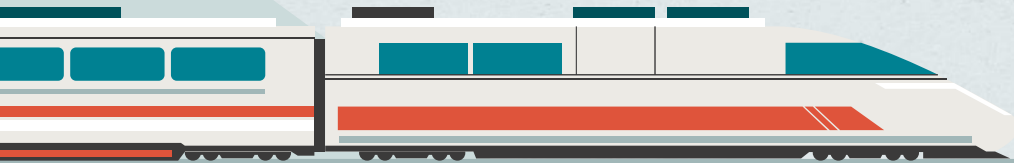


Scheduler Collision

Brief Explanation of USLP10



Group 121

André Pinho

Bianca Almeida

Carlota Lemos

Eduardo Almeida

Marcelo Carvalho

Professores

João Miguel Carneiro

Alberto Gomes Mendes

Jorge Santos



Table of contents

01

**Schedule
Explanation**

02

**Single Track
Collision**

03

**Two Single Track
Collision**

04

**Siding
Collision**



1- Schedule Explanation

Creation: Each time we add a complete train, the schedule generates all events again. It generates them in order of event date.

Events: For each event, check whether it is possible to create the event or generate a waiting event, placing it at the top of the event list.

Schedule Details

Schedule Event 1:

Verify Collision on Segment 1 - False
Create **Movement To Segment** Event
Initial Pos. – Station A 10:00
Final Pos. – Segment 1 10:00

Schedule Event 2:

Create **Movement On Segment** Event
Initial Pos. – Segment 1 10:00
Final Pos. – Station B 10:12

Schedule Event 3:

Verify Collision on Segment 1 - True
Create **Waiting** Event
Initial Pos. – Station A 10:02
Final Pos. – Segment 1 10:12

Schedule Event 4:

Verify Collision on Segment 1 - False
Create **Movement To Segment** Event
Initial Pos. – Station A 10:12
Final Pos. – Segment 1 10:12

Schedule Event 5:

Create **Movement On Segment** Event
Initial Pos. – Segment 1 10:12
Final Pos. – Station A 10:28

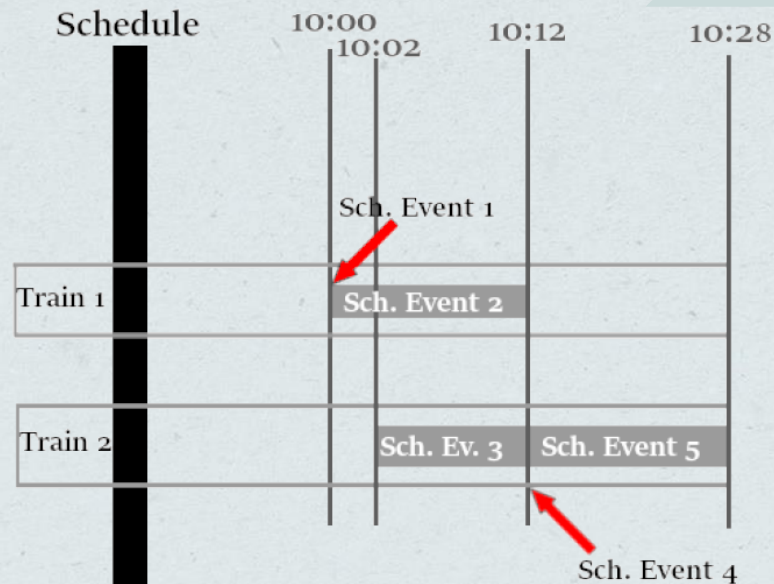


Fig.1 - Schedule Graphical Display

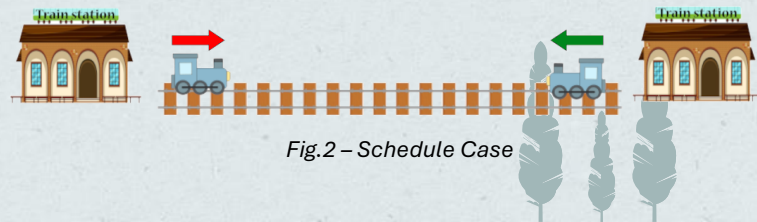


Fig.2 – Schedule Case



Fig.3 – Collision Case 1

Context

Two trains travel at the same time on a line connecting Station A to Station B. These stations are connected by a line consisting of a single segment with only one track.

2- Single Track Collision

To prevent head-on collisions on single-track railway segments, occupancy is centrally managed by the **ScheduleGenerator**. Each **Segment** with 1 track doesn't allow trains with different direction at the same time, and conflicts are resolved by delaying the train that found the collision and inserting waiting events into its schedule.

```
===== SCHEDULE RESULT =====
TrainSchedule{train=Train: 1}
ScheduleEvent{trainId=1, type=MOVEMENT_TO_SEGMENT, start=A, end=Segment: 1
interval=From 10:00:00 - 25/12/2025 to 10:00:00 - 25/12/2025}

ScheduleEvent{trainId=1, type=MOVEMENT_IN_SEGMENT, start=Segment: 1, end=B
interval=From 10:00:00 - 25/12/2025 to 10:14:02 - 25/12/2025}

TrainSchedule{train=Train: 2}
ScheduleEvent{trainId=2, type=WAITING, start=B, end=Segment: 1
interval=From 10:00:01 - 25/12/2025 to 10:14:02 - 25/12/2025}

ScheduleEvent{trainId=2, type=MOVEMENT_TO_SEGMENT, start=B, end=Segment: 1
interval=From 10:14:03 - 25/12/2025 to 10:14:03 - 25/12/2025}

ScheduleEvent{trainId=2, type=MOVEMENT_IN_SEGMENT, start=Segment: 1, end=A
interval=From 10:14:03 - 25/12/2025 to 10:28:05 - 25/12/2025}
=====
```

Fig.4 - Test Result In App

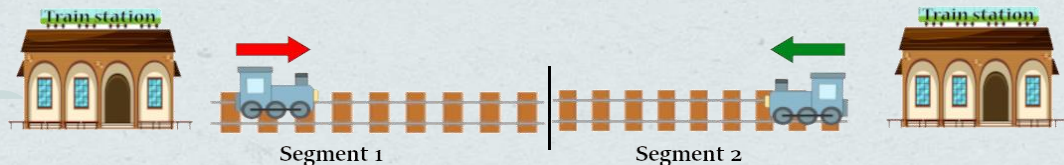


Fig.5 – Collision Case 2

Context

Two trains travel at the same time on a line connecting Station A to Station B. These stations are connected by a line consisting of Two segment with only one track each.

3- Two Single Track Collision

Since the system predicts collisions segment by segment, there may be cases where it is necessary to reverse certain movements. For this reason, the Generator has the ability to aggregate line segments that have only one track and are consecutive. This makes the problem the same as that seen in (1 - Single Track Collision).

```
===== SCHEDULE RESULT =====
TrainSchedule{train=Train: 1}
ScheduleEvent{trainId=1, type=MOVEMENT_TO_SEGMENT, start=A, end=1 2
interval=From 10:00:00 - 25/12/2025 to 10:00:00 - 25/12/2025}

ScheduleEvent{trainId=1, type=MOVEMENT_IN_SEGMENT, start=1 2, end=B
interval=From 10:00:00 - 25/12/2025 to 10:22:26 - 25/12/2025}

TrainSchedule{train=Train: 2}
ScheduleEvent{trainId=2, type=WAITING, start=B, end=2 1
interval=From 10:00:00 - 25/12/2025 to 10:22:26 - 25/12/2025}

ScheduleEvent{trainId=2, type=MOVEMENT_TO_SEGMENT, start=B, end=2 1
interval=From 10:22:27 - 25/12/2025 to 10:22:27 - 25/12/2025}

ScheduleEvent{trainId=2, type=MOVEMENT_IN_SEGMENT, start=2 1, end=A
interval=From 10:22:27 - 25/12/2025 to 10:44:53 - 25/12/2025}
=====
```

Fig.6 - Test Result In App

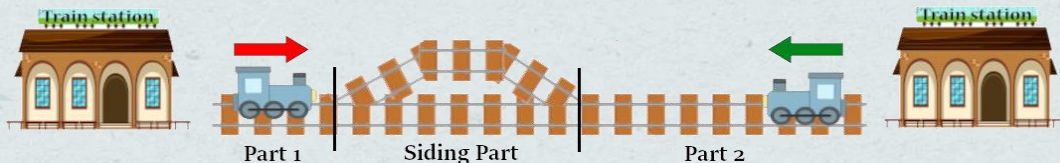


Fig.7 – Collision Case 3

Context

Two trains travel at the same time on a line connecting Station A to Station B. These stations are connected by a line consisting of a single segment with only one track that have a siding in the first 1km with 100 m length .

4- Siding Collision

In order for trains to use sidings, we decided to divide the segments that contain them so that occupancy can be checked part by part using the (1 - Single Track Collision) method. In the case of a “Siding Part”, it will be treated as a line segment with two tracks, thus allowing crossing. At the siding exit, it checks whether the next part of the segment is available or not.

```
===== SCHEDULE RESULT =====
TrainSchedule{train=Train: 1}
ScheduleEvent{trainId=1, type=MOVEMENT_TO_SEGMENT, start=A, end=Segment: 1 Part. 1
interval=From 10:00:00 - 25/12/2025 to 10:00:00 - 25/12/2025}
ScheduleEvent{trainId=1, type=MOVEMENT_IN_SEGMENT, start=Segment: 1 Part. 1, end=Siding1
interval=From 10:00:00 - 25/12/2025 to 10:02:50 - 25/12/2025}
ScheduleEvent{trainId=1, type=MOVEMENT_IN_SIDING, start=Siding1, end=Siding1
interval=From 10:02:50 - 25/12/2025 to 10:03:09 - 25/12/2025}
ScheduleEvent{trainId=1, type=WAITING, start=Siding1, end=Segment: 1 Part. 2
interval=From 10:03:09 - 25/12/2025 to 10:11:31 - 25/12/2025}
ScheduleEvent{trainId=1, type=MOVEMENT_TO_SEGMENT, start=Siding1, end=Segment: 1 Part. 2
interval=From 10:11:32 - 25/12/2025 to 10:11:32 - 25/12/2025}
ScheduleEvent{trainId=1, type=MOVEMENT_IN_SEGMENT, start=Segment: 1 Part. 2, end=B
interval=From 10:11:32 - 25/12/2025 to 10:23:03 - 25/12/2025}
TrainSchedule{train=Train: 2}
ScheduleEvent{trainId=2, type=MOVEMENT_TO_SEGMENT, start=B, end=Segment: 1 Part. 2
interval=From 10:00:00 - 25/12/2025 to 10:00:00 - 25/12/2025}
ScheduleEvent{trainId=2, type=MOVEMENT_IN_SEGMENT, start=Segment: 1 Part. 2, end=Siding1
interval=From 10:00:00 - 25/12/2025 to 10:11:31 - 25/12/2025}
ScheduleEvent{trainId=2, type=MOVEMENT_IN_SIDING, start=Siding1, end=Siding1
interval=From 10:11:31 - 25/12/2025 to 10:11:50 - 25/12/2025}
ScheduleEvent{trainId=2, type=MOVEMENT_TO_SEGMENT, start=Siding1, end=Segment: 1 Part. 1
interval=From 10:11:50 - 25/12/2025 to 10:11:50 - 25/12/2025}
ScheduleEvent{trainId=2, type=MOVEMENT_IN_SEGMENT, start=Segment: 1 Part. 1, end=A
interval=From 10:11:50 - 25/12/2025 to 10:14:40 - 25/12/2025}
=====
```

Fig.8 - Test Result In App



Thanks!

Do you have any questions?
mabecdev@gmail.com

Group 121

André Pinho
Bianca Almeida
Carlota Lemos
Eduardo Almeida
Marcelo Carvalho