Turing Machine

Generated by Doxygen 1.8.20

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 Tape Class Reference

Public Member Functions

- Tape ()
- Tape (std::vector< std::string > v)
- Tape (const Tape &toCopy)
- ~Tape ()
- int getSize () const
- int getIndex () const
- void setTape (Tape toCopy)
- bool isEmpty () const
- bool moveRight ()
- bool moveLeft ()
- bool writeBlank ()
- bool writeSymbol (std::string s)
- std::string readSymbol () const
- void clear ()
- void printTape () const
- void rewind ()

2.1.1 Constructor & Destructor Documentation

2.1.1.1 Tape() [1/3]

```
Tape::Tape ( )
```

Default constructor: creates an empty tape (i.e., no cell).

Returns

Sets data fields appropriately.

2.1.1.2 Tape() [2/3]

```
Tape::Tape (
            std::vector< std::string > v )
```

Parameterized constructor: creates a tape with symbols from v (one cell per symbol).

Parameters

v	A vector containing symbols.

Returns

Sets data fields appropriately.

2.1.1.3 Tape() [3/3]

```
Tape::Tape (
            const Tape & toCopy )
```

Copy constructor: performs a deep copy of toCopy. If toCopy is empty, then the tape is also empty.

Parameters

toCopy	A tape to copy.

Returns

Sets data fields appropriately.

2.1.1.4 ~Tape()

```
Tape::~Tape ( )
```

Destructor: deallocates memory as appropriate.

2.1.2 Member Function Documentation

2.1.2.1 clear()

```
void Tape::clear ( )
```

Clears the tape so that it is now empty.

Returns

Sets data fields appropriately.

2.1.2.2 getIndex()

```
int Tape::getIndex ( ) const
```

Returns the current tape location, i.e., cell. If the tape is empty, then return -1.

Returns

An integer denoting the current cell in the tape.

2.1.2.3 getSize()

```
int Tape::getSize ( ) const
```

Gets the current number of cells in the tape.

Returns

The integer number of cells currently in the tape.

2.1.2.4 isEmpty()

```
bool Tape::isEmpty ( ) const
```

Sees whether this tape is empty.

Returns

True if the tape is empty, or false if not.

2.1.2.5 moveLeft()

```
bool Tape::moveLeft ( )
```

Move the machine head left by one cell if the current location is not the beginning of the tape.

Returns

True if the tape can be moved to the left (i.e., there is at least one more cell to the left of the tape), or false if not.

2.1.2.6 moveRight()

```
bool Tape::moveRight ( )
```

Move the machine head right by one cell if the current cell is not the end of the tape.

Returns

True if the tape can be moved to the right (i.e., there is at least one more cell to the right of the tape), or false if not.

2.1.2.7 printTape()

```
void Tape::printTape ( ) const
```

Prints the cells on the tape from beginning to end.

Returns

Content of tape to standard output. NOTE: You do not need to test this method.

2.1.2.8 readSymbol()

```
std::string Tape::readSymbol ( ) const
```

Reads the symbol at the current cell in the tape if said cell is valid.

Returns

True if the tape can be read at the current cell, or false if not.

2.1.2.9 rewind()

```
void Tape::rewind ( )
```

Rewinds the tape to the very beginning (i.e., first cell) if the tape is not empty.

Returns

Sets data fields appropriately.

2.1.2.10 setTape()

```
void Tape::setTape (
            Tape toCopy )
```

Sets the symbols on the tape according to toCopy and overwrites existing symbols, if any. If toCopy is empty, then the tape is also reset to having no symbol.

Parameters

toCopy	A tape to copy.

Returns

Sets data fields appropriately.

2.1.2.11 writeBlank()

```
bool Tape::writeBlank ( )
```

Writes a blank symbol (" ") to the current cell in the tape if said cell is valid.

Returns

True if the tape can be written at the current cell, or false if not.

2.1.2.12 writeSymbol()

```
bool Tape::writeSymbol (
            std::string s )
```

Writes the symbol specified by s to the current cell in the tape if said cell is valid.

Parameters

s	A symbol.

Returns

True if the tape can be written at the current cell, or false if not.

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

- Tape.hpp

2.2 TuringMachine Class Reference

Public Member Functions

- TuringMachine ()

- TuringMachine (Tape toCopy, const std::vector< std::string > &p)
- ~TuringMachine ()
- std::vector< std::string > getSymbols () const
- Tape getTape ()
- bool setSymbols (std::vector< std::string > p)
- bool setTape (Tape &toCopy)
- bool setupCheck ()
- void reset ()
- bool patternMatch ()

2.2.1 Constructor & Destructor Documentation

2.2.1.1 TuringMachine() [1/2]

```
TuringMachine::TuringMachine ( )
```

Default constructor: creates an empty Turing machine.

Returns

Sets data fields appropriately.

2.2.1.2 TuringMachine() [2/2]

```
TuringMachine::TuringMachine (
            Tape toCopy,
            const std::vector< std::string > & p )
```

Parameterized constructor: creates a Turing machine with tape whose content is identical to toCopy and whose legal symbols are as specified by p.

Parameters

toCopy	A tape to copy content from.
p	A vector containing legal symbols.

Returns

Sets data fields appropriately.

2.2.1.3 ~TuringMachine()

```
TuringMachine::~TuringMachine ( )
```

Destructor: deallocates memory as appropriate.

2.2.2 Member Function Documentation

2.2.2.1 getSymbols()

```
std::vector<std::string> TuringMachine::getSymbols ( ) const
```

Gets the legal symbols that can be written to the tape.

Returns

A vector containing all the legal symbols in the Turing machine.

2.2.2.2 getTape()

```
Tape TuringMachine::getTape ( )
```

Gets the tape from the Turing machine.

Returns

The appropriate tape.

2.2.2.3 patternMatch()

```
bool TuringMachine::patternMatch ( )
```

Checks to see if content of the tape matches the pattern: a^i b^j c^j, where i and j are integers greater than or equal to 1. Also checks if the tape contains illegal symbols.

Returns

True if the pattern matches and there are no illegal symbols, or false otherwise.

2.2.2.4 reset()

```
void TuringMachine::reset ( )
```

Resets the Turing machine to empty tape and no symbol list.

Returns

Set appropriate data fields.

2.2.2.5 setSymbols()

```
bool TuringMachine::setSymbols (
            std::vector< std::string > p )
```

Sets the legal symbols that can be written to the tape, overwriting old symbols if applicable.

Parameters

p	A vector containing legal symbols.

Returns

True if the symbols have been set, or false otherwise.

2.2.2.6 setTape()

```
bool TuringMachine::setTape (
            Tape & toCopy )
```

Sets the tape content to be as specified by toCopy, overwriting old content if applicable.

Parameters

toCopyA	tape to copy content from.

Returns

True if the tape have been set, or false otherwise.

2.2.2.7 setupCheck()

```
bool TuringMachine::setupCheck ( )
```

This method is specific to the pattern we're checking. For our purpose, this method simply checks that there are exactly 3 legal, no duplicate symbols.

Returns

True if the setup is correct, or false otherwise.

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

- TuringMachine.hpp

Index