NFA2DFA

Generated by Doxygen 1.8.13

Contents

1 NFA2DFA 1

2 Class Index 3

2.1 Class List . . . 3

3 Class Documentation 5

3.1 Dfa Class Reference . . . 5

3.1.1 Detailed Description . . . 5

3.1.2 Member Function Documentation . . . 6

3.1.2.1 AddState() . . . 6

3.1.2.2 drawDFA() . . . 6

3.1.2.3 setAlphabet() . . . 7

3.2 Nfa Class Reference . . . 7

3.2.1 Detailed Description . . . 8

3.2.2 Constructor & Destructor Documentation . . . 8

3.2.2.1 Nfa() . . . 8

3.2.3 Member Function Documentation . . . 9

3.2.3.1 EClosure() . . . 9

3.2.3.2 FindPos() . . . 9

3.2.3.3 Move() . . . 10

3.2.3.4 SubSets() . . . 10

3.3 State Class Reference . . . 10

3.3.1 Detailed Description . . . 11

3.3.2 Constructor & Destructor Documentation . . . 12

3.3.2.1 State() . . . 12

3.3.3 Member Function Documentation . . . 12

3.3.3.1 Delta() . . . 12

3.3.3.2 getMark() . . . 13

3.3.3.3 getStr() . . . 13

3.3.3.4 Insert() . . . 13

3.3.3.5 operator<() . . . 13

3.3.3.6 operator==() . . . 14

3.3.3.7 setMark() . . . 14

3.3.3.8 setStr() . . . 14

Index 17

Chapter 1

NFA2DFA

Author

Alejandro Peraza González

Date

05/11/2019

Universidad

Universidad de La Laguna

Curso

2º de Ingeniería Informática

Título

La construccion de subconjuntos

Correo

alu0101211770@ull.edu.es

Referencias

https://campusvirtual.ull.es/1920/pluginfile.php/181073/mod_assign/introattachment/0/CYA_1920_Practica_7.pdf?forcedownload=1

Historial de revisiones

30/10/2019 - Creación (primera versión) del código

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Dfa
: Clase en la que se almacena el dfa equivalente al nfa dado 5

Nfa
: Se crea un NFA a partir de los datos leídos en el fichero de entrada y se imprime un DFA equivalente en el de salida 7

State
: Clase empleada para representar cada estado del DFA 10

Chapter 3

Class Documentation

3.1 Dfa Class Reference

Clase en la que se almacena el dfa equivalente al nfa dado.

`#include <Dfa.h>`

Public Member Functions

- Dfa ()

  Constructor por defecto.
- ~Dfa ()

  Destructor de la clase Autómata que libera la memoria reservada por los vectores states_, final_states_ y alphabet_.
- void setAlphabet (std::vector< char > alphabet)

  Setter del atributo alphabet_.
- void AddState (State q, int mode)

  Método para incluir los estados, tanto los genéricos(mode = 0) como el inicial (mode = 1) y los de aceptación (mode = 2)
- std::ostream & drawDFA (std::ostream &os)

  Método que realiza la conversión de los datos del DFA a un archivo con formato Dot.

3.1.1 Detailed Description

Clase en la que se almacena el dfa equivalente al nfa dado.

Archivo Dfa.h: En esta clase se encuentra el método addState que sirve para rellenar los vectores de estados states_ y final_states así como el estado inicial. Por otro lado, con la función drawDfa se imprimen los datos en un archivo de salida con formato dot.

Author

Alejandro Peraza González

Date

05/11/2019

Universidad

Universidad de La Laguna

Curso

2º de Ingeniería Informática

Título

La construccion de subconjuntos

Correo

alu0101211770@ull.edu.es

Referencias

https://campusvirtual.ull.es/1920/pluginfile.php/181073/mod_assign/introattachment/0/CYA_1920_Practica_7.pdf?forcedownload=1

Historial de revisiones

30/10/2019 - Creación (primera versión) del código

3.1.2 Member Function Documentation

3.1.2.1 AddState()

```
void Dfa::AddState (
            State q,
            int mode )
```

Método para incluir los estados, tanto los genéricos(mode = 0) como el inicial (mode = 1) y los de aceptación (mode = 2)

Parameters

q	- el estado a almacenar
mode	- selecciona dónde se almacenará q

3.1.2.2 drawDFA()

```
std::ostream & Dfa::drawDFA (
            std::ostream & os )
```

Método que realiza la conversión de los datos del DFA a un archivo con formato Dot.

Parameters

os	- fichero en el que se imprime la conversión por referencia

Returns

se devuelve por referncia el stream

3.1.2.3 setAlphabet()

```
void Dfa::setAlphabet (
            std::vector< char > alphabet )
```

Setter del atributo alphabet_.

Parameters

alphabet	- un vector de caracteres que conforma el alfabeto

The documentation for this class was generated from the following files:

- Dfa.h
- Dfa.cc

3.2 Nfa Class Reference

Se crea un NFA a partir de los datos leídos en el fichero de entrada y se imprime un DFA equivalente en el de salida.

```
#include <Nfa.h>
```

Public Member Functions

- Nfa (std::istream &is)

  Constructor de la clase Autómata que lee el contenido de un fichero y crea un DFA.
- ~Nfa ()

  Destructor de la clase Autómata que libera la memoria reservada por los vectores states_, final_states_ y alphabet_.
- int FindPos (std::string str, std::vector< State > v)

  Método que recorre el vecor states_ que contiene los estados y devuelve la posición en la que encuentra la string buscada.
- std::set< State > EClosure (std::set< State > T)

  Método que obtiene los E-clausra estados de un estado.
- std::set< State > Move (std::set< State > S, char token)

Método que devuelve el conjunto de estados que pueden alcanzarse desde el estado S con el símbolo de entrada token.

- std::ostream & SubSets (Dfa &DFA, std::ostream &os)

  Método que aplica el algoritmo de construcción de suconjuntos.
- bool **isinVector** (std::vector< std::pair< std::set< State >, State >> &s, std::set< State > &q, int &pos)
- int **marked** (std::vector< std::pair< std::set< State >, State >> &s, int &a)

3.2.1 Detailed Description

Se crea un NFA a partir de los datos leídos en el fichero de entrada y se imprime un DFA equivalente en el de salida.

Archivo Nfa.h: Esta es la clase principal y la que inicializa la lectura y creación del Nfa, requiere de la creación de los estados, un alfabeto de entrada y las transiciones entre estados. A partir de este Nfa se creará un DFA equivalente.

Author

Alejandro Peraza González

Date

05/11/2019

Universidad

Universidad de La Laguna

Curso

2º de Ingeniería Informática

Título

La construccion de subconjuntos

Correo

alu0101211770@ull.edu.es

Referencias

https://campusvirtual.ull.es/1920/pluginfile.php/181073/mod_assign/introattachment/0/CYA_1920_Practica_7.pdf?forcedownload=1

Historial de revisiones

30/10/2019 - Creación (primera versión) del código

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Nfa()

```
Nfa::Nfa (
            std::istream & is )
```

Constructor de la clase Autómata que lee el contenido de un fichero y crea un DFA.

Parameters

is	- fichero del que se lee los datos del DFA por referencia

3.2.3 Member Function Documentation

3.2.3.1 EClosure()

```
std::set< State > Nfa::EClosure (
            std::set< State > T )
```

Método que obtiene los E-clausra estados de un estado.

Parameters

T	- conjunto de estados sobre el que calcular

Returns

el conjunto de estados resultante

3.2.3.2 FindPos()

```
int Nfa::FindPos (
            std::string str,
            std::vector< State > v )
```

Método que recorre el vecor states_ que contiene los estados y devuelve la posición en la que encuentra la string buscada.

Parameters

str	- string para poder encontrar la posición en el vector

Returns

entero que indica la posición del vector

3.2.3.3 Move()

```
std::set< State > Nfa::Move (
            std::set< State > S,
            char token )
```

Método que devuelve el conjunto de estados que pueden alcanzarse desde el estado S con el símbolo de entrada token.

Parameters

S	- estado desde el que se analizan las tranisiciones
token	- símbolo de entrada

Returns

conjunto de estados correspondiente

3.2.3.4 SubSets()

```
std::ostream & Nfa::SubSets (
            Dfa & DFA,
            std::ostream & os )
```

Método que aplica el algoritmo de construcción de suconjuntos.

Parameters

DFA_states	- el conjunto de estados del DFA correspondiente

The documentation for this class was generated from the following files:

- Nfa.h
- Nfa.cc

3.3 State Class Reference

Clase empleada para representar cada estado del DFA.

```
#include <State.h>
```

Public Types

- typedef std::vector< std::pair< State, char > > **vector_pair**

Public Member Functions

- State ()

  Constructor de la clase State por defecto.
- State (std::string state)

  Constructor al recibir una string, esta es asignada a la variable state_.
- ~State ()

  Destructor que libera la memoria reservada por el vector transitions_.
- std::string getStr () const

  Getter del atributo state_.
- void setStr (std::string &str)

  Setter del atributo state_.
- bool getMark () const

  Método que devuelve el atributo marked_.
- void setMark (bool mark)

  Método que mopdiifca el atributo marked_.
- vector_pair **getTransitions** () const
- State Delta (char token)

  Método que devuelve el estado siguiente dado un estado actual y un char.
- void Insert (char token, State q)

  Método para comparar si dos estados son iguales.
- bool operator< (const State &other) const

  Sobrecarga del operador < para el correcto funcionamiento de un set.
- bool operator== (const State &other) const

  Compara si dos estados son iguales.

3.3.1 Detailed Description

Clase empleada para representar cada estado del DFA.

Archivo State.h: Clase conformada por un string y un booleano que sirve para saber si un estado está o no marcado, también tiene un vector de pares, dichos pares son un Estado (un objeto de la misma clase) y un char. Dicho vector indica las diversas transiciones de cada estado.

Author

Alejandro Peraza González

Date

05/11/2019

Universidad

Universidad de La Laguna

Curso

2º de Ingeniería Informática

Título

La construccion de subconjuntos

Correo

alu0101211770@ull.edu.es

Referencias

https://campusvirtual.ull.es/1920/pluginfile.php/181073/mod_assign/introattachment/0/CYA_1920_Practica_7.pdf?forcedownload=1

Historial de revisiones

30/10/2019 - Creación (primera versión) del código

3.3.2 Constructor & Destructor Documentation

3.3.2.1 State()

```
State::State (
            std::string state )
```

Constructor al recibir una string, esta es asignada a la variable state_.

Parameters

state	- identificador de cada estado

3.3.3 Member Function Documentation

3.3.3.1 Delta()

```
State State::Delta (
            char token )
```

Método que devuelve el estado siguiente dado un estado actual y un char.

Parameters

token	- un char

Returns

un estado siguiente

3.3.3.2 getMark()

```
bool State::getMark ( ) const
```

Método que devuelve el atributo marked_.

Returns

booleano que indica si el estado está o no marcado

3.3.3.3 getStr()

```
std::string State::getStr ( ) const
```

Getter del atributo state_.

Returns

string que identifica a cada estado

3.3.3.4 Insert()

```
void State::Insert (
            char token,
            State q )
```

Método para comparar si dos estados son iguales.

Parameters

state	- estado a comparar

Returns

un booleano que indica si son o no iguales

3.3.3.5 operator<()

```
bool State::operator< (
            const State & other ) const
```

Sobrecarga del operador < para el correcto funcionamiento de un set.

Parameters

other	- un estado, objeto de la clase

Returns

un booleano que indica si es menor o no

3.3.3.6 operator==()

```
bool State::operator== (
            const State & other ) const
```

Compara si dos estados son iguales.

Parameters

state	- estado a

Returns

un booleano que indica si son o no iguales

3.3.3.7 setMark()

```
void State::setMark (
            bool mark )
```

Método que mopdiifca el atributo marked_.

Parameters

mark	- 1 ó 0 para marcar o desmarcar el estado

3.3.3.8 setStr()

```
void State::setStr (
            std::string & str )
```

Setter del atributo state_.

Parameters

str	- una string que modifica el valor de state_

The documentation for this class was generated from the following files:

- State.h
- State.cc

Index

AddState
    Dfa, 6

Delta
    State, 12
Dfa, 5
    AddState, 6
    drawDFA, 6
    setAlphabet, 7
drawDFA
    Dfa, 6

EClosure
    Nfa, 9

FindPos
    Nfa, 9

getMark
    State, 12
getStr
    State, 13

Insert
    State, 13

Move
    Nfa, 9

Nfa, 7
    EClosure, 9
    FindPos, 9
    Move, 9
    Nfa, 8
    SubSets, 10

operator<
    State, 13
operator==
    State, 14

setAlphabet
    Dfa, 7
setMark
    State, 14
setStr
    State, 14
State, 10
    Delta, 12
    getMark, 12
    getStr, 13
    Insert, 13
    operator<, 13
    operator==, 14
    setMark, 14
    setStr, 14
    State, 12
SubSets
    Nfa, 10