerar Autómatos:

### Autómatos Animados

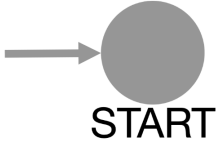
**Ações**  
**Carregar Autómatos**  
**Gerar Autómatos**

Adicionar estado inicial Adicionar estado inicial como final

Nome:  Adicionar estado Adicionar estado final

Estado de partida:  Estado de chegada:   
Transição:  Adicionar Transição

**Testar aceitação de palavra**  
**Avaliar natureza dos estados**



### Autómatos Animados


**Ações**  
**Carregar Autómatos**  
**Gerar Autómatos**

Adicionar estado inicial Adicionar estado inicial como final

Nome: B  Adicionar estado Adicionar estado final

Estado de partida:  Estado de chegada:   
Transição:  Adicionar Transição

**Testar aceitação de palavra**  
**Avaliar natureza dos estados**



### Autómatos Animados

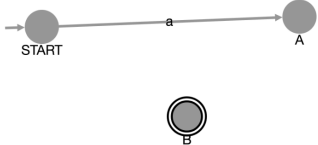
**Ações**  
**Carregar Autómatos**  
**Gerar Autómatos**

Adicionar estado inicial Adicionar estado inicial como final

Nome:  Adicionar estado Adicionar estado final

Estado de partida: START  Estado de chegada:   
A  Transição: a  Adicionar Transição

**Testar aceitação de palavra**  
**Avaliar natureza dos estados**



# Contribuições

- Testar aceitação de palavra:

## Autómatos Animados

### Ações

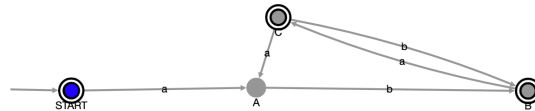
**Carregar Autómatos**

**Gerar Autómató**

**Testar aceitação de palavra**

Palavra: aba

**Avaliar natureza dos estados**



## Autómatos Animados

### Ações

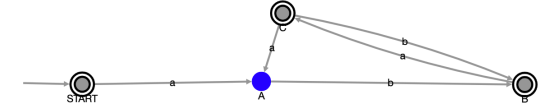
**Carregar Autómatos**

**Gerar Autómató**

**Testar aceitação de palavra**

Palavra: ba

**Avaliar natureza dos estados**



## Autómatos Animados

### Ações

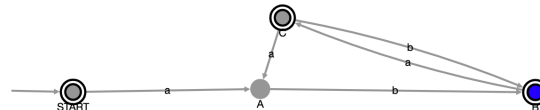
**Carregar Autómatos**

**Gerar Autómató**

**Testar aceitação de palavra**

Palavra: a

**Avaliar natureza dos estados**



## Autómatos Animados

### Ações

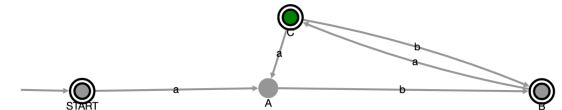
**Carregar Autómatos**

**Gerar Autómató**

**Testar aceitação de palavra**

Palavra:

**Avaliar natureza dos estados**



# Contribuições

- Avaliar natureza dos estados:

## Autómatos Animados

### Ações

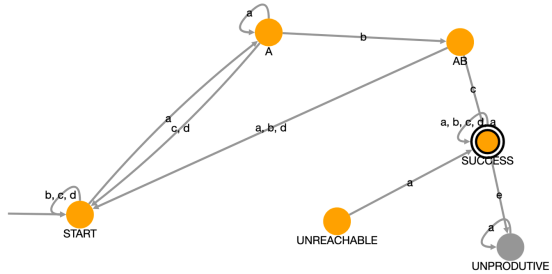
Carregar Autómatos

Gerar Autómatos

Testar aceitação de palavra

Avaliar natureza dos estados

Estados Produtivos   Estados Acessíveis   Estados Úteis



## Autómatos Animados

### Ações

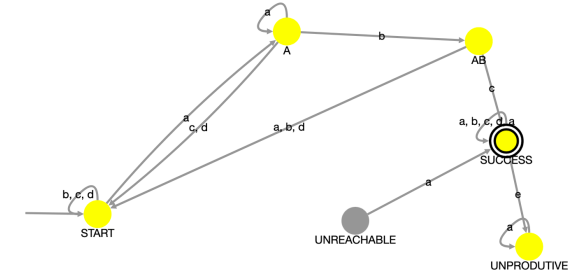
Carregar Autómatos

Gerar Autómatos

Testar aceitação de palavra

Avaliar natureza dos estados

Estados Produtivos   Estados Acessíveis   Estados Úteis



## Autómatos Animados

### Ações

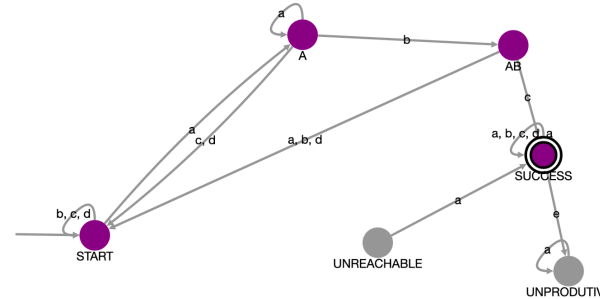
Carregar Autómatos

Gerar Autómatos

Testar aceitação de palavra

Avaliar natureza dos estados

Estados Produtivos   Estados Acessíveis   Estados Úteis





# Ocsigen Framework

- Pré-requisito: utilizar apenas (ou majoritariamente) OCaml para realizar a prova de conceito.
- Framework programável em OCaml para criação de ferramentas web interativas
  - Programação funcional como uma solução elegante e robusta para problemas de interação nas páginas web
  - Código cliente e servidor como um só programa
  - Compila o código OCaml do cliente para JavaScript
    - Trabalho conjunto com JavaScript.
    - Utilização de bibliotecas JavaScript
- Composto por vários componentes:
  - Eliom
  - Js\_of\_ocaml
  - Lwt
  - Tyxml
  - Ocsigen-start
  - Ocsigen-Toolkit

# Cytoscape.js

- Biblioteca de teoria de grafos escrita em JavaScript
- Permite facilmente mostrar e manipular grafos
- Contém funções para análise de grafos
- Utilização tanto em browsers de desktop como em browsers de sistemas móveis
- Intuitiva, fácil de usar e muito completa
- API muito desenvolvida, bem explicada e com exemplos
- Disponibiliza exemplos demo que o utilizador pode usar como base para os seus projetos
- Não é OCaml mas a integração como Ocsigen é simples
  - Ocsigen liga com qualquer biblioteca JavaScript.

# Estado do Estado da Arte

- Há muitas ferramentas:
  - Com os seus próprios princípios e objetivos
  - Umas mais completas que outras
  - Umas de mais fácil acesso que outras
- Implementações não verificadas dos algoritmos
- Nem sempre a utilização/visualização é intuitiva
- Poucas permitem um sistema de avaliação de exercícios
- A maioria das aplicações são em inglês;
- Lugar para uma ferramenta interativa em português
  - Para disponibilizar nos PALOP, onde houver menor domínio de inglês

# Estado da Arte: Ferramentas FLAT

- Muitas ferramentas para o estudo de temas relacionados com Teoria da Computação desenvolvidas desde a década de 60
- Surgem também diferentes tipos de bibliotecas
- Todas diferentes e com os seus próprios princípios e objetivos
- Das ferramentas de Teoria da Computação destacam-se:
  - Automaton Simulator
  - FSM Simulator, Regular Expressions Gym, FSM2Regex
  - JFlap

# Estado da Arte: Ferramentas de Avaliação Automática

- Contexto OCaml: Projeto Learn OCaml
  - Contém exercícios de aprendizagem de OCaml
  - Oferece uma infraestrutura sofisticada com diferentes tipos de utilizadores
  - Permite aos professores desenvolver exercícios de aula e de avaliação
- Dois exemplos de ferramentas que já integram FLAT com um sistema de avaliação são:
  - AutoMate
  - Automata Tutor

# Trabalho Futuro

- No Futuro pretende-se:
  - Investigar e pôr em prática outras funcionalidades do *framework*, de forma a tirar partido das suas potencialidades.
  - Incluir mais funcionalidades: Expressões Regulares, linguagens não regulares, linguagens LL, linguagens independentes de contexto, Autómatos de pilha e Máquinas de Turing
  - integrar com suporte à avaliação, nomeadamente permitindo a submissão e classificação de exercícios.

# Obrigada!

# VISUALIZAÇÃO E ANIMAÇÃO DE AUTÓMATOS EM OCSIGEN FRAMEWORK

**RITA MACEDO**

**ANTÓNIO RAVARA**

**ARTUR MIGUEL DIAS**



**NOVALINCS**  
LABORATORY FOR COMPUTER  
SCIENCE AND INFORMATICS



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE NOVA DE LISBOA



UNIVERSIDADE  
BEIRA INTERIOR