

Assignment 4 Specification

Di Wu, 400117248, wud43

April 13, 2019

This Module Interface Specification (MIS) document contains modules, types and methods for implementing the state of a game of Forty Thieves solitaire.

[The parts that you need to fill in are marked by comments, like this one. In several of the modules local functions are specified. You can use these local functions to complete the missing specifications. —SS]

[As you edit the tex source, please leave the `wss` comments in the file. Put your answer **before** the comment. This will make grading easier. —SS]

Board Module

Generic Template Module

Stack

Uses

N/A

Syntax

Exported Types

Board = ?

Exported Constants

None

Exported Access Programs

| Routine name | In | Out | Exceptions |
|--------------|---|-------------------------------|------------|
| Board | Integer | Board | none |
| set | \mathbb{N}, \mathbb{N} | | none |
| PrintBoard | | | none |
| NextStage | | | none |
| CountNeigh | seq of (seq of \mathbb{N}) , \mathbb{N} , \mathbb{N} | \mathbb{N} | none |
| toVec | | seq of (seq of \mathbb{N}) | none |

Semantics

State Variables

None

State Invariant

None

Assumptions & Design Decisions

- The Board constructor is called for each object instance before any other access routine is called for that object. The constructor can only be called once.

Access Routine Semantics

Stack(*num*):

- transition: $S := BoardState[num][num]$
- output: $out := self$
- exception: none

set(*a*, *b*):

- transition: $S[a][b] := 1$
- exception: none

PrintBoard():

- exception: none

NextStage():

- transition: $(i, j \in [0...BoardState.size()])((CountNeigh(BoardState, i, j) == 3 \wedge BoardState[i][j] == 0) \implies BoardState[i][j] = 1, (\neg CountNeigh(BoardState, i, j) == 3) \wedge BoardState[i][j] == 1) \implies BoardState[i][j] = 0)$
- exception: none

CountNeigh(*s*, *a*, *b*):

- $out := (+1 | i \in [a-1...a+1] \wedge j \in [b-1...b+1] \wedge i \neq a \wedge j \neq b) BoardState[i][j] == 1)$
- exception: None

toVec():

- output: $out := BoardState$
- exception: None

Critique of Design

[Write a critique of the interface for the modules in this project. Is there anything missing? Is there anything you would consider changing? Why? —SS]

Firstly, I believe that I created a function called `toVec()` in the `BoardADT`. My initial thoughts is that it will be convenient for achieving the other goals of other functions. But it violates the information hiding policy. Secondly, in the `Board` constructor, I used a number as the input to create a `num * num` board. That is a little bit complex. Instead of doing it, I believe that I can directly use the content from the file-reading and build the `BoardADT` directly.