# Validation of “Description of States”

To validate the states we will look at the USE-cases again to see if every USE-case is implemented. To do this we look at the basic flow and trigger of every use case and see what states we use to realize this.

We also validate the states to the relations. For every USE-case we looked at what states would be necessary to achieve it.

## Starting the machine

**Preconditions:** -

**Trigger:** Booting the machine / finished the abort or start/stop routine

**Postconditions:** The machine starts the sorting process.

|  |  |  |
| --- | --- | --- |
| Basic Flow | State | Explanation |
| Before Trigger | Any State | It does not really mater which state the machine is in before the trigger |
| After Trigger | Initial State | Initial state is the first state, so after booting the machine we will be here.  Finishing the abort or start/stop routine will also end in the initial state |
| 1. Machine puts devices in their initial state. | Initial State + Initial State 2 + Resting State | The only thing that needs to be put into an initial state is the sorter mechanism. In initial state the machine moves the sorter up until it touches the push button. It then transitions to initial state 2 where it starts moving down. After a set amount of time it will stop moving the sorter and transition to the resting state. This way we know exactly where the sorter is positioned |
| 1. The user presses the START/STOP button | Running State | From the Resting State the transition to the running state is pressing the START/STOP button |
| Post-conditions | Running State | The running state is the start of the sorting process |

## Stopping the machine

**Preconditions:** The machine is running.

**Trigger:** The START/STOP button is pressed.

**Post-conditions:** The machine is sent into an inactive state with no process interrupted.

|  |  |  |
| --- | --- | --- |
| Basic Flow | State | Explanation |
| Preconditions | Not initial state, initial state 2 or abort | When the machine is not in any of these states it is running. |
| After Trigger | One of the Timer Start statesaria the machine is not in any of these states it is running.s means they do not use states. This also means we can'sitioned. th | When the START/STOP is pressed the machine transitions to a timer start state, which starts a timer and stops the feeder mechanism. |
| 1. The machine finishes sorting the disks currently in the machine | One of the sorting states | While the timer is running the machine keeps sorting. The timer is the time it takes for the conveyor belt to make a complete rotation, guaranteeing there are no more disks on the belt. |
| 1. The machine enters an inactive state and will not take any more disks form the storage\* unless the START/STOP button is pressed. | Initial State + Initial State 2 + Resting State | After going through the initialize process we go back to the resting state, which waits on the START/STOP button. |
| Postconditions | Resting StateRstiresting state, which waits on the START/STOP button. timer is the time it takes for the conveyor belt to make a complete ro | Resting state in an inactive state and we finished the sorting process. |

## Sort unsorted discs

**Preconditions:** The machine is not already running.

**Trigger:** The user provides unsorted discs and presses the “START” button.

**Postconditions:** There are no unsorted disks left, all sorted disks are in a container based on their colour

|  |  |  |
| --- | --- | --- |
| Basic Flow | State | Explanation |
| Preconditions | Resting State | The program first initializes and then waits for the user to press that start button. This waiting happens in the Resting State. In the resting state the machine is not running |
| After Trigger | Running State | Pressing START/STOP is the input to transition to the running state |
| 1. An unsorted disc is moved to the colour detector | Running State 2 + Running State 3 | When moving to the colour detector it will have to pass the position Sensor which is the input to move to Running State 2, the disc is then still in front of the position sensor so the program moves to running state 3 |
| 1. The machine decides to which of the two containers the disc needs to be moved | Running State 2 + Running State 3  OR  Motor Up  + White-Wait | Depending on whether the disc is white or black the sorter either needs to move down or keep its down position. If it keeps its down position it should just keep checking for an unsorted disc and when it detects one it will move to running state 3  If it needs to move up the colour detector will detect a white disc and therefore transition to Motor Up. Moving the sorter up will trigger the pushButton, which is the input to transition to White-Wait |
| 1. The machine moves the disc to the designated container | Running State 2 + Running State 3  OR  Motor Down + Running state 2 | If the sorter did not detect a white disc we are still waiting like in basic flow 2.  If it did detect one then while the disc is moving to the designated container the sorttimer will count down making the machine transition to Motor Down |
| 1. The machine repeats step 2 through 4 until all discs have been sorted | - |  |
| 1. The machine pauses within 4 seconds | Initial State + Initial State 2 + Resting State | If there are no discs anymore the machine will stay in running state 2 waiting for the timer interrupt which will come within 4 seconds, making the machine transition to initial state.  There it will reset the sorter and transition to the resting state |
| Post-conditions | Resting State | We repeated the sorting step until all discs where sorted, meaning all discs are now sorted |

## Abort the process

**Preconditions:** The machine is sorting disks

**Trigger:** The use wants to immediately stop the machine.

**Post-conditions:** The machine stopped running and is ready to start again.

|  |  |  |
| --- | --- | --- |
| Basic Flow | State | Explanation |
| Preconditions | Every that is not initial state, initial state 2, resting state or Abort cdlsls | All other states are states in which discs are being sorted |
| After Trigger | Abort | Every state (apart from the one mentioned in before trigger) have a line to abort with Abort as input |
| 1. The machine stops transporting the discs. And doesn’t put any more discs on the transporting mechanism. | Abort | Because the machine is now in the abort state, which has all outputs set to 0, nothing will be moving. |
| 1. The user is required to remove all discs that are neither in the container unit nor sorted. | Abort | The machine will remain in Abort until the user presses START/STOP. This means everything is stopped and the user can safely remove all discs |
| 1. When the user removed all unsorted discs that where not in the container unit he presses the START/STOP button. | Initial State + Initial State 2 + Resting State | Pressing the START/STOP button is the input for the transition to Initial State  There it will reset the sorter and transition to the resting state |
| Post-conditions | Resting State | We are in the resting state, so the machine has stopped running. The resting State is also the state from which you can start the machine again |

Booting of the machine & Shutting down the machine does nothing with our software. This means they do not use states. This also means we can’t validate those USE-Cases here.