Collaborative Robotics Modeling

FM

CS 2018

Davide Santambrogio a.k.a M. Santa

Federico Sandrelli a.k.a Pier Lanzi

Andrea Tricarico a.k.a S. Zanero

## 

## Sommario

[Sommario](#_y99skdtdohw9) 2

[**Introduction**](#_y4516bo9p0x1) **5**

[**Domains**](#_dfvzskhna7vs) **5**

[**Predicates (event / state)**](#_kibs9mj9qzi8) **6**

[Distance(cellDomain,cellDomain,integer)](#_igdeh1k6l4oa) 6

[BlueArea(cellDomain)](#_idrp7rkkx7q3) 6

[BinArea(cellDomain)](#_sxh78riw4k4q) 6

[WorkArea(cellDomain)](#_fd3bd5fyq16h) 6

[LoadArea(cellDomain)](#_s2ibb1i1kent) 6

[ReleaseArea(cellDomain)](#_3x4uxnotda6m) 6

[BasePosition(cellDomain)](#_gotwzskn4jln) 6

[ArmPosition( cellDomain, cellDomain, cellDomain, cellDomain, cellDomain)](#_6enn3bawxr7x) 6

[BaseSpeed(speedValue)](#_qy8mkhdduftg) 6

[ArmSpeed(speedValue)](#_ejtx6f7rmg3g) 6

[Working(operationType)](#_4ng40bj6gk2t) 6

[Occupies(armDomain)](#_fch0ppor8sgs) 6

[Mode(stateType)](#_vfw0e986a9yi) 6

[BinFull](#_arrwy7n8q9rv) 6

[BinEmpty](#_jic0httb899k) 6

[HumanBodyPosition(cellDomain)](#_78kewb4q3tfg) 6

[HumanHeadPosition(cellDomain)](#_3pspjf6hbf69) 6

[HumanArmPosition(cellDomain, cellDomain, cellDomain)](#_e5eawbdjyk97) 6

[Signal](#_87ft32ouo9ip) 7

[AddedWP](#_jrqico28luv) 7

[RemovedWP](#_s9kcc9ndbc2s) 7

[**Variables**](#_37mw1lf1xk2m) **7**

[**Axioms**](#_fmd4zrmfvok4) **8**

[Cell](#_5qlfewewede3) 8

[Distance((x,y),(w,z), d)](#_z40xyjbvesb2) 8

[Layout](#_bm5cclv9vqb3) 10

[Walls((x,y))](#_mfmb2umpi336) 10

[BlueArea((x,y))](#_oax4j9phq1ex) 10

[WorkArea((x,y))](#_3mg6temz513p) 10

[BinArea((x,y))](#_1hbeqjkspm97) 11

[LoadArea(x,y)](#_5dl76sjebt0h) 11

[ReleaseArea(x,y)](#_k1vmgj8dab8) 11

[WorkPieces](#_lkhhoicfrrl0) 11

[WPposition](#_vq42ck7bidoe) 11

[Robot](#_q1nsp5uh9kh2) 12

[BasePosition](#_glapnbg18m9i) 12

[There is Only One Base](#_8nc95wgbx3i9) 12

[ChangeSpeed(up/down)](#_inwcvvclvjt7) 12

[Base Speed Change](#_1e2sfoej0yjt) 13

[Arm](#_w9r0gm7twuus) 13

[ArmPosition(celli, cellj)](#_l6sqd8pbje73) 16

[There is Only One Arm](#_c1f1q2kjfqz3) 16

[ArmBase Movement](#_rz3gxkqiwgw) 16

[EndEffector Movement](#_j0qy6dxeacc9) 16

[EEFull](#_rycsnadmurac) 16

[Arm in Rest Position](#_ycult9qv3gfb) 17

[ArmSpeed - Mode(Manipulator)](#_2hdfaurylvcw) 18

[ArmSpeed(still)](#_jjk6lasvjmgd) 18

[RobotRotation](#_uf3xmiwwuwh3) 19

[Bin](#_rq2jxz3k5gyh) 20

[BinPosition](#_n884yk730jnr) 20

[There is Only one Bin](#_x94x3sgdzacm) 20

[Arm Speed Change](#_xhpa6wg8nus8) 21

[Operator](#_aa0225plfir) 22

[Operator Position](#_dildk4fln5lm) 22

[Cells Occupied by a HumanArm are Connected](#_9tprdtoqjm24) 22

[External Signals](#_e7crlcvsyugf) 23

[Signal](#_z3kk662tllka) 23

[CloseTo Signals](#_6iuxg27mmb5z) 23

[BaseCloseToBody](#_trrasbdgc0) 24

[BaseCloseToArms](#_9imo6w29cg2i) 24

[BaseCloseToHead](#_t6tyj6ktglmv) 24

[EECloseToBody](#_nkah37vluahh) 25

[EECloseToArms](#_xj4dq4ml87za) 25

[EECloseToHead](#_ze4p66lfp1vq) 25

[ArmBaseCloseToBody](#_9fx67mxaomyu) 26

[ArmBaseCloseToArms](#_bj1d2aml1qrg) 26

[ArmBaseCloseToHead](#_jubgdpqvmel2) 26

[Robot-Layout Interaction](#_1oldn9f2zc2g) 28

[Robot Body Always in Blue Area](#_lkg9atlgzii6) 28

[Base Speed](#_7y6wduys4ioa) 28

[Wall’s Constraint](#_lanndyi4fsrl) 29

[Operating Mode - Mode(mobile)](#_uglxrfrhn1de) 29

[Operating Mode - Mode(manipulator)](#_ym5wvcgr4l5c) 29

[Robot-Operator Interaction](#_60nzwxqczokq) 30

[FSA](#_m8b8kh1niiej) 30

[Base Close to Operator](#_yewrldoyttvg) 30

[KeepDistance](#_apmjtf6iw2zo) 30

[DontMoveWhenDistanceIsOne](#_nmee47rstqr5) 30

[SlowInWorkStation](#_6lzjcxr1a639) 30

[Robot does not entrap the operator](#_d0xe0wywz99w) 30

[Robot Workflow](#_orkacs3tani2) 31

[Destination of the robot in mobile mode](#_rqzjidabhkzk) 31

[Working in Manipulator Mode](#_ks39zro10ssf) 32

[Operation Timing](#_2t7l59f5o96y) 32

[Working(pickWP/putWP)](#_1s7cjyqw6om1) 32

[Ending PickWP/PutWP](#_vpxg62qh1q96) 32

[Working Procedure](#_add5r8d1fuoo) 32

[**Safety Property**](#_77vhcxnzz430) **33**

[Discuss why you think your model would never violate this safety property.](#_r73uox23txg7) 34

# Introduction

In this document we are describing the Human-Robot Collaboration (HRC) between the operators and the robot Kuka. In particular we are modelling the constraints that describes the robot behavior to assure the safety of the operator without penalize too much the efficiency of the robot.

# Domains

cellDomain: {(x, y) | x >= 0, y >= 0}

robotDomain: {base}

binDomain: {bin1, bin2, bin3}

binStateDomain: {full, empty}

humanDomain: { humanHead, humanBody, humanArm1, humanArm2}

operationType: {pickWP, placeWP}

stateType: {manipulator, mobile}

speedValue: {still, low, medium, high}

# Predicates (event / state)

#### Distance(cellDomain,cellDomain,integer)

#### BlueArea(cellDomain)

#### BinArea(cellDomain)

#### WorkArea(cellDomain)

#### LoadArea(cellDomain)

#### ReleaseArea(cellDomain)

#### BasePosition(cellDomain)

#### ArmPosition( cellDomain, cellDomain, cellDomain, cellDomain, cellDomain)

#### BaseSpeed(speedValue)

#### ArmSpeed(speedValue)

#### Working(operationType)

#### Occupies(armDomain)

#### Mode(stateType)

#### BinFull

#### BinEmpty

#### HumanBodyPosition(cellDomain)

#### HumanHeadPosition(cellDomain)

#### HumanArmPosition(cellDomain, cellDomain, cellDomain)

#### Signal

#### AddedWP

#### RemovedWP

# Variables

: cellDomain;

operation: operationType;

state: stateType;

integer

# Axioms

## Cell

NOTE ON CELLS

This model is based on “cells”. Cells identify a 20cm x 20cm area and they represent the granularity of our sensor. The layout is divided in cells and the position of every object that enters the scene is defined by the cells it occupies. Movements are therefore discrete.

Throughout this project, cells will be identified in two ways:

* With its coordinates on the 2D plane  
  (for when we need the coordinates):
* With a variable:

So, for example:

Are two equivalent axioms. They are saying that if the position of the base is detected on (generic cell), then ((x, y)) must be one of the cells in the blue area.

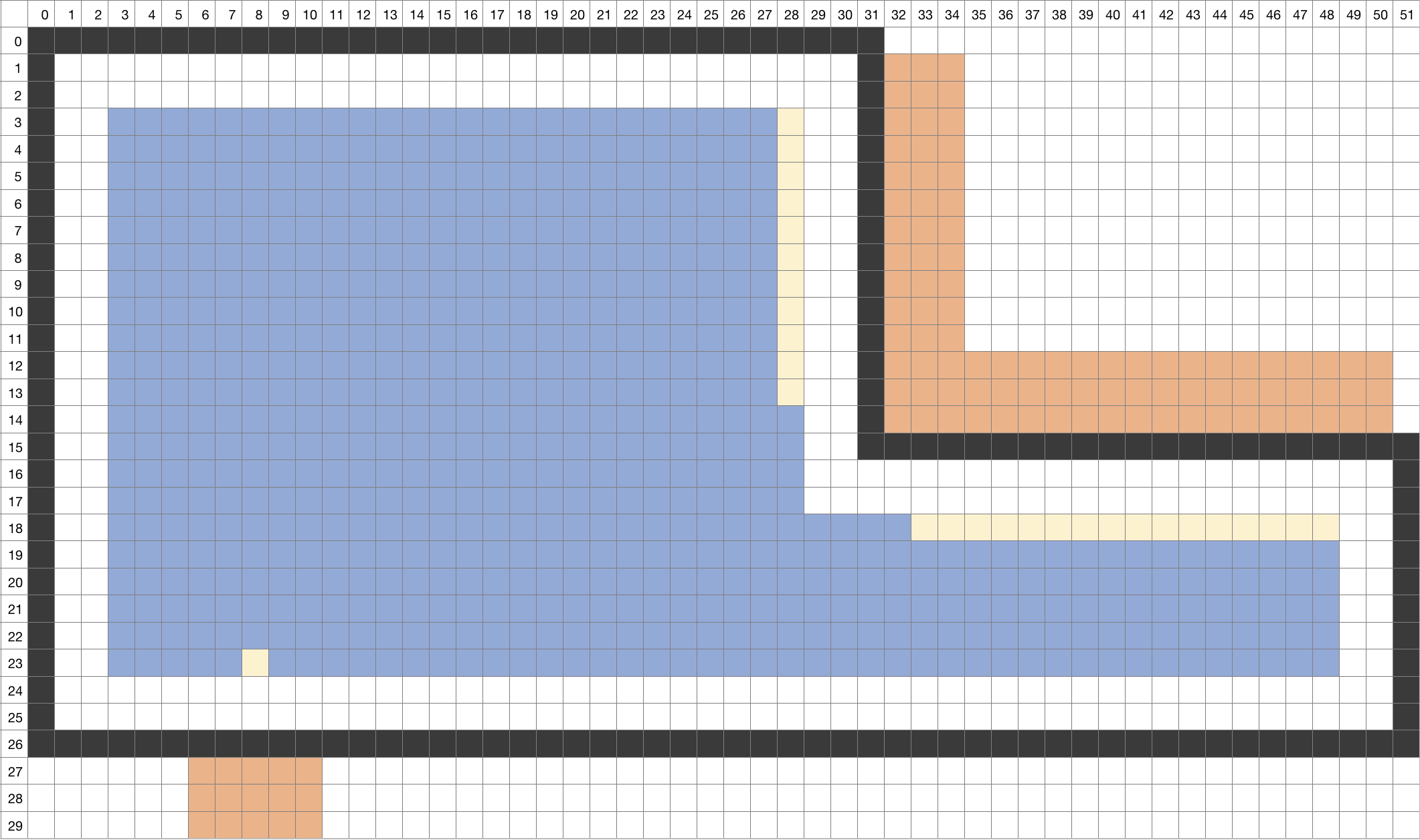
### Distance((x,y),(w,z), d)

This predicate checks whether the two cells, of which the coordinates are specified, are [d] cells apart from each other (i.e. there are [d-1] cells between them). This distance measure is an approximation that measures distance with a square perimeter. We have chosen to calculate distance this way because it fits well with the how movements have been modelled.

*Note that “Distance(cell1, cell2, 0)” means that cell1 and cell2 are the same cell.*

### 

## Layout



### Walls((x,y))

This predicates delimits the area of the walls, that delimit the area in which the robot can operate.

### BlueArea((x,y))

This predicates delimits the area of the blue cells. These are the allowed positions for the center of the robot’s base. It will be more clear after reading the “Robot” chapter.

### WorkArea((x,y))

### BinArea((x,y))

### LoadArea(x,y)

### ReleaseArea(x,y)

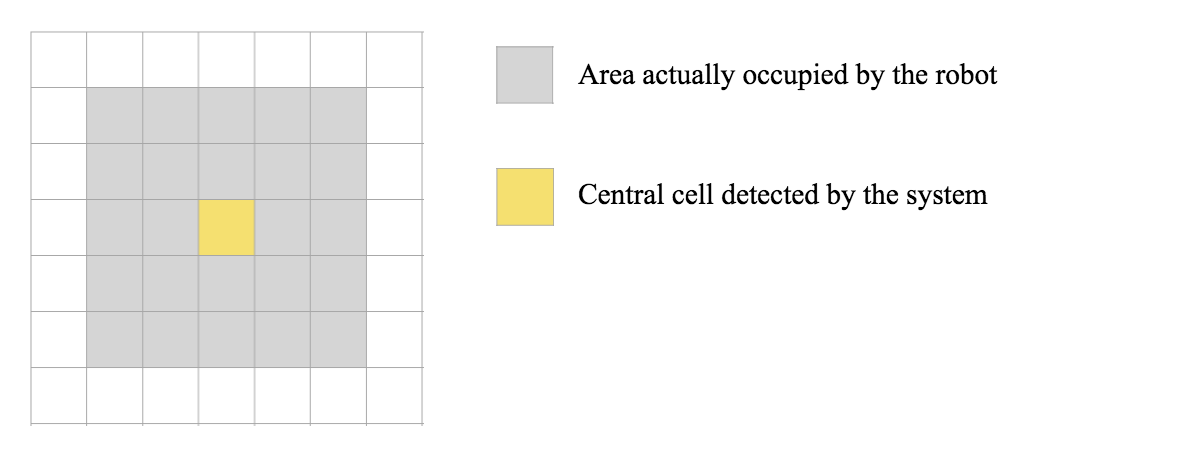
## WorkPieces

WorkPieces (WPs) are characterized by the single cell in which they are located, with no particular constraints.

### WPposition

This predicate implies nothing but it’s useful for future axioms.

## Robot



### BasePosition

This predicate assures that the robot always has a position and this is identified by a unique cell (it represents the cell occupied by the center of the robot’s base).

### There is Only One Base

In the scenario of this project we only have one robot, this predicate ensures that this fact is respected.

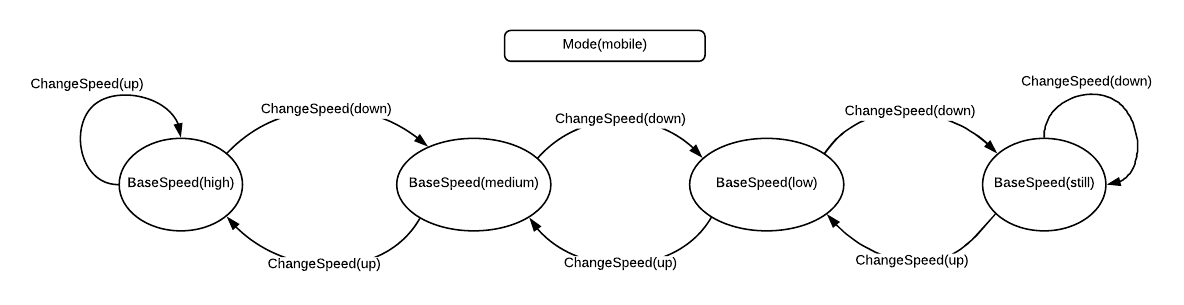
### ChangeSpeed(up/down)

ChangeSpeed is a signal modelled as an event, that lets the robot know it should change its operating speed.

(NOTE: ChangeSpeed is an event, and therefore, it is instantaneous)

### Base Speed Change

These axioms model the change of speed in the robot’s movements.

[](https://www.lucidchart.com/documents/edit/1bf36cf4-9790-4347-8c97-ab4e171d4f20/1?callback=close&name=docs&callback_type=back&v=2069&s=592)

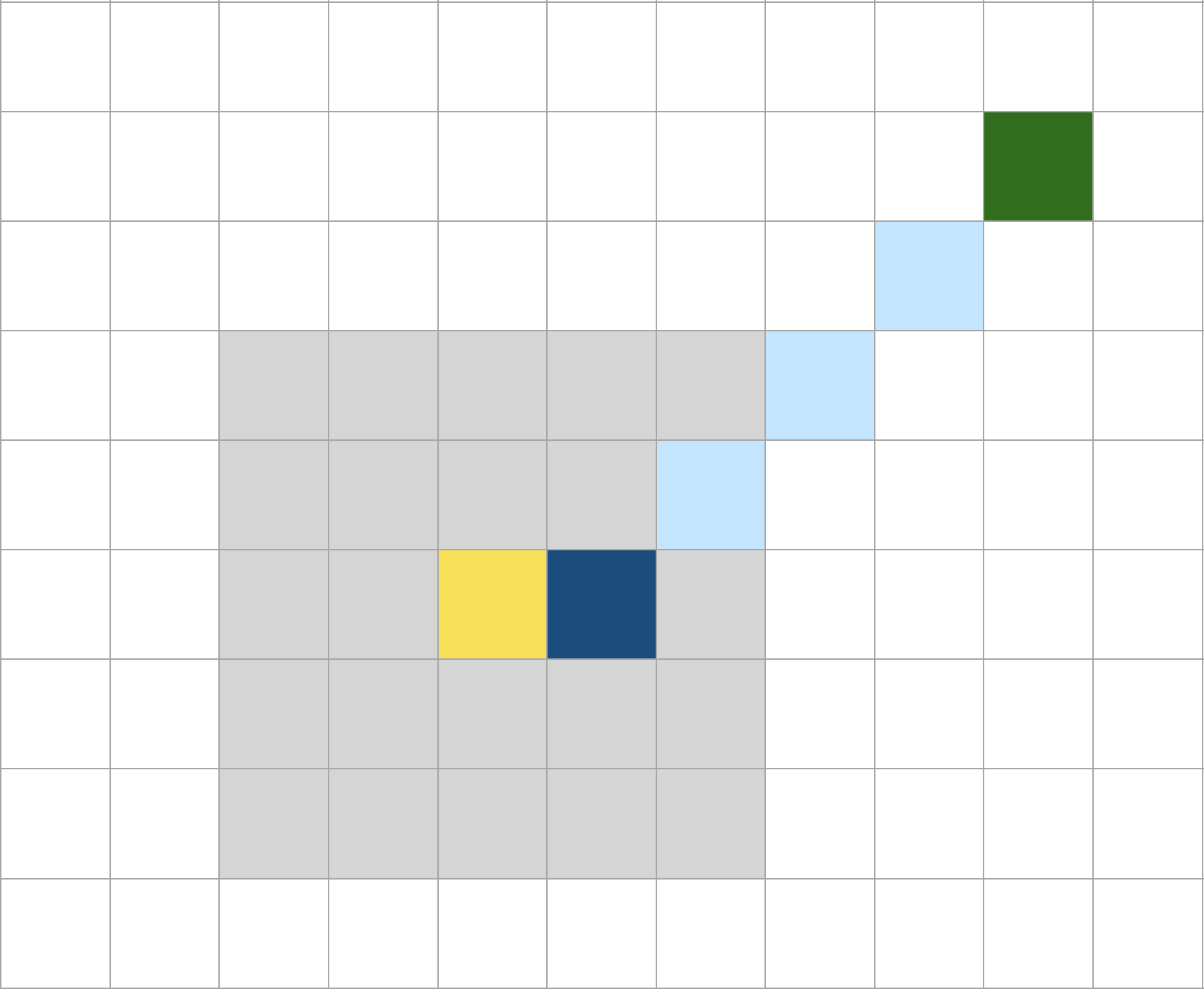
## Arm

The arm of the robot is modelled by the ArmBase and the EndEffector. These two components are on the opposite extremities of the arm, therefore it is sufficient to consider only these two elements, knowing that the cells connecting them (in a straight row) are also occupied by the arm.

This simplification will allow us to simplify our axioms while still keeping all the information required to ensure safety.

The fully extended arm can occupy a straight row of 5 cells (total of 1 meter).

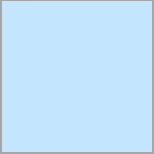
APPROXIMATION NOTE:   
The robot arm is modeled as a row of five cells that can also be diagonally aligned. We realize that the diagonal length is longer than the vertical or horizontal length, but we have decided to accept this approximation of the model.



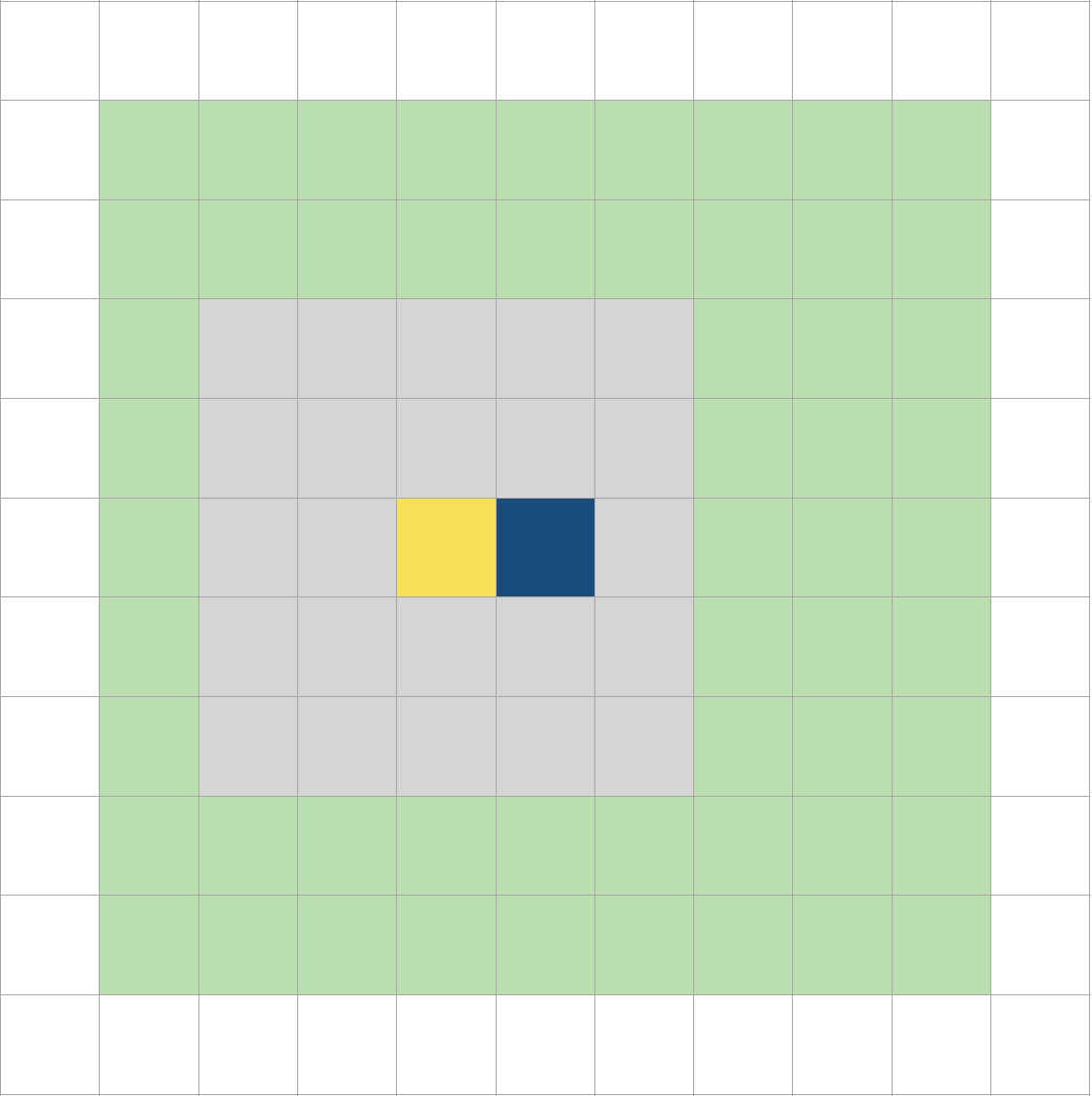


ArmBasePosition



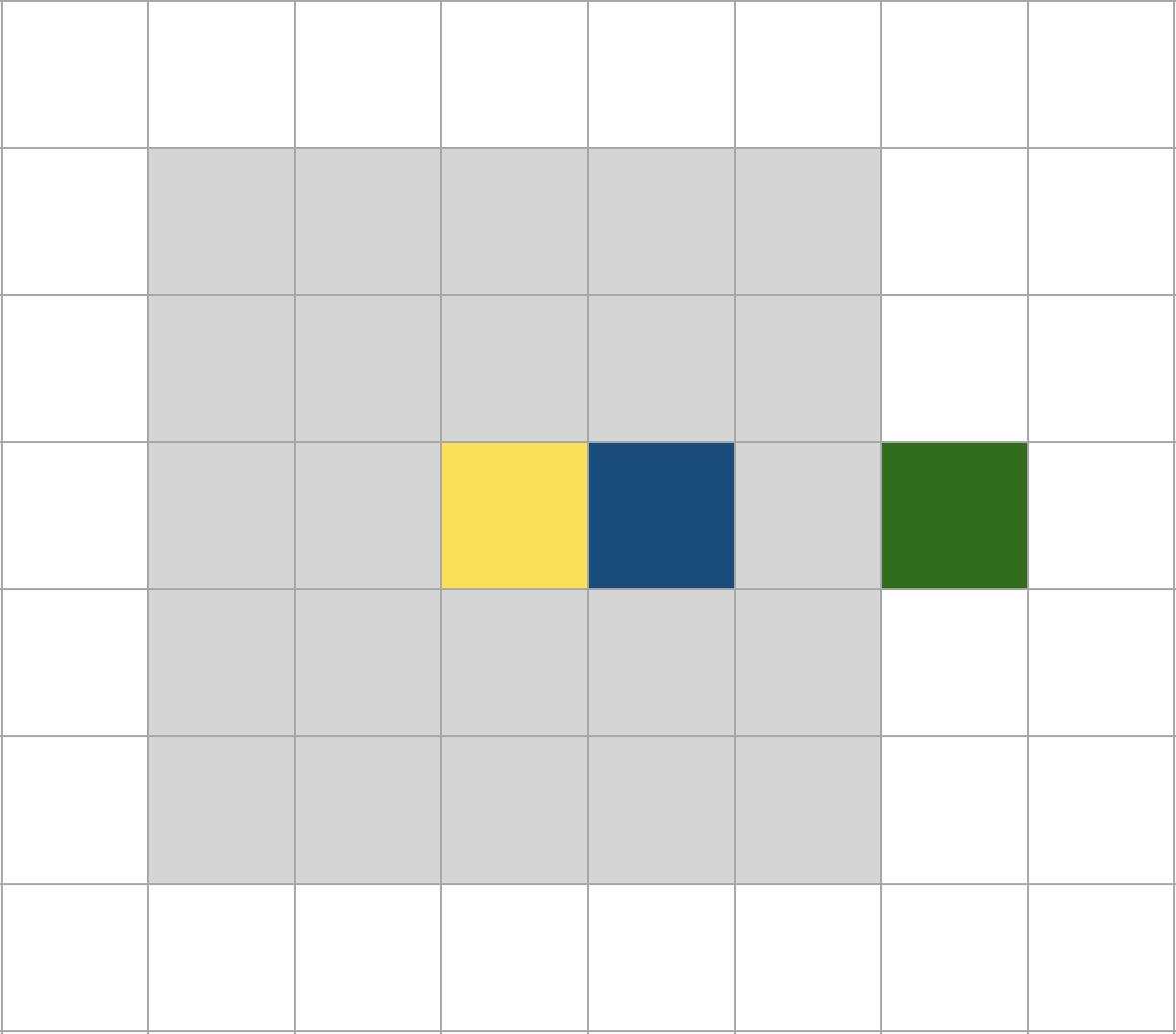
EEPosition

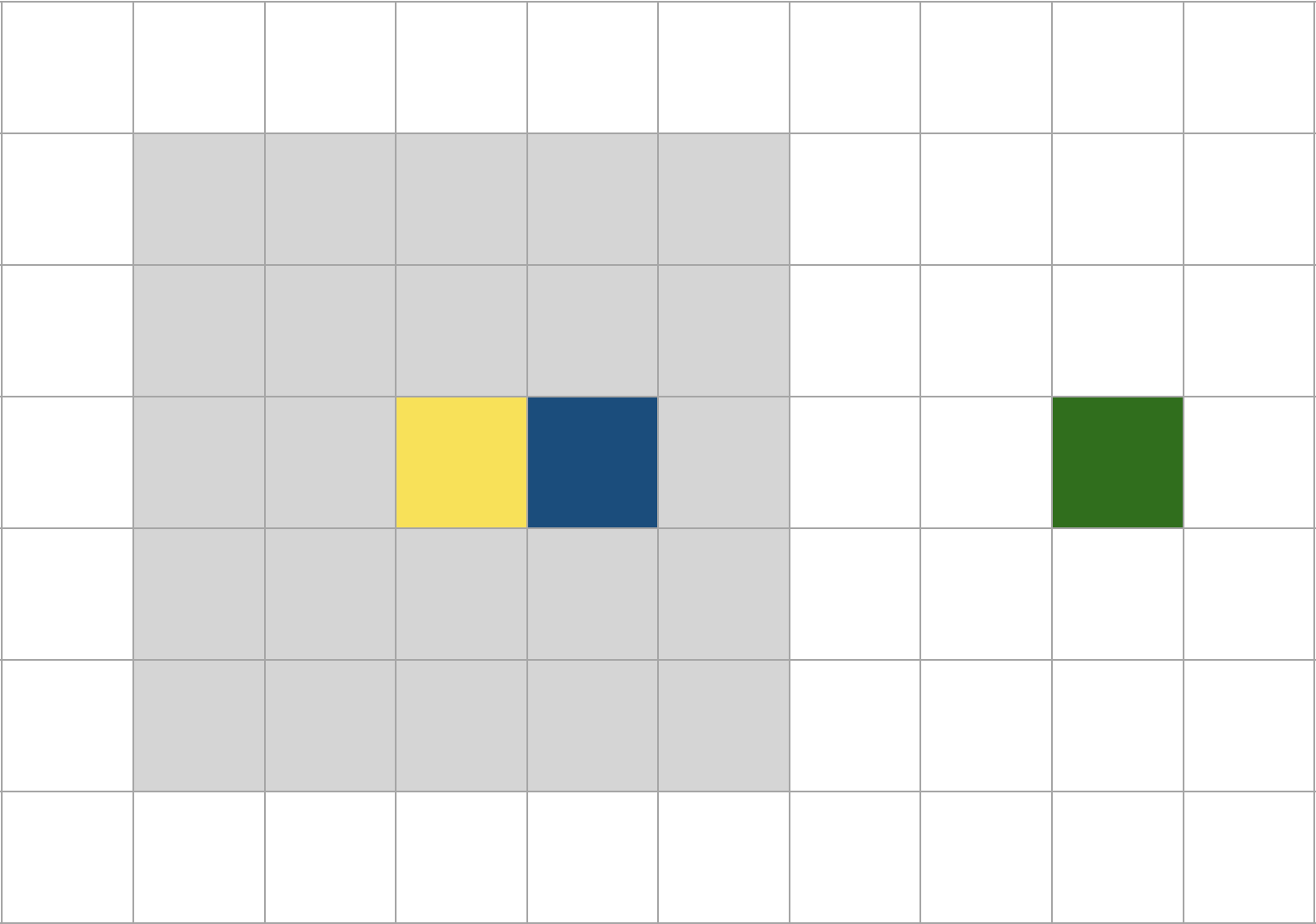
Area occupied by the arm (not detected by the system)



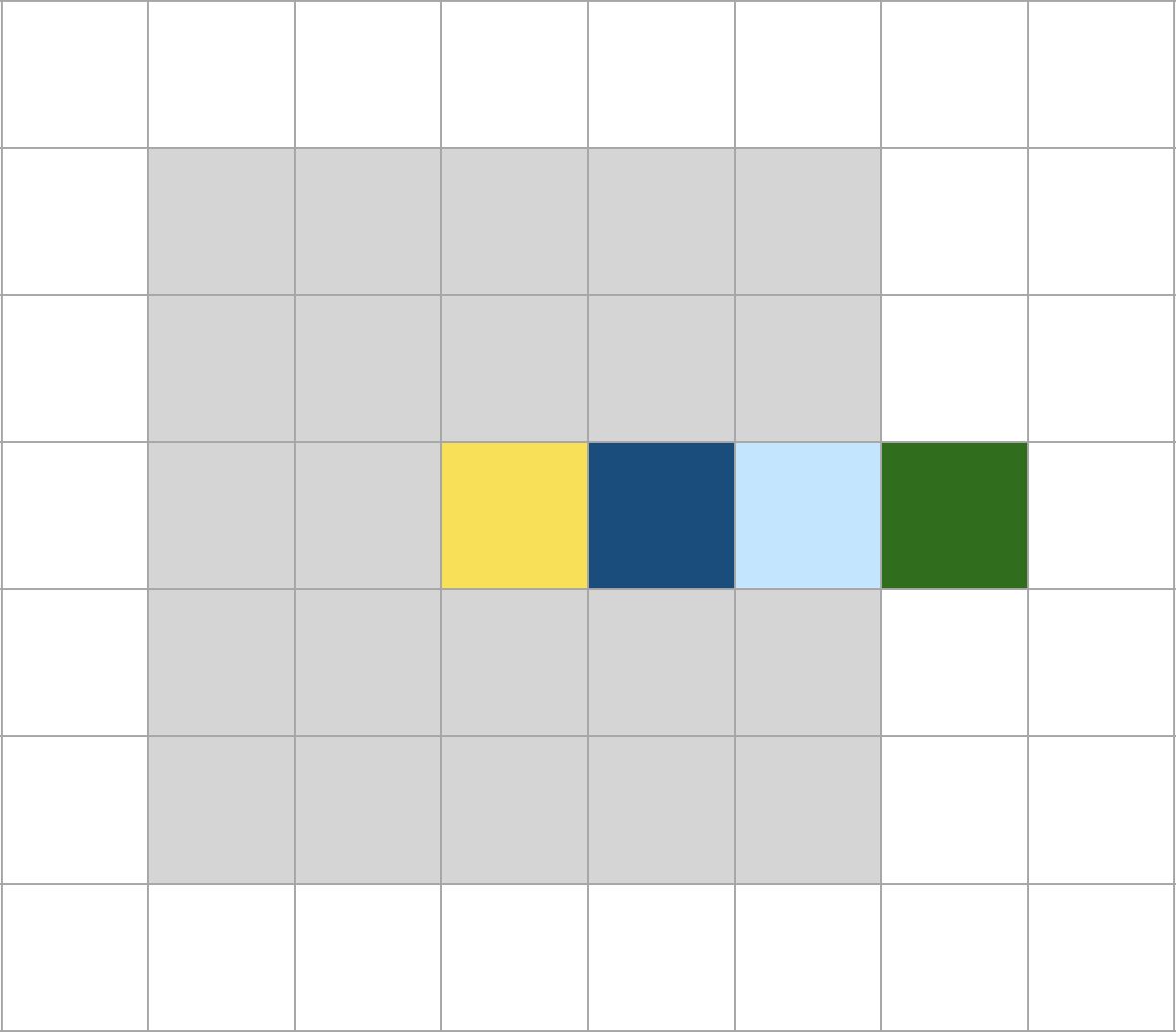
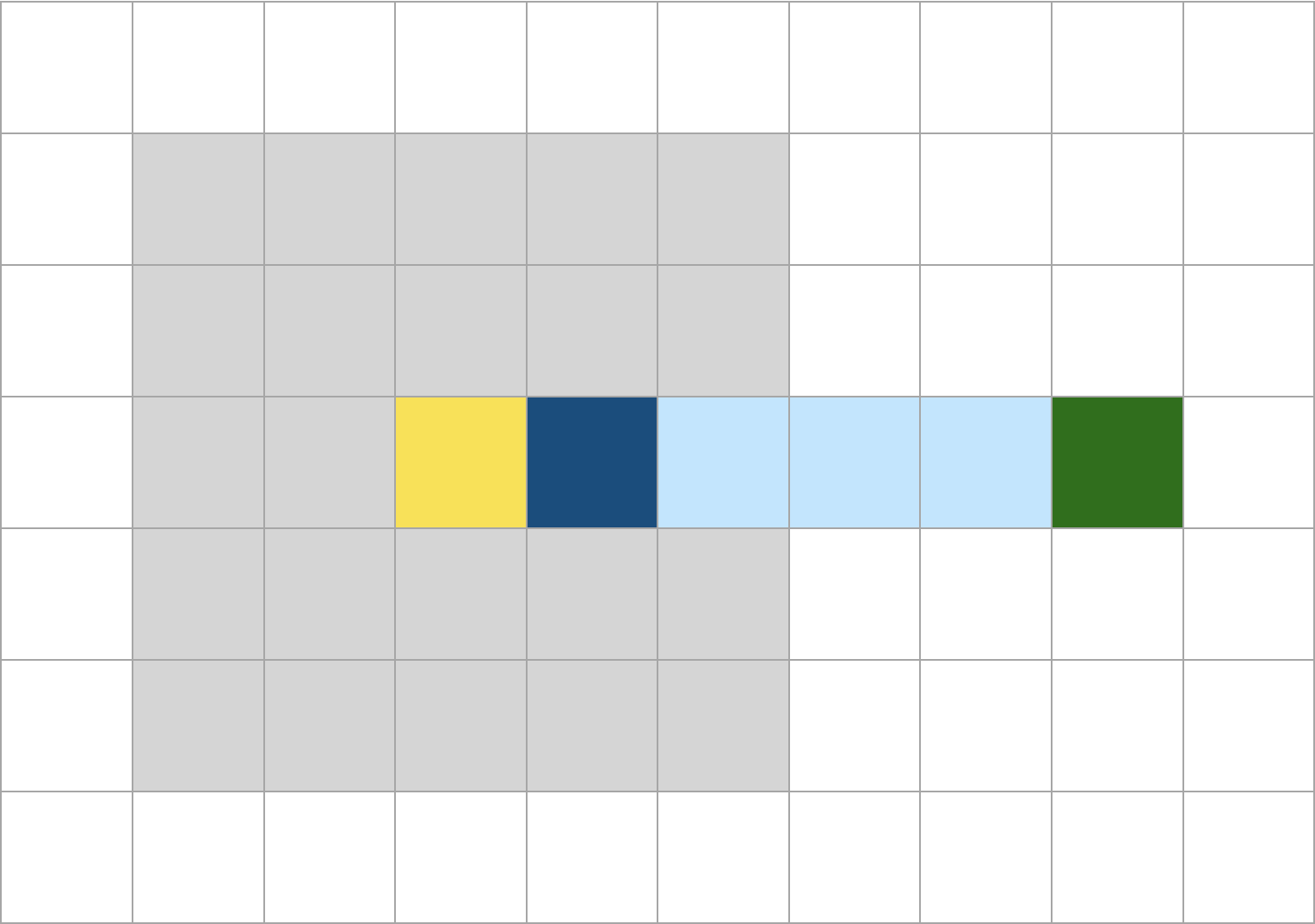
This is the full area (light green) in which the End Effector can move (the Robot’s Base is included), given the position of the ArmBase (dark blue).

The following two images show how the bending of the arm results in a contraction of the area occupied by it. Only detecting the End Effector is very effective in modelling this phenomenon in a simple but precise way.

What the system sees:



What we know is happening:



### ArmPosition()

The two cells represent, respectively, the positions of the ArmBase and of the EndEffector.

### There is Only One Arm

In the scenario of this project we only have one robot, therefore only one robot arm as well.

### ArmBase Movement

The ArmBase represents the point of attachment between robot and arm. To model the rotation of the robot, the arm base can change position around the robot’s BasePosition.

### EndEffector Movement

Here we describe the movement limitations that the EE must comply to when the robot is operating in “manipulator” mode.

Since the Arm Base is not going to move in manipulator mode, we just need to limit the relative distance of the End Effector with the respect to the base.

### EEFull

When the end effector is full, it is carrying a workpiece (WP).

Ps: it can only carry one WP at a time.

*What this is saying:*

If the end effector is full, then it should be carrying a WP (only one!). We did not use double implication because the EE could be just hovering above the WP without having picked it up.

### Arm in Rest Position

At any instant the robot can be operating in only one of the following two states:

* Mobile
* Manipulator.

When it is moving around (i.e. it is operating in the Mobile state), the arm should not protrude from the robot’s body and it should not be moving. We can model this by simply assuring that the EE is within the robot’s base area.

*What this is saying:*  
None of the cells occupied by the arm are at a distance greater than 2 from the robot central cell.

### ArmSpeed - Mode(Manipulator)

Here we define the speed of the arm when the robot is operating in “Manipulator mode”. Given the nature of the arm as seen from the model (straight long object, rotating with center of rotation on the ArmBasePosition) we know that the EE will be the fastest component during arm movement. Therefore we will only consider the EE for speed limitation.

;

;

;

NOTE:

You can see that “ArmSpeed(still)” is missing; this is because here we are describing the speed of the arm only for when the robot is operating in Manipulator mode. “ArmSpeed(still)” will be specified separately since it applies to both manipulator and mobile mode.

### ArmSpeed(still)

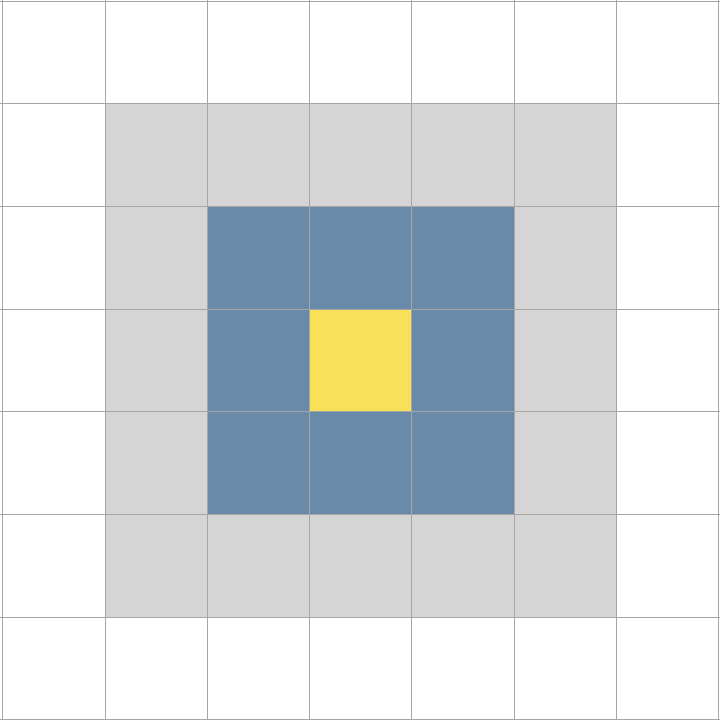
This predicate makes sure that, when the arm speed is “still”, the robot’s arm is not moving with respect to the robot’s body.

*What this is saying:*

The first part is saying that, at every time instant, all cells of the arm have moved exactly as the robot’s base has moved. The last clause is to take into account the case in which the robot is rotating.

### RobotRotation

This predicate models the rotation of the robot. This is done by detecting the movement of the ArmBasePosition around the robot’s central base cell.





BasePosition



Area around the BasePosition in which the ArmBase can move

*What this is saying:*

When rotating, the center of the base remains still since it is the center of rotation while the rest of the body rotates around it. We are not detecting the rest of the robot’s body but only the arm. Therefore, as the robot rotates, the arm will move with it and in particular, the arm’s base.

The “ArmBasePosition” axiom ensures that the new position of the arm base is still next to the robot’s base.

APPROXIMATION NOTE:

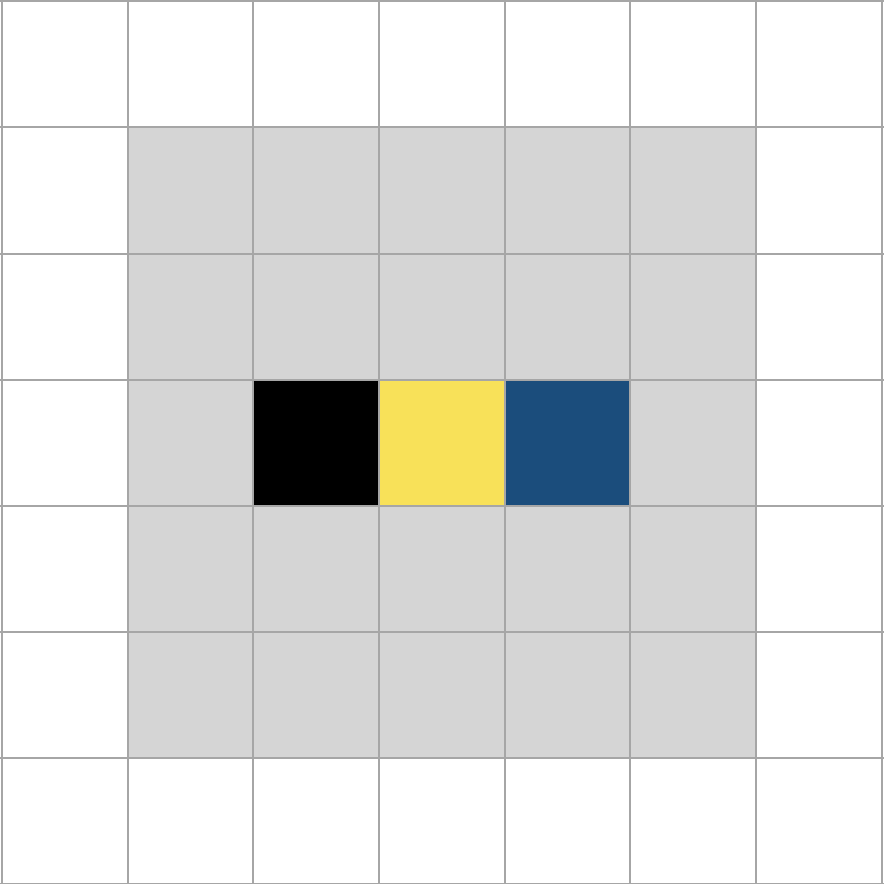
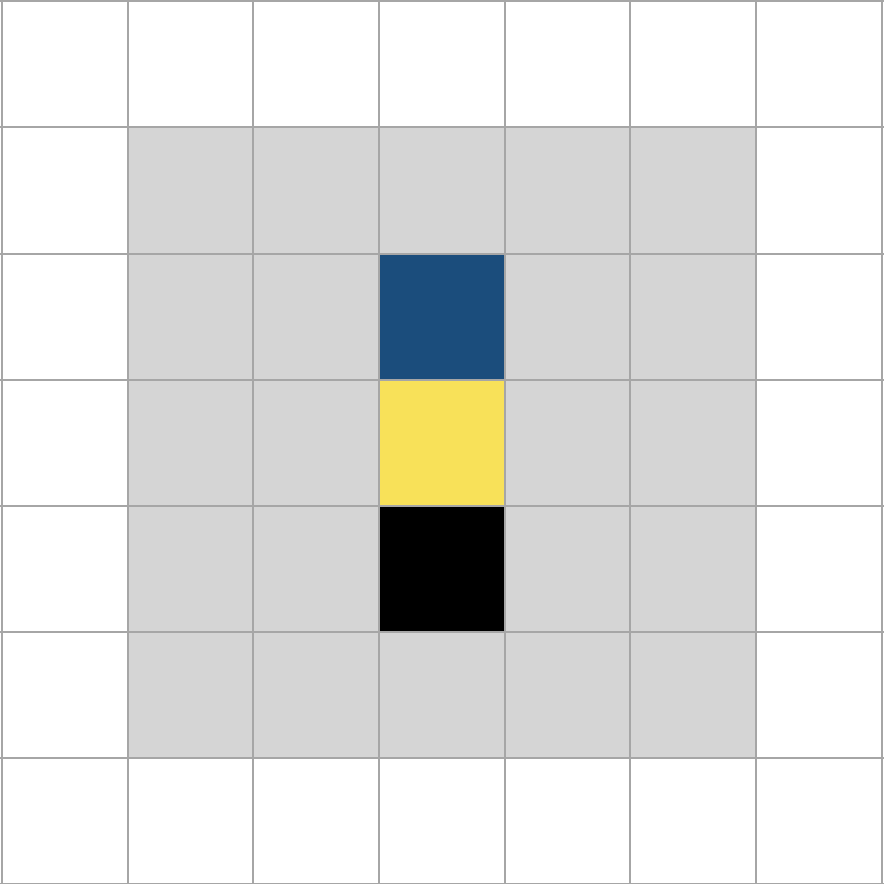
We are saying nothing on the behaviour of the EE in this situation. The EE will still be limited to stay within the bounds of the robot because it is in Mobile mode so we think this approximation will not add any particular risks.

## Bin

### BinPosition

We have decided to model the bin to have the dimensions of a single cell on the robot’s body. It is located on the side opposite to the arm base (the arm base is the first cell of link1). The motion of the arm base simulates rotation of the robot, so the bin will obviously rotate with it, remaining on the opposite side of the arm base as it moves around the robot base’s center.

In the following images we can see and example of how the bin follows the arm base symmetrically.





BasePosition



ArmBasePosition

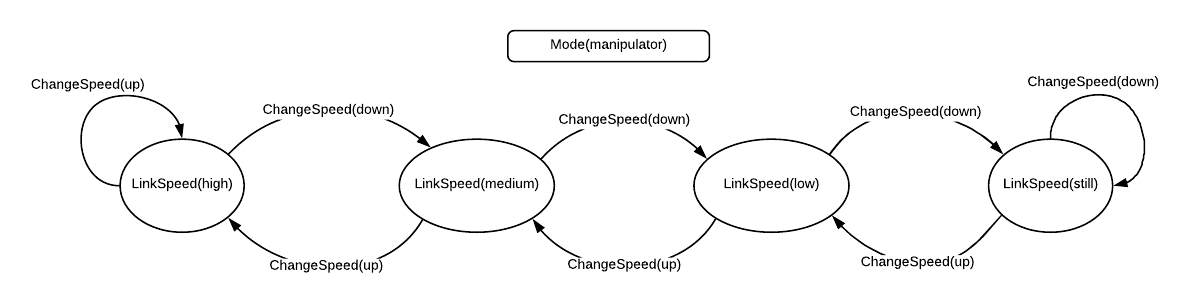


BinPosition

### There is Only one Bin

### Arm Speed Change

These axioms model the change of speed in the arm’s movements.

[](https://www.lucidchart.com/documents/edit/1bf36cf4-9790-4347-8c97-ab4e171d4f20/3?callback=close&name=docs&callback_type=back&v=2069&s=595.2755905511812)

## Operator

The operator is composed of a body (1 cell), a head (1 cell) and two arms (can occupy 1 to 3 aligned cells).

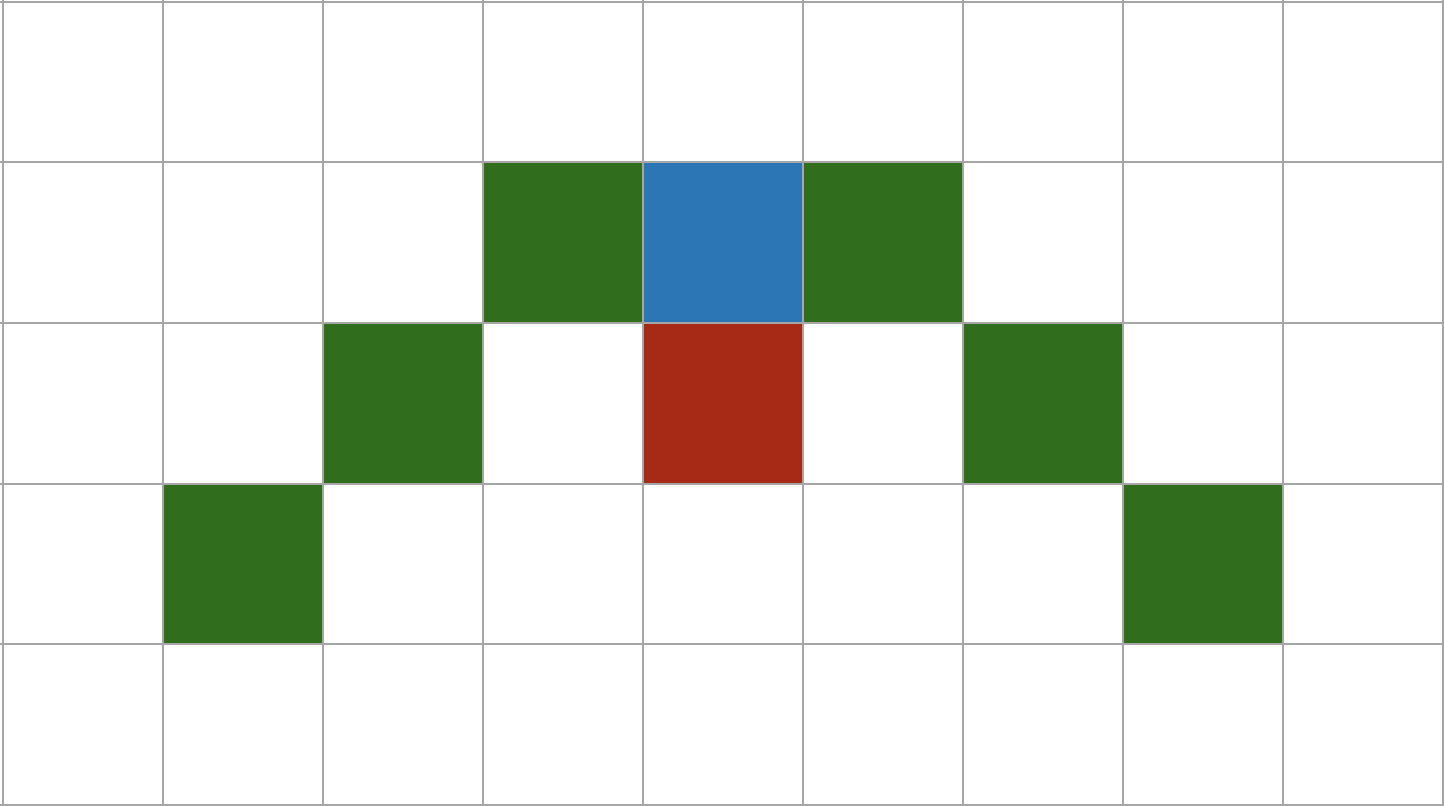
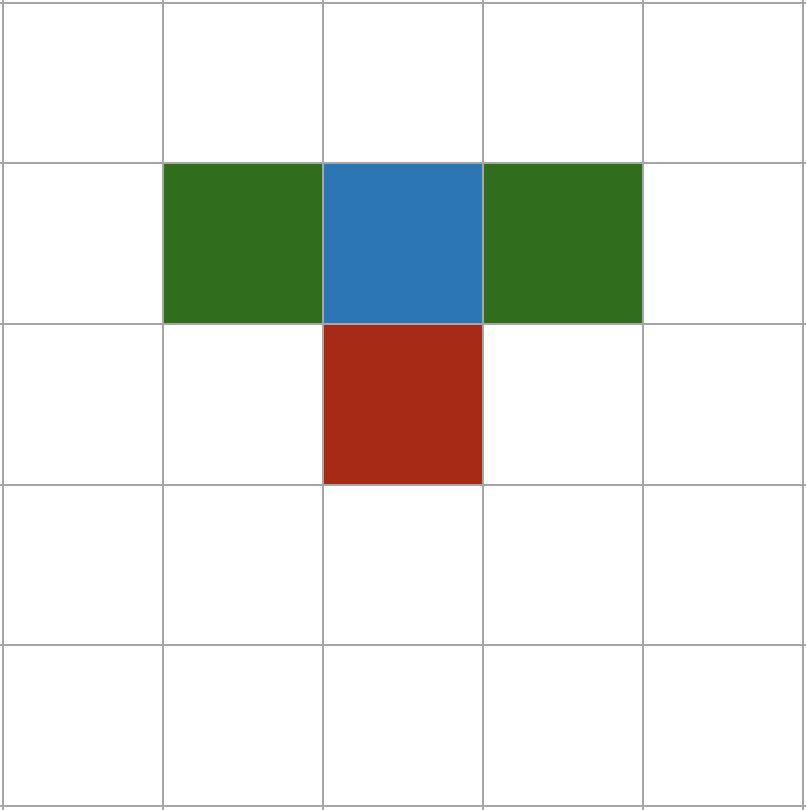
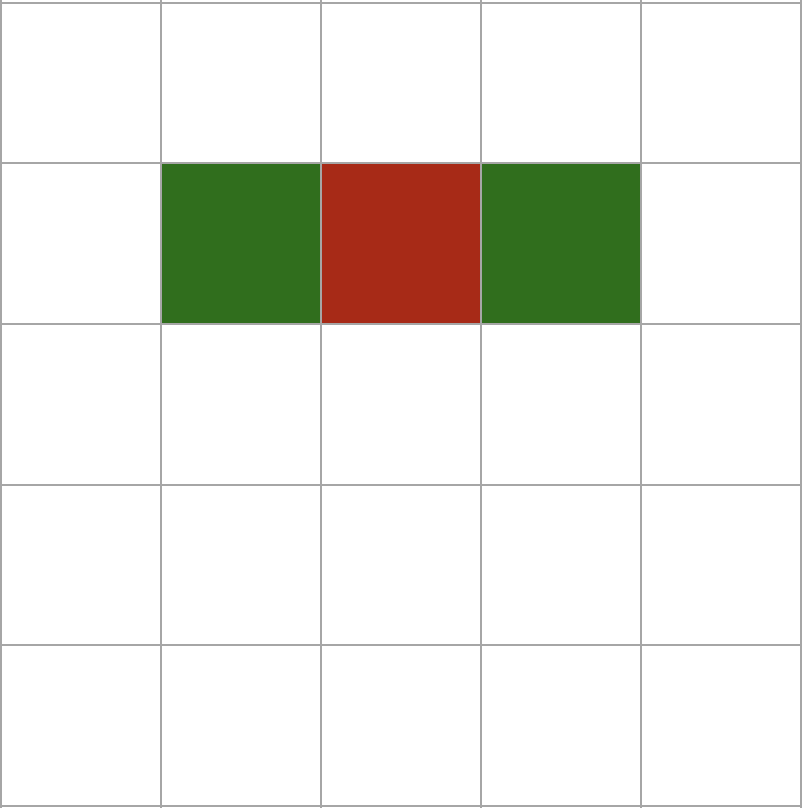
The head can be directly over the body or next to it.

The arms are attached to the body and therefore the first cell they occupy is always next to the body.

The three images represent (from left to right), respectively, the operator (looking toward south):

* Standing straight with arms along waist
* Leaning forward
* Leaning forward with arms fully extended forward

(many more positions are possible, these are just a few relevant examples)





Head Body Arm(s)

### Operator Position

This predicate describes the structure of an operator.

### Cells Occupied by a HumanArm are Connected

Here we ensure that an operator’s arm occupies at most 3 connected cells.

## External Signals

Mettere i segnali del tipo: “pick up WP”. in questo caso la condizione sarebbe notEEfull e che l’EEPosition sia la stessa del WP.

lo stato empty/full puo’ cambiare solo in certi istanti:

empty->full solo se sono sopra un WP

full->empty solo sopra il local bin

Il cambio di stato da operator a mobile deve assicurarsi che il braccio sia nella rest

Si potrebbe fare un po’ come una rete di petri dove i token sono i segnali:

Tipo per iniziare a muoversi aspetta vari segnali tipo bin full, arm in rest position, safeto move

### Signal

This is the generic signal. It allows the robot to realise there is an exception to be handled and interrupt its routine in order to manage the issue recognized by the signal.

### CloseTo Signals

BaseCloseToBody, BaseCloseToHead, BaseCloseToArms, FarFromBody, FarFromHead, FarFromArm are states.

For the Proximity Signals we use the signals state “CloseTo” which becomes true when the Operator is within a certain distance area determined by .

This parameter depends on the BaseSpeed and on the ChangeSpeed parameter, so it is dynamic and it is central for determining the safety property.

The CloseTo signals raise the ChangeSpeed(down) event in order to avoid dangerous contacts.

If there are no CloseTo signals the Robot tries to move/work with the fastest speed possible

### BaseCloseToBody

Alw(

When BaseCloseToBody is true, the robot slows down until its speed is “low” (obviously if the robot is “still”, the signal does not arise).

N.B.: ChangeSpeed is an event, so we must ensure that the robot receives this command every instants, otherwise there could be a conflict in the way the Robot slows down or speeds up.

### BaseCloseToArms

When BaseCloseToHead is true, the robot slows down.

### BaseCloseToHead

When BaseCloseToHead is true, the robot slows down.

### EECloseToBody

### EECloseToArms

### EECloseToHead

### ArmBaseCloseToBody

When BaseCloseToBody is true, the robot slows down..

### ArmBaseCloseToArms

When EECloseToArms or ArmBseCloseToArms are true, the robot slows down.

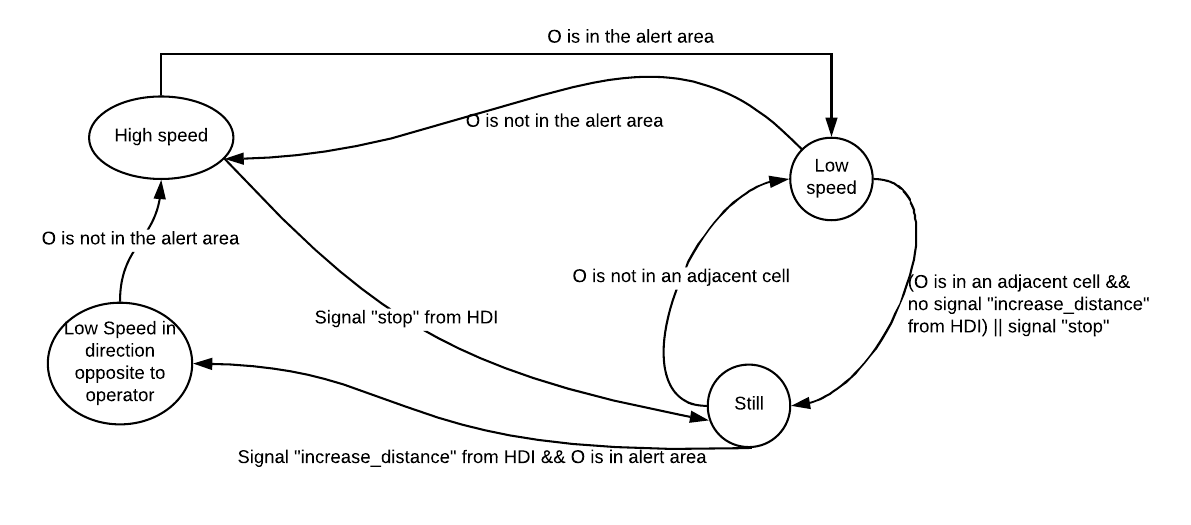
### ArmBaseCloseToHead

When EECloseToHead or ArmBaseCloseToHead is true, the robot slows down.

## Robot-Layout Interaction

### Robot Body Always in Blue Area

### Base Speed

[](https://www.lucidchart.com/documents/edit/2e8b033b-0667-4496-8d8e-2c38c0597433/0?callback=close&name=docs&callback_type=back&v=741&s=595.4399999999999)

These three axioms model the movement of the robot. They differentiate between three different speeds checking that the robots movement are coherent with the speed it should be moving at.

APPROXIMATION NOTE

We are assuming that if the system detects the robots base in a cell and in the following instant it detects it occupying another cell, then in the time in between the two instants, the robot has moved along the shortest path connecting the two cells.

### Wall’s Constraint

The position of the robot’s base has to fulfill the physical constraints given by the walls.

### Operating Mode - Mode(mobile)

This predicate tells us that, when located in the blue area outside of the LoadArea or the ReleaseArea, it should be operating in “mobile” stateType. This means that only the robot’s body is functioning (to move around), while the arm is in the rest position to improve safeness of movements.

### Operating Mode - Mode(manipulator)

At any instant the robot can be operating in only one of the following two states: Mobile, Manipulator. When it is in the manipulator mode, the robot’s base should be still and the arm should be operating on a task (picking up or putting down the WPs).

## Robot-Operator Interaction

### FSA

We have four states that describes the robot’s behavior

### Base Close to Operator

Distance(base, operator, d)

### KeepDistance

se un operatore e’ vicino, il robot non puo’ avvicinarsi (puo’ solo allontanarsi) (tipo “if operator has not moved and I moved, then distance has increased”)

### DontMoveWhenDistanceIsOne

se il robot e’ affianco all’operatore allora e’ “still”

### SlowInWorkStation

Se il robot e’ nell’area workstation o bin area allora la velocita’ non e’ ne’ media ne’ veloce

### Robot does not entrap the operator

These axioms model the behavior of the robot when it is near the operator and near the wall, the robot has to avoid the situation in which the operator is entrapped between the robot and the wall.

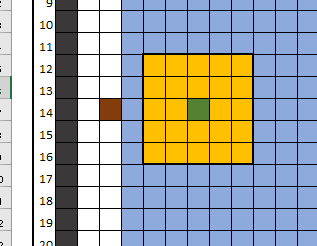
If the operator is between the robot and the wall and the distance of the robot from the wall is less than 6 cells (and so there are 5 cells between the center of the robot and the wall) then the robot should not decrease its distance from the operator (so it must stop or move farther away from the operator).

are, respectively:

* The distance between the wall and the operator
* The initial distance between the robot and the wall
* The initial distance between the robot and the operator
* The succeeding distance between the robot and the operator in the case in which the robot moves in order to move away from the operator.

are, respectively:

* A cell of the wall
* The two cells in which the operator is located (before and after)
* The two cells in which the robot is located (before and after).



### Robot Workflow

### Destination of the robot in mobile mode

At every instant, in absence of signals, the robot is working or it is moving between the LoadArea and the ReleaseArea to perform the next operation.

These axioms model the destination of the robot (when it is moving in mobile mode), relying of the last area in which the robot has worked (ReleaseArea or LoadArea).

If, up to now, the robot is in mobile mode and its last position in a working area was in the LoadArea then it is moving to the ReleaseArea without returning in the LoadArea in the meantime; the robot’s bin has to be not empty too, otherwise it has no sense to move to the ReleaseArea.

This is the viceversa of the previous axiom.

### Working in Manipulator Mode

### Operation Timing

At every instant, after beginning a certain operation (either pickWP or putWP), the robot will continue performing the same operation or will change its mode from manipulator to mobile but only in a discrete way

### Working(pickWP/putWP)

The “Working(operation)” predicates states that the Robot remained still in the correct Area (without having BinFull or Signal events) since reached the correct position itself for that specific operation (the LoadArea for pickWP and the ReleaseArea for putWP)

### Ending PickWP/PutWP

The Robot will continue picking up/putting down WPs until the bin becomes full/empty (or a special signal force it), then it will move its arm in the rest postìition

### Working Procedure

Every time the robot adds/removes a WP into/from the bin and it has not finish the entire operation (so the bin it is not already Full/Empty) it will continue the procedure, if there are no signals that stop it in the meanwhile

# Safety Property

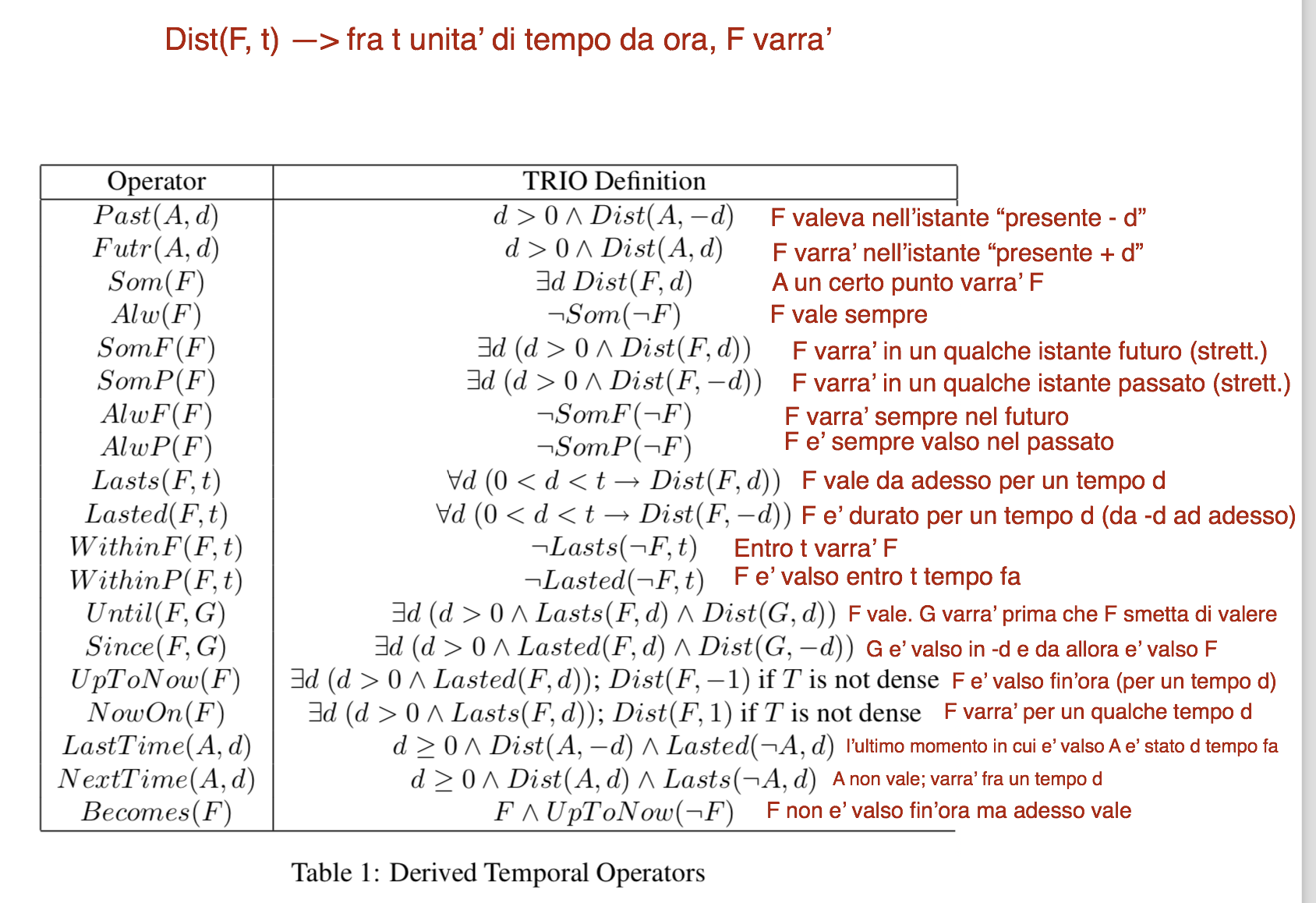
The First Safety Property requires that the Robot’s Base does not come in contact with the operator while moving (if any contact occurs, BaseSpeed must be ”still”).

Dealing with Human Arms and Head the Safety Property is a little bit different because there is the possibility that they will be on the same cell of the Robot (the parameter d in the Distance predicates could be ). However, as for the body, the Robot must be still

The Second Safety Property is about the EESpeed and is more complex. Actually we want to ensure that if the body or the arms of the operator are close to the Robot Arm, the latter will continue to move but with slow speed. On the other hand, if the Human Head is too close, the Robot Arm will be still. The so call “secure area” that we want to create around the Robot Arm is composed by one square of dimension 5 (of ray ) around the EE and another one of dimension 3 (of ray ) around the Arm Base.

In order to achieve these fundamental properties in our model we have to deal with the parametric values of With the former we characterized the way in which the Robot can change its speed, a sort of Reaction Time parameter. Ergo, it is possible to know how many cells the Robot will pass over

## Discussion of the safety property in the model



--------------------------------------------------------------------------------------------------

BOZZE

------------------------------------------------------------------------------------------------

-------------------------------------------------------------------

O

O

--------------------------

Dubbi:

->mandrioli ha detto:

per uso degli eventi nel progetto:

mettiamo un capitolo con scritto quali eventi usiamo : gli eventi sono “istantanei” (almeno dal punto di vista matematico)