

My Project

Generated by Doxygen 1.9.8

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 Action Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Function Documentation	5
3.1.2.1 Get_Move()	5
3.1.2.2 Get_New_State()	6
3.1.2.3 Get_New_Word()	6
3.1.2.4 Set_Move()	6
3.1.2.5 Set_New_State()	6
3.1.2.6 Set_New_Word()	6
3.2 Head Class Reference	7
3.2.1 Detailed Description	7
3.2.2 Member Function Documentation	7
3.2.2.1 Get_Position()	7
3.2.2.2 Get_State()	7
3.2.2.3 Set_Position()	7
3.2.2.4 Set_State()	8
3.3 Rules Class Reference	8
3.3.1 Constructor & Destructor Documentation	8
3.3.1.1 ~Rules()	8
3.3.2 Member Function Documentation	9
3.3.2.1 Get_Amount_Of_States()	9
3.3.2.2 Get_Amount_Of_Words()	9
3.3.2.3 Get_Rule()	9
3.3.2.4 Get_State()	9
3.3.2.5 Get_Word()	10
3.3.2.6 New_Rule()	10
3.3.2.7 Remove_Rule()	10
3.3.2.8 View()	11
3.4 Tape Class Reference	11
3.4.1 Detailed Description	11
3.4.2 Member Function Documentation	11
3.4.2.1 Add_Cell() [1/2]	11
3.4.2.2 Add_Cell() [2/2]	12
3.4.2.3 Get_Cell()	12
3.4.2.4 Get_Size()	12
3.4.2.5 Remove_Cell() [1/2]	12

3.4.2.6 Remove_Cell() [2/2]	13
3.4.2.7 Set_Cell()	13
3.4.2.8 View()	13
4 File Documentation	15
4.1 Action.cpp File Reference	15
4.2 Action.hpp File Reference	15
4.3 Action.hpp	15
4.4 Execution.cpp File Reference	16
4.4.1 Macro Definition Documentation	16
4.4.1.1 LIMIT	16
4.4.2 Function Documentation	16
4.4.2.1 Turing_Mashine()	16
4.5 Execution.hpp File Reference	16
4.5.1 Function Documentation	17
4.5.1.1 Turing_Mashine()	17
4.6 Execution.hpp	17
4.7 Head.cpp File Reference	17
4.8 Head.hpp File Reference	17
4.9 Head.hpp	18
4.10 Interface.cpp File Reference	18
4.10.1 Function Documentation	18
4.10.1.1 Interface()	18
4.11 Interface.hpp File Reference	18
4.11.1 Macro Definition Documentation	19
4.11.1.1 ADD_CELL	19
4.11.1.2 EDIT_HEAD	19
4.11.1.3 EXIT	19
4.11.1.4 GO	19
4.11.1.5 HELP	19
4.11.1.6 NEW_RULE	20
4.11.1.7 REMOVE_CELL	20
4.11.1.8 REMOVE_RULE	20
4.11.1.9 SET_CELL	20
4.11.1.10 VIEW_RULES	20
4.11.1.11 VIEW_TAPE	20
4.11.2 Function Documentation	20
4.11.2.1 Interface()	20
4.12 Interface.hpp	21
4.13 Lab1.cpp File Reference	21
4.13.1 Function Documentation	21
4.13.1.1 main()	21

4.14 Loading_From_File.cpp File Reference	21
4.14.1 Function Documentation	22
4.14.1.1 Edit_Head()	22
4.14.1.2 Get_Word()	22
4.14.1.3 Loading_From_File()	22
4.14.1.4 New_Rule()	23
4.15 Loading_From_File.hpp File Reference	23
4.15.1 Function Documentation	23
4.15.1.1 Edit_Head()	23
4.15.1.2 Get_Word()	24
4.15.1.3 Loading_From_File()	24
4.15.1.4 New_Rule()	24
4.16 Loading_From_File.hpp	24
4.17 Rules.cpp File Reference	25
4.18 Rules.hpp File Reference	25
4.18.1 Macro Definition Documentation	25
4.18.1.1 L	25
4.18.1.2 N	25
4.18.1.3 R	25
4.19 Rules.hpp	26
4.20 Tape.cpp File Reference	26
4.21 Tape.hpp File Reference	26
4.21.1 Macro Definition Documentation	26
4.21.1.1 VOID	26
4.22 Tape.hpp	27
Index	29

Chapter 1

Class Index

1.1 Class List

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

Action	< Class for storing actions	5
Head	< Class for implementing the machine head	7
Rules	8
Tape	Class for implementing the machine tape	11

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Action.cpp	15
Action.hpp	15
Execution.cpp	16
Execution.hpp	16
Head.cpp	17
Head.hpp	17
Interface.cpp	18
Interface.hpp	18
Lab1.cpp	21
Loading_From_File.cpp	21
Loading_From_File.hpp	23
Rules.cpp	25
Rules.hpp	25
Tape.cpp	26
Tape.hpp	26

Chapter 3

Class Documentation

3.1 Action Class Reference

< Class for storing actions

```
#include <Action.hpp>
```

Public Member Functions

- void [Set_New_State](#) (std::string newstate)
- void [Set_New_Word](#) (std::string newword)
- void [Set_Move](#) (char move)
- std::string [Get_New_State](#) () const
- std::string [Get_New_Word](#) () const
- char [Get_Move](#) () const

3.1.1 Detailed Description

< Class for storing actions

3.1.2 Member Function Documentation

3.1.2.1 [Get_Move\(\)](#)

```
char Action::Get_Move ( ) const
```

Method to get move

Returns

Move

3.1.2.2 Get_New_State()

```
std::string Action::Get_New_State ( ) const
```

Method to get new state

Returns

New state

3.1.2.3 Get_New_Word()

```
std::string Action::Get_New_Word ( ) const
```

Method to get new word

Returns

New word

3.1.2.4 Set_Move()

```
void Action::Set_Move (
    char move )
```

Method for setting a move

Parameters

in	<i>move</i>	Move
----	-------------	------

3.1.2.5 Set_New_State()

```
void Action::Set_New_State (
    std::string newstate )
```

Method for setting a new state

Parameters

in	<i>newstate</i>	New state
----	-----------------	-----------

3.1.2.6 Set_New_Word()

```
void Action::Set_New_Word (
    std::string newword )
```

Method for setting a new word

Parameters

in	<i>newword</i>	New word
----	----------------	----------

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

- [Action.hpp](#)
- [Action.cpp](#)

3.2 Head Class Reference

< Class for implementing the machine head

```
#include <Head.hpp>
```

Public Member Functions

- void [Set_Position](#) (int position)
- void [Set_State](#) (std::string state)
- int [Get_Position](#) () const
- std::string [Get_State](#) () const

3.2.1 Detailed Description

< Class for implementing the machine head

3.2.2 Member Function Documentation

3.2.2.1 Get_Position()

```
int Head::Get_Position ( ) const
```

Method to get head position

Returns

[Head](#) position

3.2.2.2 Get_State()

```
std::string Head::Get_State ( ) const
```

Method to get head state

Returns

[Head](#) state

3.2.2.3 Set_Position()

```
void Head::Set_Position (
    int position )
```

Method for setting the head position

Parameters

in	<i>position</i>	New position
----	-----------------	--------------

3.2.2.4 Set_State()

```
void Head::Set_State (
    std::string state )
```

Method for setting the head state

Parameters

in	<i>state</i>	New state
----	--------------	-----------

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

- [Head.hpp](#)
- [Head.cpp](#)

3.3 Rules Class Reference

```
#include <Rules.hpp>
```

Public Member Functions

- [~Rules](#) ()
- [std::string Get_State](#) (int position) const
- [std::string Get_Word](#) (int position) const
- [int Get_Amount_Of_States](#) () const
- [int Get_Amount_Of_Words](#) () const
- [Action * Get_Rule](#) (int state, int word) const
- [void New_Rule](#) (char move, std::string newword, std::string newstate, std::string word, std::string state)
- [bool Remove_Rule](#) (std::string word, std::string state)
- [std::string View](#) () const

3.3.1 Constructor & Destructor Documentation**3.3.1.1 ~Rules()**

```
Rules::~Rules ( )
```

Destructor for clearing the rules table

3.3.2 Member Function Documentation

3.3.2.1 Get_Amount_Of_States()

```
int Rules::Get_Amount_Of_States ( ) const
```

Method for getting the amount of different states

Returns

Amount of different states

3.3.2.2 Get_Amount_Of_Words()

```
int Rules::Get_Amount_Of_Words ( ) const
```

Method for getting the amount of different words

Returns

Amount of different states

3.3.2.3 Get_Rule()

```
Action * Rules::Get_Rule (
    int state,
    int word ) const
```

Method for obtaining actions based on specified conditions

Parameters

in	<i>state</i>	State number
in	<i>word</i>	Word number

Returns

Action

3.3.2.4 Get_State()

```
std::string Rules::Get_State (
    int position ) const
```

Method for obtaining the state of the number

Parameters

in	<i>position</i>	State number
----	-----------------	--------------

Returns

State

3.3.2.5 Get_Word()

```
std::string Rules::Get_Word (
    int position ) const
```

Method for obtaining the word of the number

Parameters

in	<i>position</i>	Word number
----	-----------------	-------------

Returns

Word

3.3.2.6 New_Rule()

```
void Rules::New_Rule (
    char move,
    std::string newword,
    std::string newstate,
    std::string word,
    std::string state )
```

Method for adding a new rule

Parameters

in	<i>move</i>	Symbol with move
in	<i>newword</i>	String with new word
in	<i>newstate</i>	String with new state
in	<i>word</i>	String with word
in	<i>state</i>	String with state

3.3.2.7 Remove_Rule()

```
bool Rules::Remove_Rule (
    std::string word,
    std::string state )
```


Method for deleting a rule

Parameters

in	<i>word</i>	Word of the rule to be deleted
in	<i>state</i>	State of the rule to be deleted

Returns

TRUE if such a rule exists else FALSE

3.3.2.8 View()

```
std::string Rules::View ( ) const
```

Method for viewing rules

Returns

[Rules](#) recorded in string

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

- [Rules.hpp](#)
- [Rules.cpp](#)

3.4 Tape Class Reference

Class for implementing the machine tape.

```
#include <Tape.hpp>
```

Public Member Functions

- void [Add_Cell](#) (std::string word)
- void [Add_Cell](#) (int position, std::string word)
- bool [Remove_Cell](#) ()
- bool [Remove_Cell](#) (int position)
- void [Set_Cell](#) (int position, std::string word)
- std::string [Get_Cell](#) (int position) const
- int [Get_Size](#) () const
- std::string [View](#) ([Head](#) head) const

3.4.1 Detailed Description

Class for implementing the machine tape.

3.4.2 Member Function Documentation

3.4.2.1 Add_Cell() [1/2]

```
void Tape::Add_Cell (
    int position,
    std::string word )
```

Method for adding a cell by position

Parameters

in	<i>position</i>	Cell position
in	<i>word</i>	Added cell

3.4.2.2 Add_Cell() [2/2]

```
void Tape::Add_Cell (
    std::string word )
```

Method for adding a cell to the end of a tape

Parameters

in	<i>word</i>	Added cell
----	-------------	------------

3.4.2.3 Get_Cell()

```
std::string Tape::Get_Cell (
    int position ) const
```

Method for receiving cell

Parameters

in	<i>position</i>	Cell position
----	-----------------	---------------

Returns

A cell

3.4.2.4 Get_Size()

```
int Tape::Get_Size ( ) const
```

Method for obtaining tape length

Returns

A tape lenght

3.4.2.5 Remove_Cell() [1/2]

```
bool Tape::Remove_Cell ( )
```

Method to delete the last cell

Returns

TRUE if such a cell exists else FALSE

3.4.2.6 Remove_Cell() [2/2]

```
bool Tape::Remove_Cell (
    int position )
```

Method to delete a cell by position

Parameters

in	<i>position</i>	Cell position
----	-----------------	---------------

Returns

TRUE if such a cell exists else FALSE

3.4.2.7 Set_Cell()

```
void Tape::Set_Cell (
    int position,
    std::string word )
```

Method for cell replacement

Parameters

in	<i>position</i>	Cell position
in	<i>word</i>	New cell

3.4.2.8 View()

```
std::string Tape::View (
    Head head ) const
```

Method for viewing tape with head position

Parameters

in	<i>head</i>	Machine head \retutn Tape recorded in string
----	-------------	--

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

- [Tape.hpp](#)
- [Tape.cpp](#)

Chapter 4

File Documentation

4.1 Action.cpp File Reference

```
#include "Action.hpp"
```

4.2 Action.hpp File Reference

```
#include <string>
```

Classes

- class [Action](#)
 < *Class for storing actions*

4.3 Action.hpp

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002 #include <string>
00004 class Action {
00005 public:
00010     void Set_New_State(std::string newstate);
00015     void Set_New_Word(std::string newword);
00020     void Set_Move(char move);
00025     std::string Get_New_State() const;
00030     std::string Get_New_Word() const;
00035     char Get_Move() const;
00036 private:
00037     std::string new_state;
00038     std::string new_word;
00039     char move;
00040 };
```

4.4 Execution.cpp File Reference

```
#include <windows.h>
#include "Execution.hpp"
```

Macros

- `#define` [LIMIT](#) 100

Functions

- `bool` [Turing_Mashine](#) ([Head](#) &head, [Tape](#) &tape, [Rules](#) &rules)

4.4.1 Macro Definition Documentation

4.4.1.1 LIMIT

```
#define LIMIT 100
```

4.4.2 Function Documentation

4.4.2.1 Turing_Mashine()

```
bool Turing_Mashine (
    Head & head,
    Tape & tape,
    Rules & rules )
```

Method for performing Turing machine simulation

Parameters

in, out	<i>head</i>	Mashine head
in, out	<i>tape</i>	Machine tape
in	<i>rules</i>	Machine rules

Returns

FALSE if the number of steps is more than 100 else TRUE

4.5 Execution.hpp File Reference

```
#include "Tape.hpp"
#include "Rules.hpp"
```

Functions

- bool [Turing_Mashine](#) ([Head](#) &head, [Tape](#) &tape, [Rules](#) &rules)

4.5.1 Function Documentation

4.5.1.1 Turing_Mashine()

```
bool Turing_Mashine (
    Head & head,
    Tape & tape,
    Rules & rules )
```

Method for performing Turing machine simulation

Parameters

in, out	<i>head</i>	Mashine head
in, out	<i>tape</i>	Machine tape
in	<i>rules</i>	Machine rules

Returns

FALSE if the number of steps is more than 100 else TRUE

4.6 Execution.hpp

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002 #include "Tape.hpp"
00003 #include "Rules.hpp"
00011 bool Turing\_Mashine(Head& head, Tape& tape, Rules& rules);
```

4.7 Head.cpp File Reference

```
#include "Head.hpp"
```

4.8 Head.hpp File Reference

```
#include <iostream>
```

Classes

- class [Head](#)
< Class for implementing the machine head

4.9 Head.hpp

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002 #include <iostream>
00004 class Head {
00005 public:
00010     void Set_Position(int position);
00015     void Set_State(std::string state);
00020     int Get_Position() const;
00025     std::string Get_State() const;
00026 private:
00027     int position;
00028     std::string state;
00029 };
```

4.10 Interface.cpp File Reference

```
#include "Interface.hpp"
```

Functions

- bool [Interface](#) ([Head](#) &head, [Tape](#) &tape, [Rules](#) &rules, std::string choice)

4.10.1 Function Documentation

4.10.1.1 Interface()

```
bool Interface (
    Head & head,
    Tape & tape,
    Rules & rules,
    std::string choice )
```

Method for interaction between user and program

Parameters

in, out	<i>head</i>	Mashine head
in, out	<i>tape</i>	Machine tape
in, out	<i>rules</i>	Machine rules
in	<i>choice</i>	Action choice string

Returns

FALSE if such action do not exist, TRUE if [Action](#) choice string is Exit

4.11 Interface.hpp File Reference

```
#include "Loading_From_File.hpp"
```



```
#include "Execution.hpp"
```

Macros

- `#define ADD_CELL "AddCell"`
- `#define GO "Go"`
- `#define SET_CELL "SetCell"`
- `#define REMOVE_CELL "RemoveCell"`
- `#define NEW_RULE "NewRule"`
- `#define REMOVE_RULE "RemoveRule"`
- `#define EDIT_HEAD "EditHead"`
- `#define VIEW_RULES "ViewRules"`
- `#define VIEW_TAPE "ViewTape"`
- `#define HELP "Help"`
- `#define EXIT "Exit"`

Functions

- `bool Interface (Head &head, Tape &tape, Rules &rules, std::string choice)`

4.11.1 Macro Definition Documentation

4.11.1.1 ADD_CELL

```
#define ADD_CELL "AddCell"
```

4.11.1.2 EDIT_HEAD

```
#define EDIT_HEAD "EditHead"
```

4.11.1.3 EXIT

```
#define EXIT "Exit"
```

4.11.1.4 GO

```
#define GO "Go"
```

4.11.1.5 HELP

```
#define HELP "Help"
```

4.11.1.6 NEW_RULE

```
#define NEW_RULE "NewRule"
```

4.11.1.7 REMOVE_CELL

```
#define REMOVE_CELL "RemoveCell"
```

4.11.1.8 REMOVE_RULE

```
#define REMOVE_RULE "RemoveRule"
```

4.11.1.9 SET_CELL

```
#define SET_CELL "SetCell"
```

4.11.1.10 VIEW_RULES

```
#define VIEW_RULES "ViewRules"
```

4.11.1.11 VIEW_TAPE

```
#define VIEW_TAPE "ViewTape"
```

4.11.2 Function Documentation

4.11.2.1 Interface()

```
bool Interface (
    Head & head,
    Tape & tape,
    Rules & rules,
    std::string choice )
```

Method for interaction between user and program

Parameters

in, out	<i>head</i>	Mashine head
in, out	<i>tape</i>	Machine tape
in, out	<i>rules</i>	Machine rules
in	<i>choice</i>	Action choice string

Returns

FALSE if such action do not exist, TRUE if [Action](#) choice string is Exit

4.12 Interface.hpp

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002 #include "Loading_From_File.hpp"
00003 #include "Execution.hpp"
00004 #define ADD_CELL "AddCell"
00005 #define GO "Go"
00006 #define SET_CELL "SetCell"
00007 #define REMOVE_CELL "RemoveCell"
00008 #define NEW_RULE "NewRule"
00009 #define REMOVE_RULE "RemoveRule"
00010 #define EDIT_HEAD "EditHead"
00011 #define VIEW_RULES "ViewRules"
00012 #define VIEW_TAPE "ViewTape"
00013 #define HELP "Help"
00014 #define EXIT "Exit"
00015
00025 bool Interface(Head& head, Tape& tape, Rules& rules, std::string choice);
```

4.13 Lab1.cpp File Reference

```
#include "Interface.hpp"
```

Functions

- int [main](#) ()

4.13.1 Function Documentation

4.13.1.1 main()

```
int main ( )
```

4.14 Loading_From_File.cpp File Reference

```
#include "Loading_From_File.hpp"
```

Functions

- bool [Edit_Head](#) (std::string str, [Head](#) &head, [Tape](#) &tape)
- bool [New_Rule](#) (std::string str, [Rules](#) &rules)
- std::string [Get_Word](#) (std::string str, int &position)
- bool [Loading_From_File](#) (std::string file_name, [Head](#) &head, [Tape](#) &tape, [Rules](#) &rules)

4.14.1 Function Documentation

4.14.1.1 Edit_Head()

```
bool Edit_Head (
    std::string str,
    Head & head,
    Tape & tape )
```

Method for setting the properties of the head, the value of which is taken from the string

Parameters

in	<i>str</i>	A string containing the properties of the machine head
out	<i>head</i>	Machine head
out	<i>tape</i>	Machine tape \retutn FALSE if there was an error reading from the string else TRUE

4.14.1.2 Get_Word()

```
std::string Get_Word (
    std::string str,
    int & position )
```

Helper method to get a word from a string

Parameters

in	<i>str</i>	A string
in	<i>position</i>	Position of the first letter of a word \retutn Received word

4.14.1.3 Loading_From_File()

```
bool Loading_From_File (
    std::string file,
    Head & head,
    Tape & tape,
    Rules & rules )
```

Method for reading car data from a file

Parameters

in	<i>file</i>	File name
out	<i>head</i>	Machine head
out	<i>tape</i>	Machine tape
out	<i>rules</i>	Machine rules \retutn TRUE if this file exist else FALSE

4.14.1.4 New_Rule()

```
bool New_Rule (
    std::string str,
    Rules & rules )
```

Method for adding a new rule from a string

Parameters

in	<i>str</i>	String with rules
out	<i>rules</i>	Machine rules \retutn FALSE if there was an error reading from the string else TRUE

4.15 Loading_From_File.hpp File Reference

```
#include <fstream>
#include "Tape.hpp"
#include "Rules.hpp"
```

Functions

- bool [Edit_Head](#) (std::string str, [Head](#) &, [Tape](#) &)
- bool [New_Rule](#) (std::string str, [Rules](#) &rules)
- std::string [Get_Word](#) (std::string str, int &position)
- bool [Loading_From_File](#) (std::string file, [Head](#) &head, [Tape](#) &tape, [Rules](#) &rules)

4.15.1 Function Documentation

4.15.1.1 Edit_Head()

```
bool Edit_Head (
    std::string str,
    Head & head,
    Tape & tape )
```

Method for setting the properties of the head, the value of which is taken from the string

Parameters

in	<i>str</i>	A string containing the properties of the machine head
out	<i>head</i>	Machine head
out	<i>tape</i>	Machine tape \retutn FALSE if there was an error reading from the string else TRUE

4.15.1.2 Get_Word()

```
std::string Get_Word (
    std::string str,
    int & position )
```

Helper method to get a word from a string

Parameters

in	<i>str</i>	A string
in	<i>position</i>	Position of the first letter of a word \retutn Received word

4.15.1.3 Loading_From_File()

```
bool Loading_From_File (
    std::string file,
    Head & head,
    Tape & tape,
    Rules & rules )
```

Method for reading car data from a file

Parameters

in	<i>file</i>	File name
out	<i>head</i>	Machine head
out	<i>tape</i>	Machine tape
out	<i>rules</i>	Machine rules \retutn TRUE if this file exist else FALSE

4.15.1.4 New_Rule()

```
bool New_Rule (
    std::string str,
    Rules & rules )
```

Method for adding a new rule from a string

Parameters

in	<i>str</i>	String with rules
out	<i>rules</i>	Machine rules \retutn FALSE if there was an error reading from the string else TRUE

4.16 Loading_From_File.hpp

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002 #include <fstream>
00003 #include "Tape.hpp"
00004 #include "Rules.hpp"
00012 bool Edit_Head(std::string str, Head&, Tape&);
00019 bool New_Rule(std::string str, Rules& rules);
00026 std::string Get_Word(std::string str, int& position);
00035 bool Loading_From_File(std::string file, Head& head, Tape& tape, Rules& rules);
```

4.17 Rules.cpp File Reference

```
#include "Rules.hpp"
```

4.18 Rules.hpp File Reference

```
#include <vector>
#include "Action.hpp"
```

Classes

- class [Rules](#)

Macros

- #define [R](#) 'R'
- #define [L](#) 'L'
- #define [N](#) 'N'

Rule class.

4.18.1 Macro Definition Documentation

4.18.1.1 L

```
#define L 'L'
```

4.18.1.2 N

```
#define N 'N'
```

Rule class.

4.18.1.3 R

```
#define R 'R'
```

4.19 Rules.hpp

[Go to the documentation of this file.](#)

```

00001 #pragma once
00002 #include <vector>
00003 #include "Action.hpp"
00004 #define R 'R'
00005 #define L 'L'
00006 #define N 'N'
00008 class Rules {
00009 public:
00013     ~Rules();
00019     std::string Get_State(int position) const;
00025     std::string Get_Word(int position) const;
00030     int Get_Amount_Of_States() const;
00035     int Get_Amount_Of_Words() const;
00042     Action* Get_Rule(int state, int word) const;
00051     void New_Rule(char move, std::string newword, std::string newstate, std::string word, std::string
state);
00058     bool Remove_Rule(std::string word, std::string state);
00063     std::string View() const;
00064 private:
00065     std::vector <std::string> states;
00066     std::vector <std::string> words;
00067     std::vector <std::vector <Action*>> table;
00068 };

```

4.20 Tape.cpp File Reference

```
#include "Tape.hpp"
```

4.21 Tape.hpp File Reference

```
#include <vector>
#include "Head.hpp"
```

Classes

- class [Tape](#)
Class for implementing the machine tape.

Macros

- #define [VOID](#) ""

4.21.1 Macro Definition Documentation

4.21.1.1 VOID

```
#define VOID ""
```


4.22 Tape.hpp

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002 #include <vector>
00003 #include "Head.hpp"
00004 #define VOID ""
00008 class Tape {
00009 public:
00014     void Add_Cell(std::string word);
00020     void Add_Cell(int position, std::string word);
00025     bool Remove_Cell();
00031     bool Remove_Cell(int position);
00037     void Set_Cell(int position, std::string word);
00043     std::string Get_Cell(int position) const;
00048     int Get_Size() const;
00054     std::string View(Head head) const;
00055 private:
00056     std::vector<std::string> cells;
00061     void Search(int& position);
00062 };
```


Index

- ~Rules
 - Rules, [8](#)
- Action, [5](#)
 - Get_Move, [5](#)
 - Get_New_State, [5](#)
 - Get_New_Word, [6](#)
 - Set_Move, [6](#)
 - Set_New_State, [6](#)
 - Set_New_Word, [6](#)
- Action.cpp, [15](#)
- Action.hpp, [15](#)
- ADD_CELL
 - Interface.hpp, [19](#)
- Add_Cell
 - Tape, [11](#), [12](#)
- EDIT_HEAD
 - Interface.hpp, [19](#)
- Edit_Head
 - Loading_From_File.cpp, [22](#)
 - Loading_From_File.hpp, [23](#)
- Execution.cpp, [16](#)
 - LIMIT, [16](#)
 - Turing_Mashine, [16](#)
- Execution.hpp, [16](#)
 - Turing_Mashine, [17](#)
- EXIT
 - Interface.hpp, [19](#)
- Get_Amount_Of_States
 - Rules, [9](#)
- Get_Amount_Of_Words
 - Rules, [9](#)
- Get_Cell
 - Tape, [12](#)
- Get_Move
 - Action, [5](#)
- Get_New_State
 - Action, [5](#)
- Get_New_Word
 - Action, [6](#)
- Get_Position
 - Head, [7](#)
- Get_Rule
 - Rules, [9](#)
- Get_Size
 - Tape, [12](#)
- Get_State
 - Head, [7](#)
- Rules, [9](#)
- Get_Word
 - Loading_From_File.cpp, [22](#)
 - Loading_From_File.hpp, [23](#)
 - Rules, [10](#)
- GO
 - Interface.hpp, [19](#)
- Head, [7](#)
 - Get_Position, [7](#)
 - Get_State, [7](#)
 - Set_Position, [7](#)
 - Set_State, [8](#)
- Head.cpp, [17](#)
- Head.hpp, [17](#)
- HELP
 - Interface.hpp, [19](#)
- Interface
 - Interface.cpp, [18](#)
 - Interface.hpp, [20](#)
- Interface.cpp, [18](#)
 - Interface, [18](#)
- Interface.hpp, [18](#)
 - ADD_CELL, [19](#)
 - EDIT_HEAD, [19](#)
 - EXIT, [19](#)
 - GO, [19](#)
 - HELP, [19](#)
 - Interface, [20](#)
 - NEW_RULE, [19](#)
 - REMOVE_CELL, [20](#)
 - REMOVE_RULE, [20](#)
 - SET_CELL, [20](#)
 - VIEW_RULES, [20](#)
 - VIEW_TAPE, [20](#)
- L
 - Rules.hpp, [25](#)
- Lab1.cpp, [21](#)
 - main, [21](#)
- LIMIT
 - Execution.cpp, [16](#)
- Loading_From_File
 - Loading_From_File.cpp, [22](#)
 - Loading_From_File.hpp, [24](#)
- Loading_From_File.cpp, [21](#)
 - Edit_Head, [22](#)
 - Get_Word, [22](#)
 - Loading_From_File, [22](#)

- New_Rule, 22
- Loading_From_File.hpp, 23
 - Edit_Head, 23
 - Get_Word, 23
 - Loading_From_File, 24
 - New_Rule, 24
- main
 - Lab1.cpp, 21
- N
 - Rules.hpp, 25
- NEW_RULE
 - Interface.hpp, 19
- New_Rule
 - Loading_From_File.cpp, 22
 - Loading_From_File.hpp, 24
 - Rules, 10
- R
 - Rules.hpp, 25
- REMOVE_CELL
 - Interface.hpp, 20
- Remove_Cell
 - Tape, 12
- REMOVE_RULE
 - Interface.hpp, 20
- Remove_Rule
 - Rules, 10
- Rules, 8
 - ~Rules, 8
 - Get_Amount_Of_States, 9
 - Get_Amount_Of_Words, 9
 - Get_Rule, 9
 - Get_State, 9
 - Get_Word, 10
 - New_Rule, 10
 - Remove_Rule, 10
 - View, 11
- Rules.cpp, 25
- Rules.hpp, 25
 - L, 25
 - N, 25
 - R, 25
- SET_CELL
 - Interface.hpp, 20
- Set_Cell
 - Tape, 13
- Set_Move
 - Action, 6
- Set_New_State
 - Action, 6
- Set_New_Word
 - Action, 6
- Set_Position
 - Head, 7
- Set_State
 - Head, 8
- Tape, 11
 - Add_Cell, 11, 12
 - Get_Cell, 12
 - Get_Size, 12
 - Remove_Cell, 12
 - Set_Cell, 13
 - View, 13
- Tape.cpp, 26
- Tape.hpp, 26
 - VOID, 26
- Turing_Mashine
 - Execution.cpp, 16
 - Execution.hpp, 17
- View
 - Rules, 11
 - Tape, 13
- VIEW_RULES
 - Interface.hpp, 20
- VIEW_TAPE
 - Interface.hpp, 20
- VOID
 - Tape.hpp, 26