INSTITUTO SUPERIOR TÉCNICO

PÓLO DO I.S.T

Stochastic Satisfiability Planning for Multi-Robot Systems

5th ISLab Workshop

Hugo Costelha
Institute For Systems and Robotics
Instituto Superior Técnico
Lisbon, Portugal

Summary

- Motivation
- Planning as SSat
- Application to Robotic Soccer
- Results and Discussion
- Conclusions & Future Work

Motivation

Done as a project for *Algoritmos para Lógica Computacional*.

Current (overall) goal

To be able to specify and execute a multi-robot task with predefined quantitative and qualitative properties.

Project goal

Test the applicability of SSat planning to multi-robot systems.

INSTITUTO
SUPERIOR
TÉCNICO

PÓLO DO I.S.T

Summary

- Motivation
- **Planning as SSat**
- Application to Robotic Soccer
- Results and Discussion
- Conclusions & Future Work

Satisfiability

Finding an assignment in a given boolean formula, in Conjunctive normal form, that makes it True.

$$x = \langle x_1, x_2, \ldots, x_n \rangle$$

$$\exists x_1, \exists x_2, \ldots, \exists x_n, (\phi(x) \Leftrightarrow True)$$

$$Ex.: \phi(x) = (x_1 \vee \bar{x}_2 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee x_5) \wedge (\bar{x}_1 \vee \bar{x}_3)$$

Stochastic Satisfiability

- Proposed by Littman in 1997
- New randomized quantifier: ᴚ^{π}
- Problem is now finding an assignment in a given boolean formula, in CNF, that **maximizes the probability** of being True.

$$x = \langle x_{1,} x_{2,} \ldots , x_n \rangle \rightarrow \exists \qquad \text{(choice variables)}$$

$$y = \langle y_{1,} y_{2,} \ldots , y_m \rangle \rightarrow \text{ᴚ}^{\pi} \qquad \text{(chance variables)}$$

$$\exists x_{1,} \text{ᴚ}^{\pi_1} y_{1,} \exists x_{2,} \text{ᴚ}^{\pi_2} y_{2,} \ldots , \exists x_n , \text{ᴚ}^{\pi_m} y_m , (E[\phi(x) \Leftrightarrow True])$$

Solving SSat Problems (1)

Consider an SSat problem Φ composed by a formula ϕ and quantifier order Q. Let $val(\phi,Q)$ denote the maximum probability of satisfaction of ϕ under ordering Q.

Solving SSat Problems (2)

$val(\phi, Q)$ is defined recursively using the following rules:

1. If ϕ contains an empty clause, then $val(\phi, Q) = 0.0$;
2. If ϕ contains no clauses, then $val(\phi, Q) = 1.0$;
3. If $Q(x_1) = \exists$, then

   $$val(\phi, Q) = max\left(val\left(\phi\lceil_{x_1=0}, Q\right), val\left(\phi\lceil_{x_1=1}, Q\right)\right);$$

4. If $Q(x_1) = \text{ꓤ}^{\pi}$, then;

$$val(\phi, Q) = \left(val\left(\phi\lceil_{x_1=0}, Q\right) * (1.0 - \pi) + val\left(\phi\lceil_{x_1=1}, Q\right) * \pi\right).$$

Solving SSat Problems (3)

Additionally to these rules we have:

- Unit propagation
- Pure variable elimination
- Splitting and threshold pruning
- Memoizing

INSTITUTO SUPERIOR TÉCNICO

Representing Probabilistic Planning Problems

Use of Sequential-effects-tree (ST) representation to describe actions.

The STs are described textually in a *Probabilistic Planning Language* (*PPL*) in the following form:

$$a \text{ causes } \pi \text{ withp } p \text{ if } c_1 \text{ and ... and } c_m$$

Additional domain specifications (including action-preconditions) are also specified directly in the *PPL* file.

PÓLO DO I.S.T

INSTITUTO SUPERIOR TÉCNICO

PÓLO DO I.S.T

Encoding Probabilistic Planning Problems as Ssat Problems
-Variables & Clauses-

- All actions and propositions are indexed for all time steps;
- Initial and final conditions;
- Exactly-one-of actions clauses;
- One clause for each path in the ST;

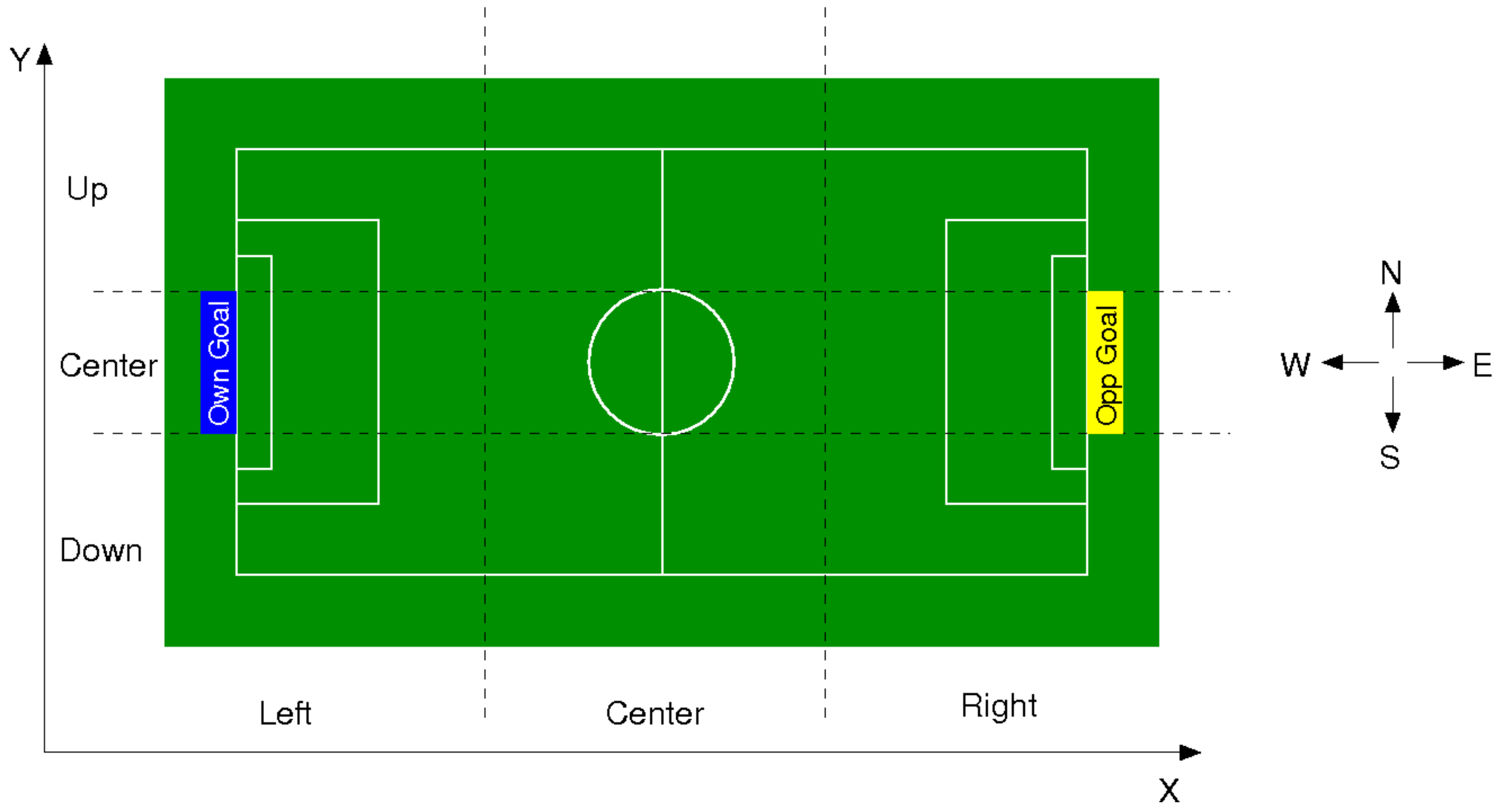
Summary

- Motivation
- Planning as SSat
- **Application to Robotic Soccer**
- Results and Discussion
- Conclusions & Future Work

Application to Robotic Soccer

- Tested with a 2 player team vs a static opponent;
- Considered impossible to regain the ball possession once captured by the opponent;
- The goal was to score;
- Started with a complete deterministic case and then extended to the probabilistic case;

The Environment



The Sensors

➔ Robots and ball position (ex.: P1_X_L, P2_Y_C, BALL_X_R);

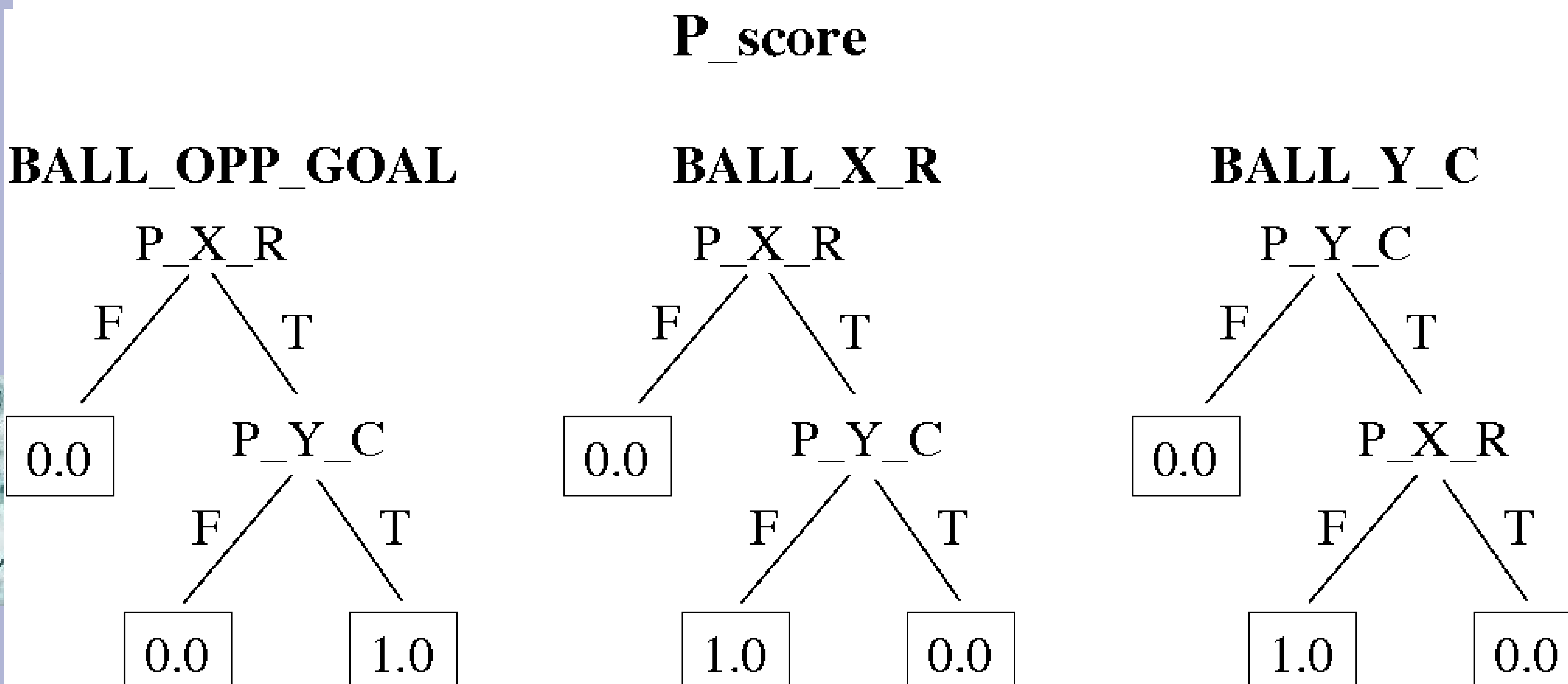
➔*Has_Ball* (for each robot);

The Actions

- moveN, MoveS, moveE, moveW;
- takeBallN, takeBallS, takeBallE, takeBallW;
- pass;
- score;
- standBy.

The score Action – deterministic case

Sequential-effects-tree



The score Action – deterministic case
PPL description

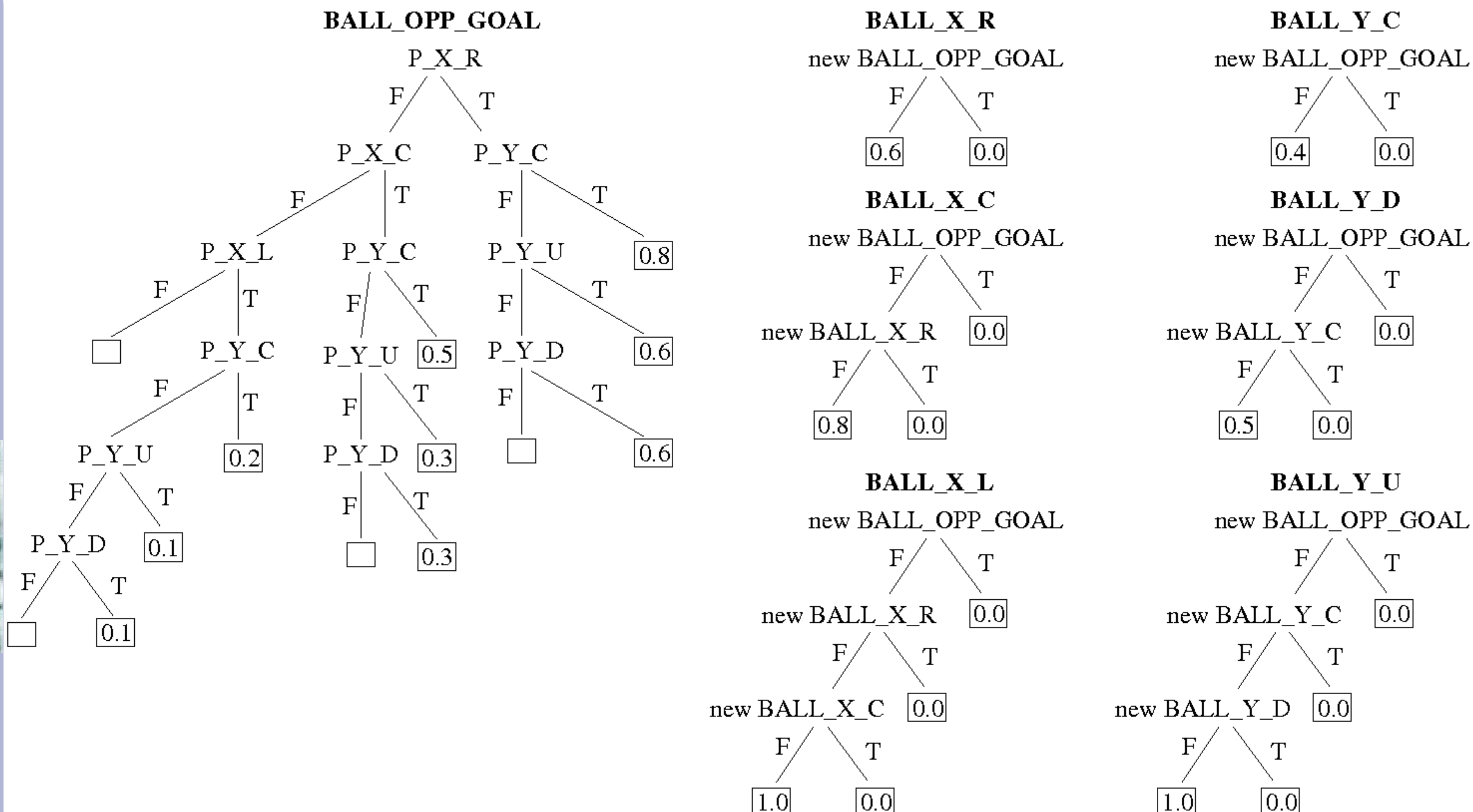
P_score causes -BALL_OPP_GOAL withp 1.0 if -P_X_R
P_score causes -BALL_OPP_GOAL withp 1.0 if P_X_R and -P_Y_C
P_score causes BALL_OPP_GOAL withp 1.0 if P_X_R and P_Y_C
P_score causes -BALL_X_R withp 1.0 if -P_X_R
P_score causes BALL_X_R withp 1.0 if P_X_R and -P_Y_C
P_score causes -BALL_X_R withp 1.0 if P_X_R and P_Y_C
P_score causes -BALL_Y_C withp 1.0 if -P_X_R
P_score causes BALL_Y_C withp 1.0 if -P_X_R and P_Y_C
P_score causes -BALL_Y_C withp 1.0 if P_X_R and P_Y_C

- Pre-conditions:

impossible P_score if -P_HAS_BALL

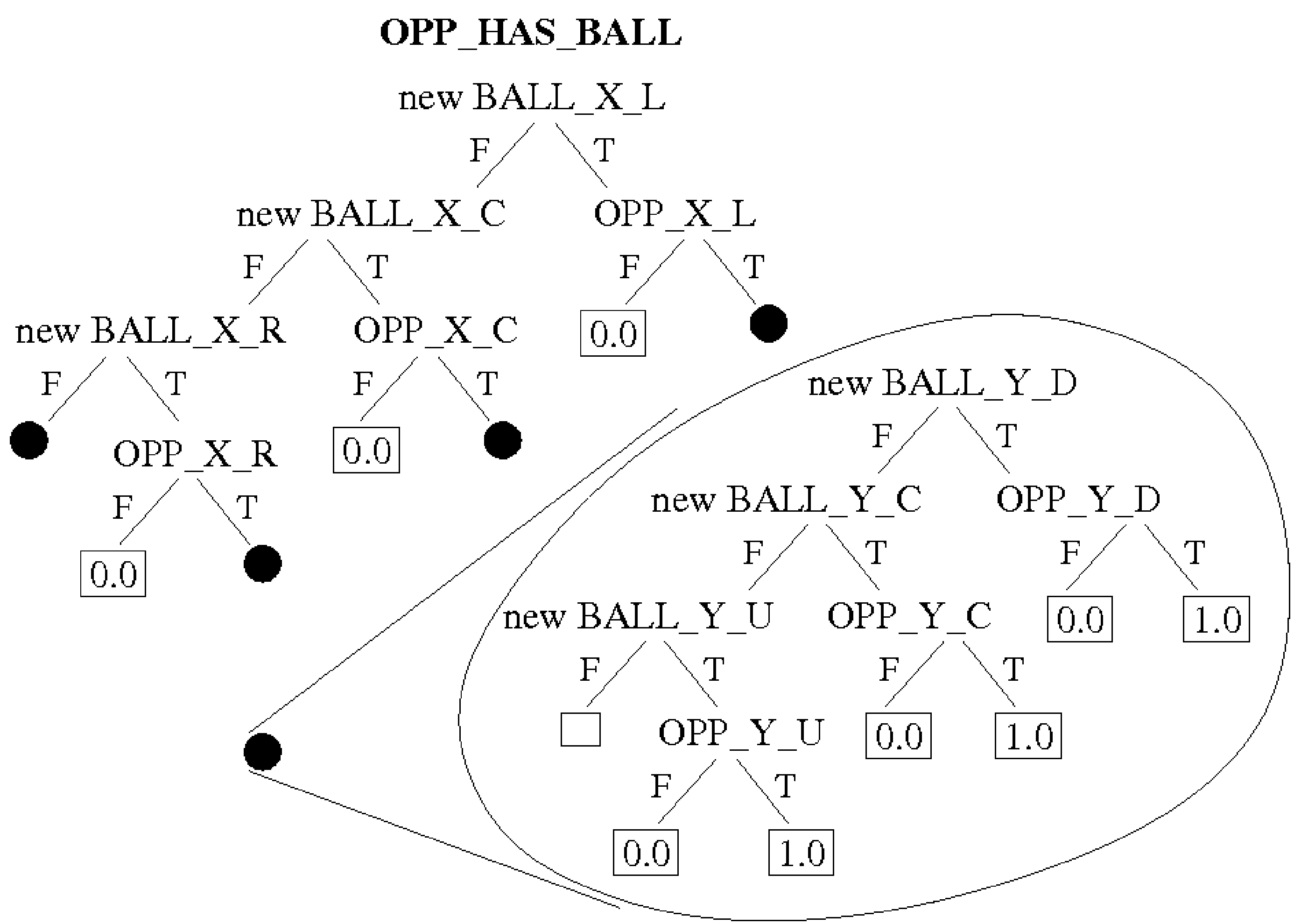
The score Action – probabilistic case

Sequential-effects-tree (1)



The score Action – probabilistic case

Sequential-effects-tree (2)

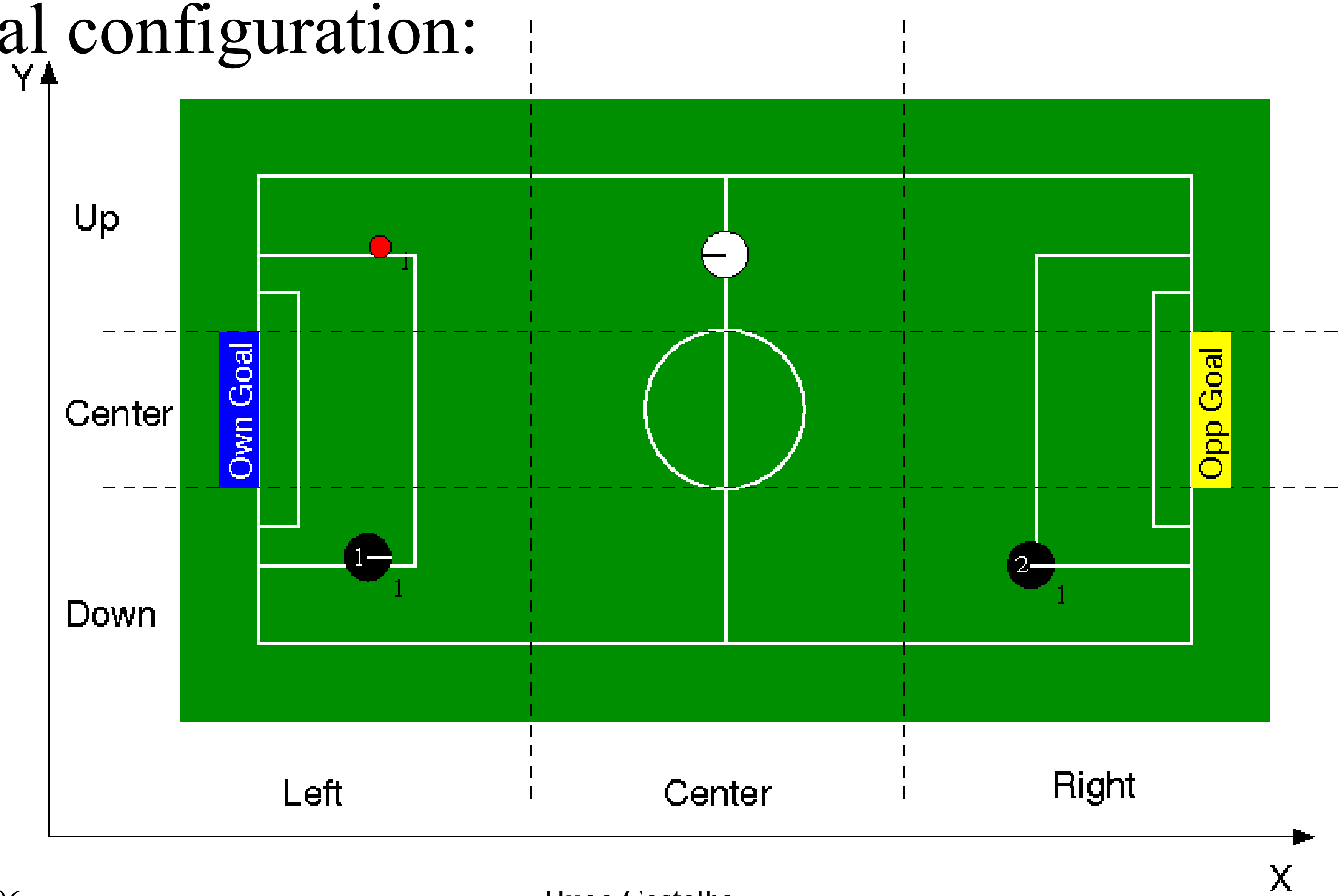


Summary

- Motivation
- Planning as SSat
- Application to Robotic Soccer
- **Results and Discussion**
- Conclusions & Future Work

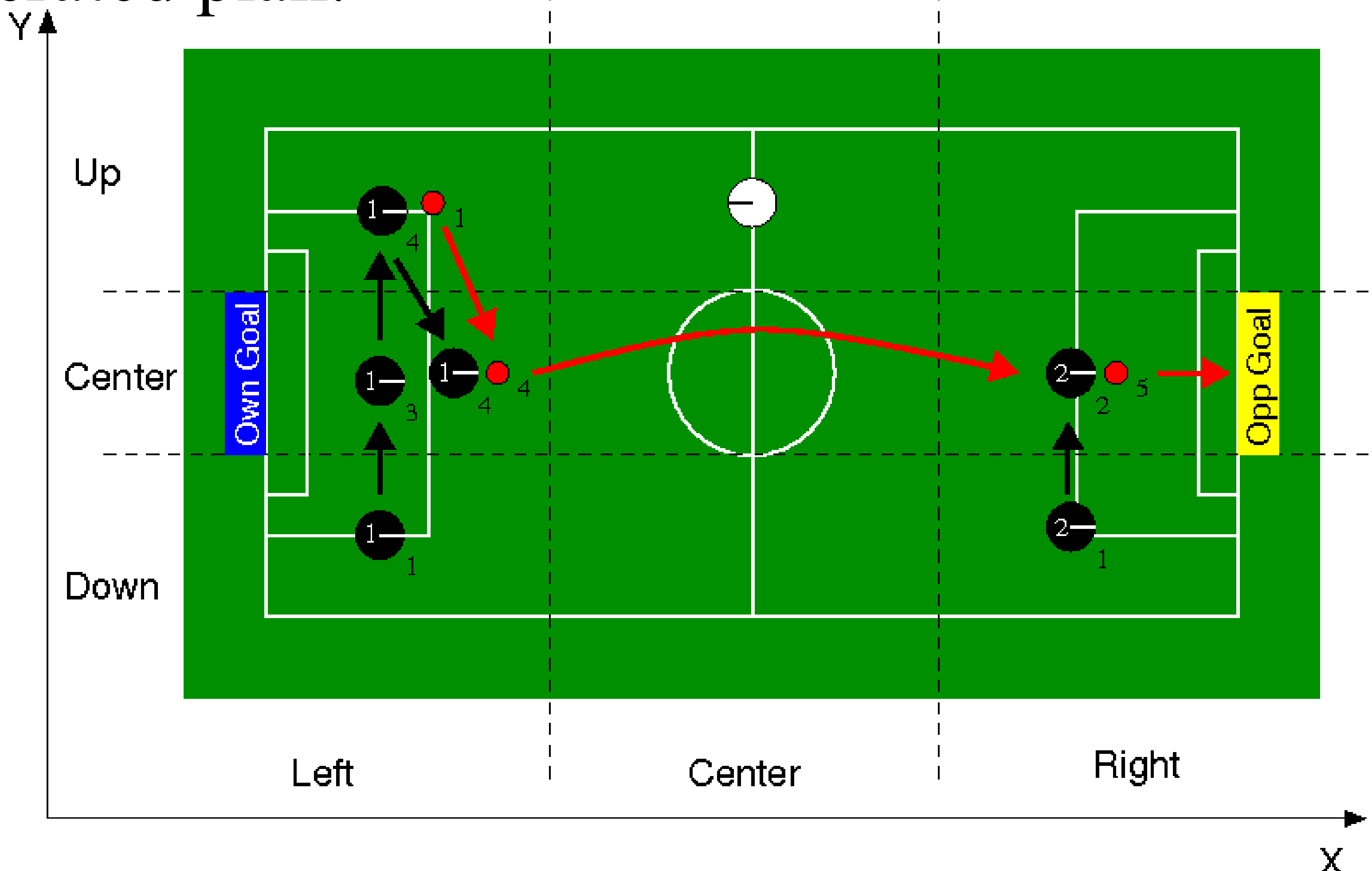
Deterministic Environment
Case 1

Initial configuration:



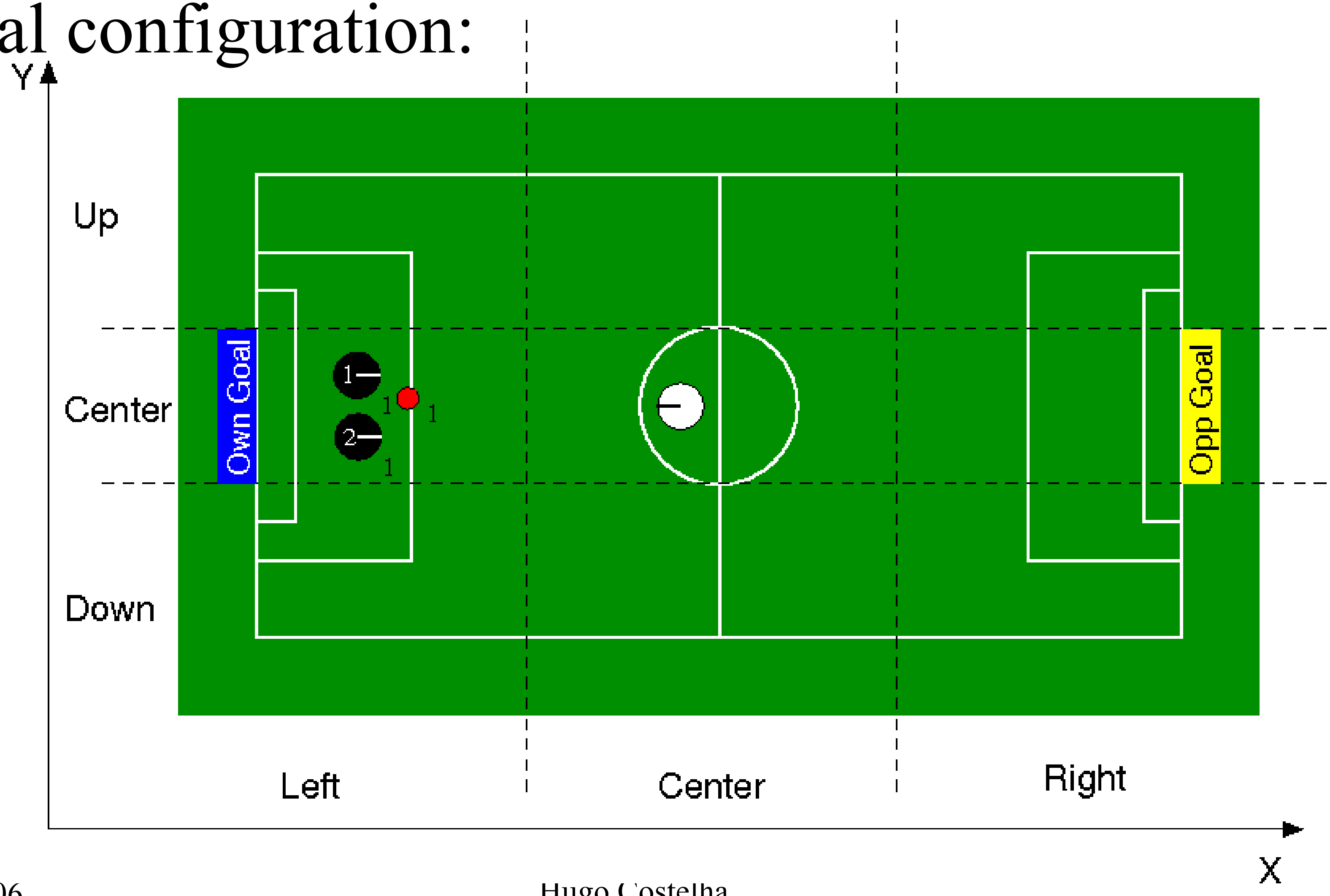
Deterministic Environment
Case 1

Generated plan:



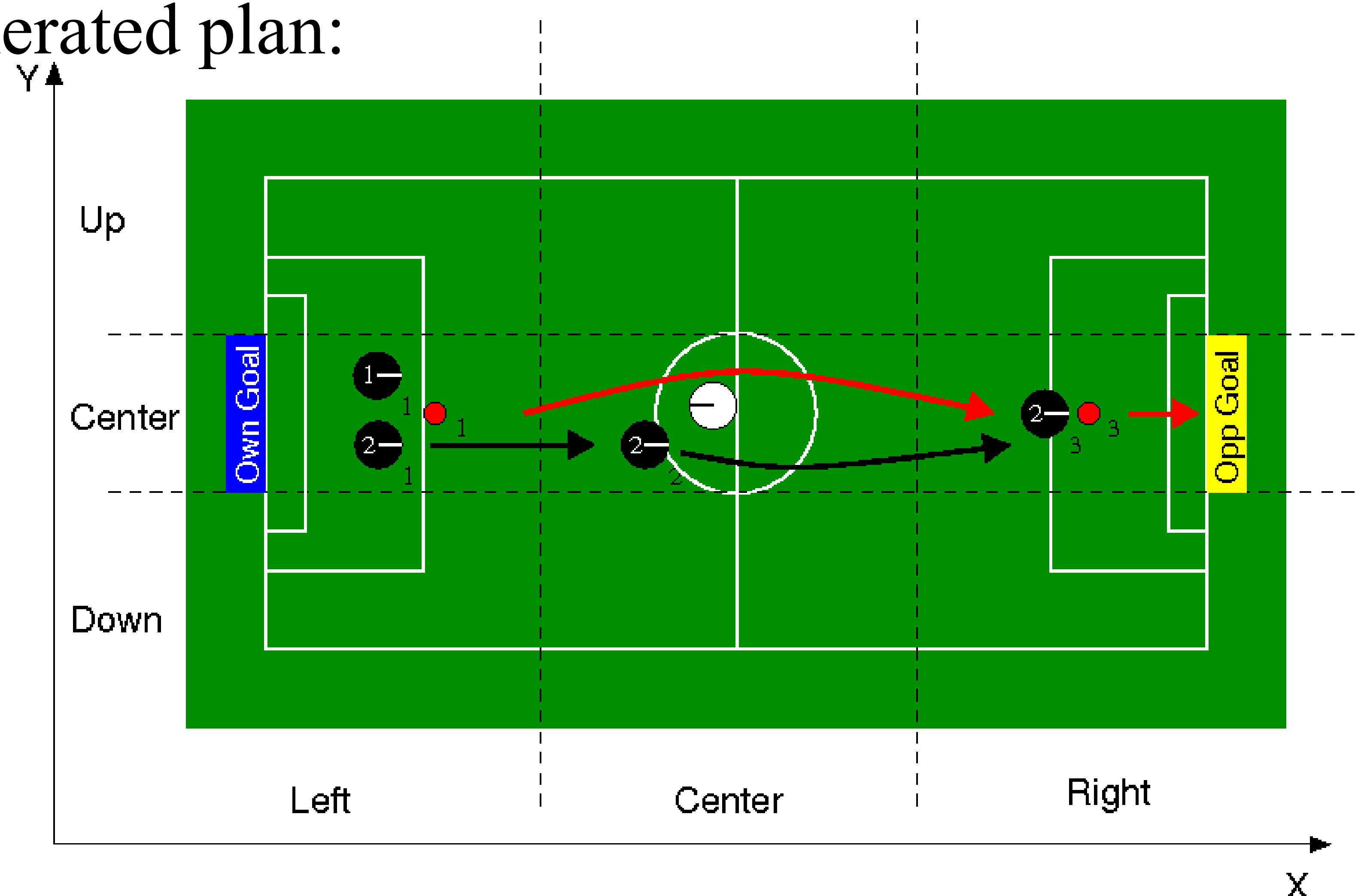
Deterministic Environment
Case 2

Initial configuration:



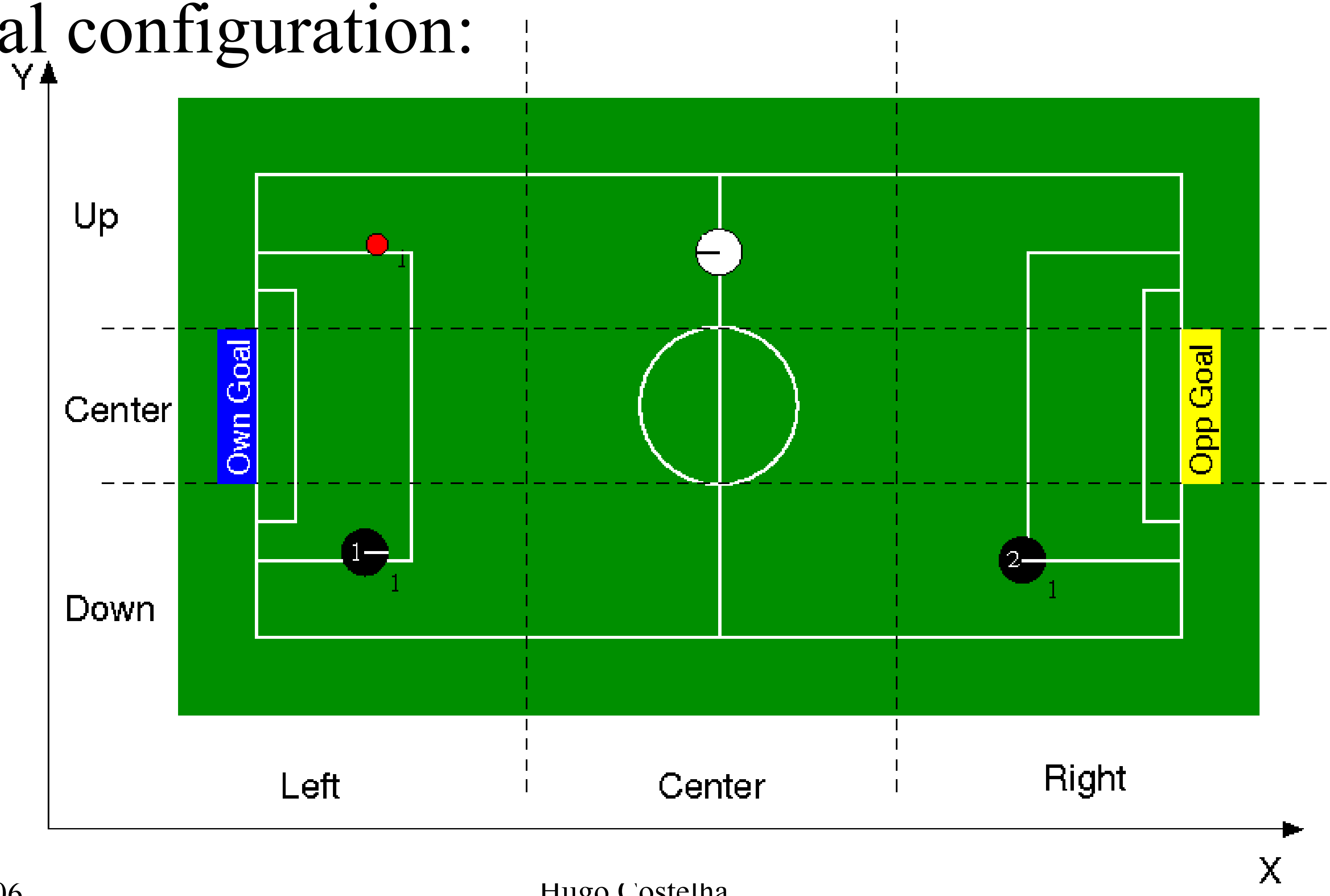
Deterministic Environment
Case 2

Generated plan:



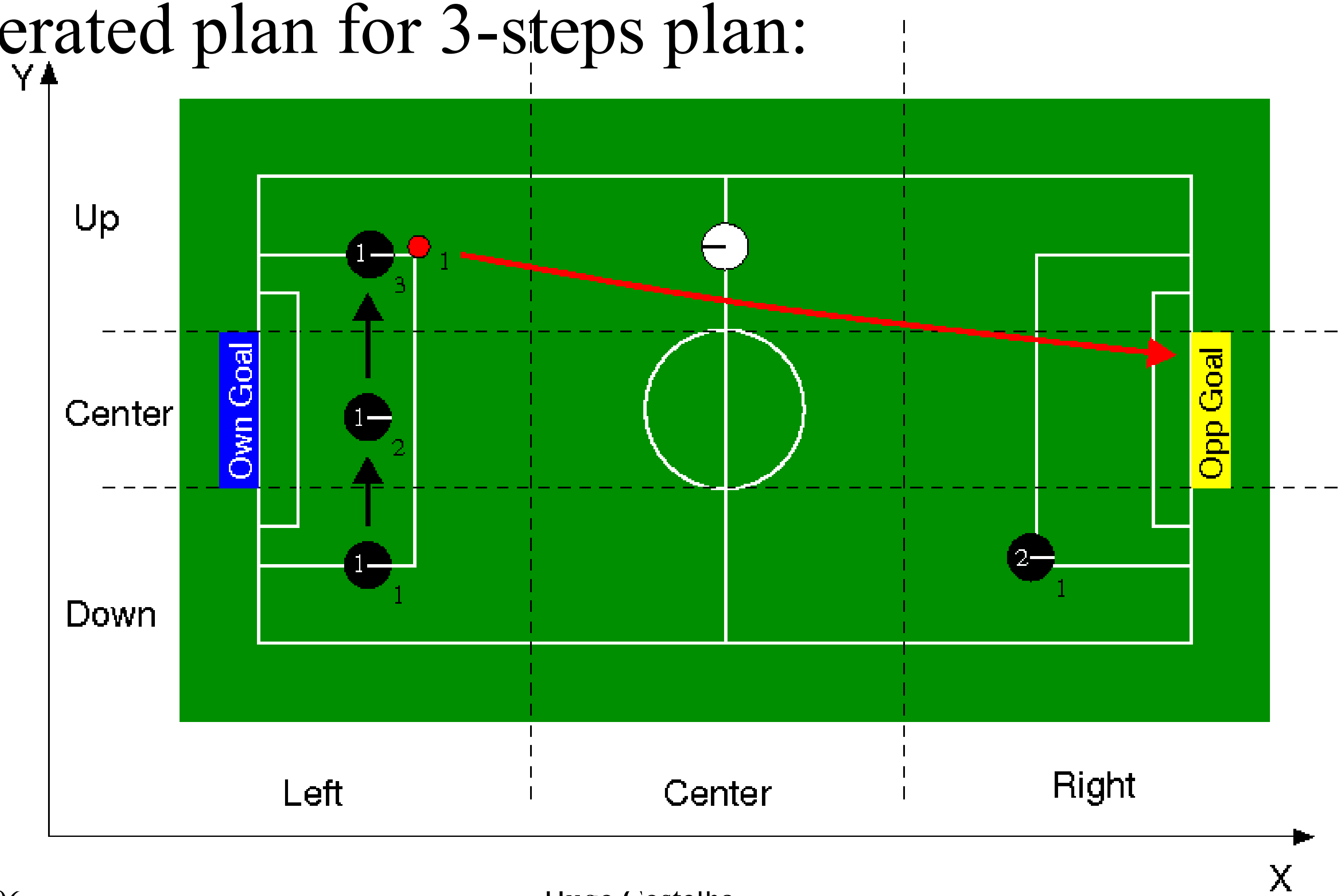
Probabilistic Environment

Initial configuration:



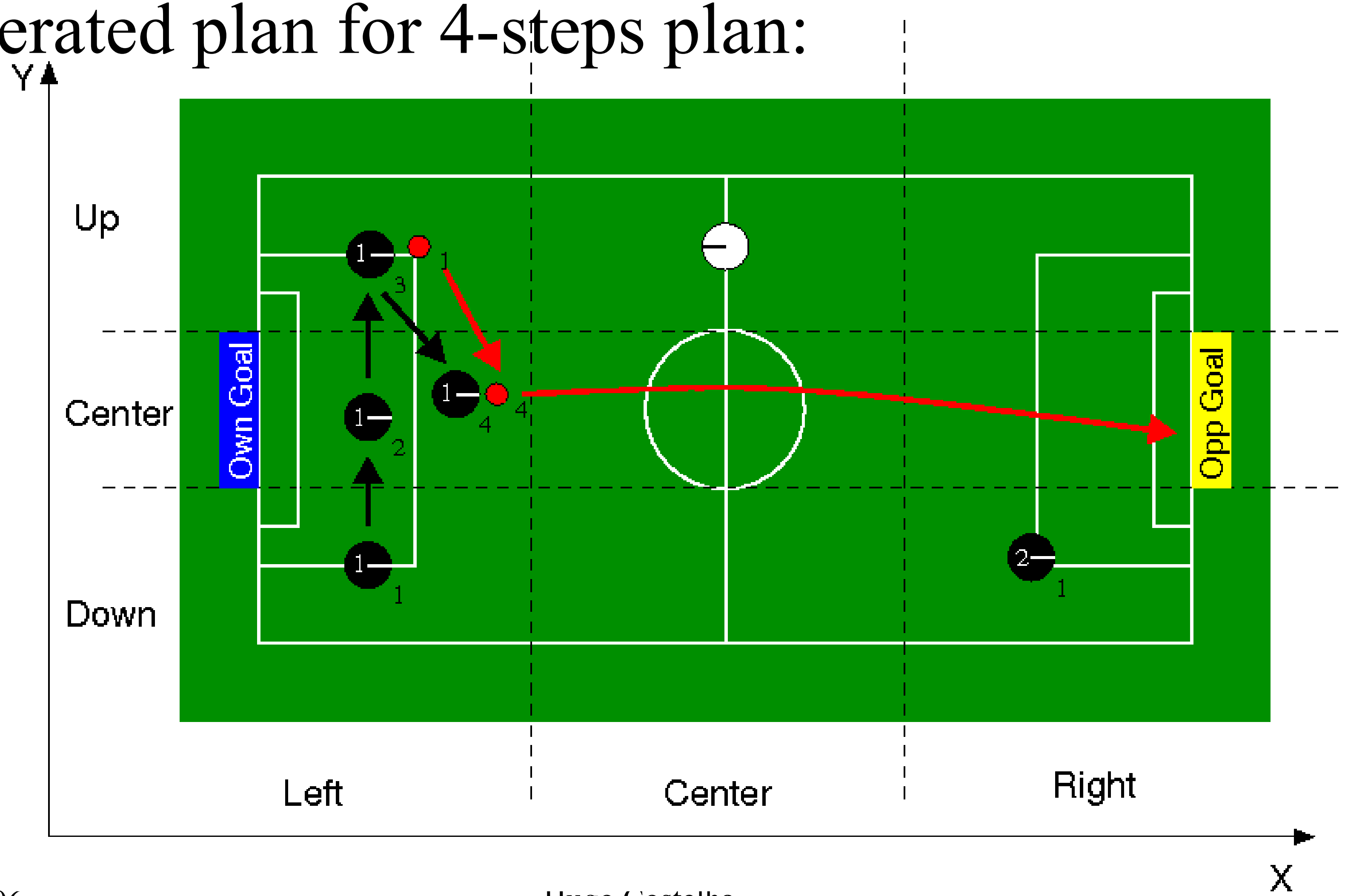
Probabilistic Environment

Generated plan for 3-steps plan:



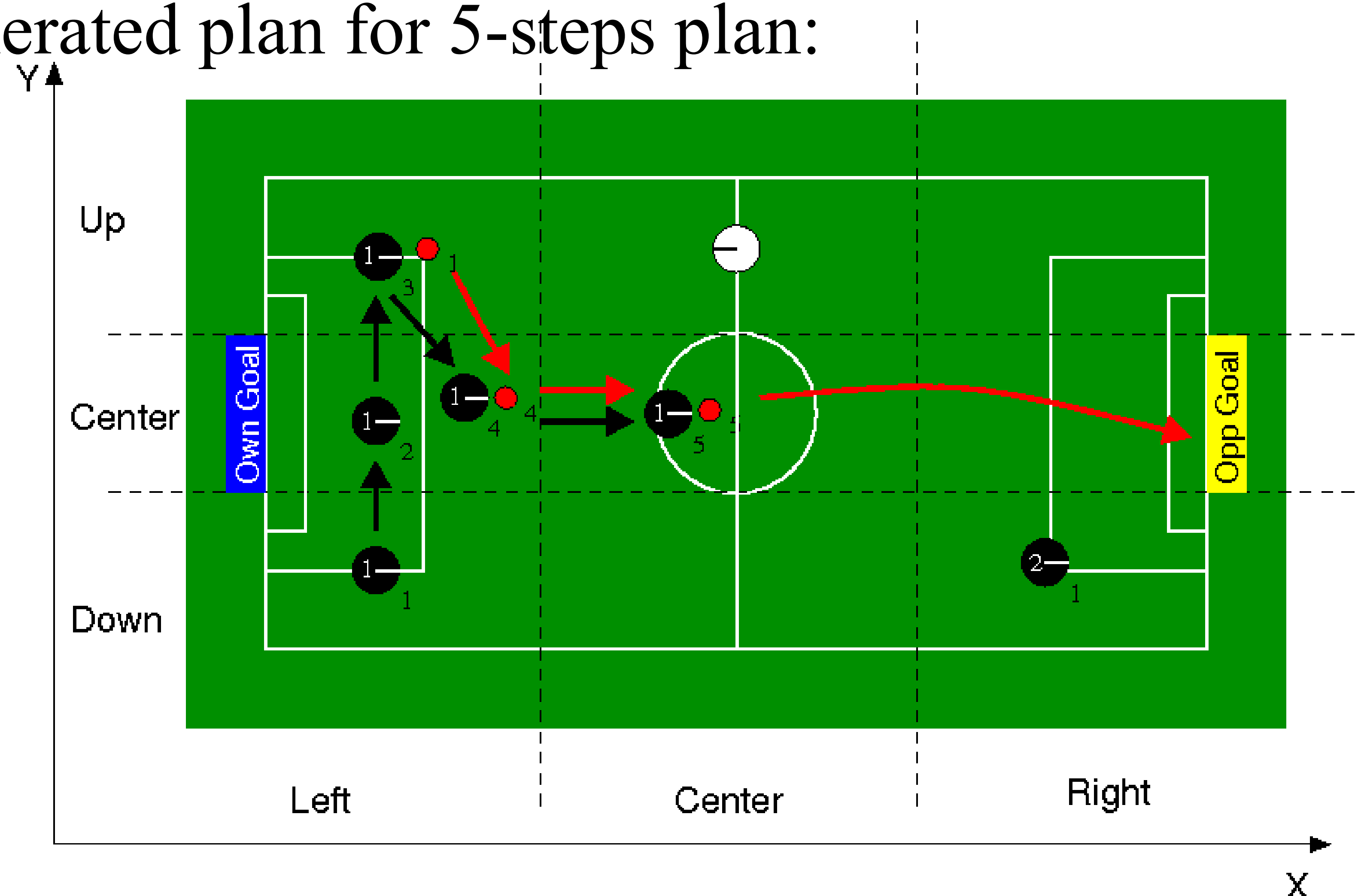
Probabilistic Environment

Generated plan for 4-steps plan:



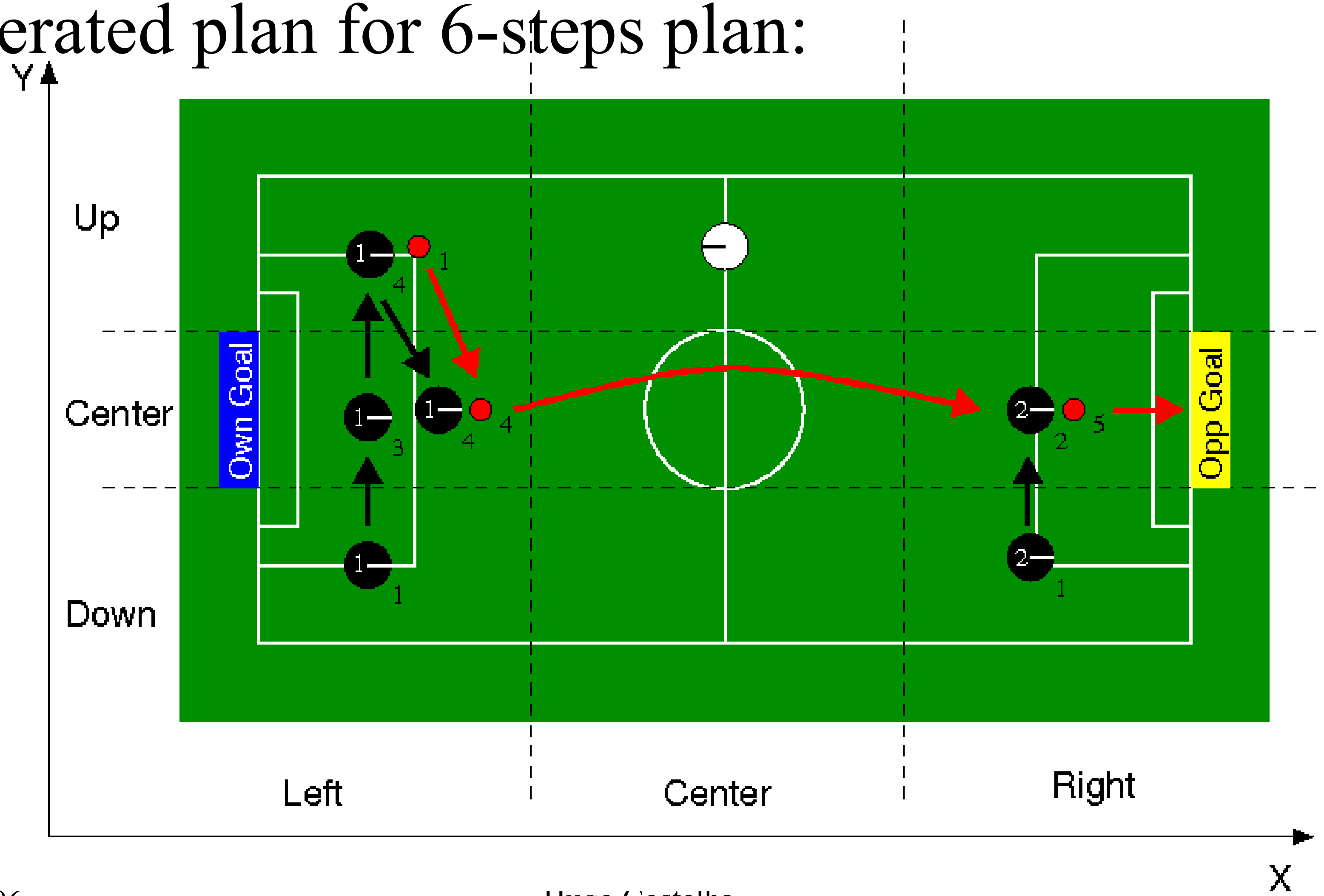
Probabilistic Environment

Generated plan for 5-steps plan:



Probabilistic Environment

Generated plan for 6-steps plan:



Probabilistic Environment

Generated plans:

Steps	Plans				
	3 Steps	**4 Steps**	**5 Steps**	**6 Steps**	**7 Steps**
1	P1_mN	P1_mN	P1_mN	P2_mN	P2_mN
2	P1_mN	P1_mN	P1_mN	P1_mN	P1_mN
3	P1_s	P1_tBS	P1_tBS	P1_mN	P1_mN
4	-	P1_s	P1_tBE	P1_tBS	P1_tBS
5	-	-	P1_s	P1_p	P1_p
6	-	-	-	P2_s	P2_s
7a	-	-	-	-	P2_s
7b	-	-	-	-	Ps_mW
Probability	0.02	0.04	0.1	0.16	0.17

INSTITUTO SUPERIOR TÉCNICO

PÓLO DO I.S.T

Time Statistics

- 10000 runs -

Deterministic:

Max. steps	Mean time [s]	Max. time [s]	% of runs
8	10.39	2319.02	100
7	5.17	242.35	99.77
6	2.08	39.49	98.39
5	0.64	8.42	92.13

Probabilistic:

Number of steps	Min. time [s]	Mean time [s]	Max. time	Mean Probability
5	0.55	1.82	11.02	0.16

Action Selection Statistics

Results with 10000 runs:

			Actions [%]									
	Players	StandBy	move				takeBall				pass	score
			N	S	E	W	N	S	E	W		
P	1	0	10.36	9.94	9.26	7.77	11.69	11.07	24.04	0	10.91	42.03
	2	0	23.41	22.58	22.35	19.34	15.64	14.95	42.53	0	15.86	57.97
D	1	0	53.02	55.17	19.19	70.99	17.47	17.02	41.02	0	19.29	251.36
	2	0	68.83	70.07	50.32	283.72	20.77	20.3	62.34	0.01	31.23	273.39

Summary

- Motivation
- Planning as SSat
- Application to Robotic Soccer
- Results and Discussion
- **Conclusions & Future Work**

INSTITUTO SUPERIOR TÉCNICO

PÓLO DO I.S.T

Conclusions & Future Work

- Although in early stages, SSat planning is already fast enough to be applied in a real-time situation;
- The generated plans are sequential, but could be applied concurrently;
- Need to address communications and observability issues;
- Use a topological description of the environment;
- Combine Discrete Event Systems for lower level modelling and analysis with these type of higher level planning;

Thanks for your attention!

Q&A

Encoding Probabilistic Planning Problems as Ssat Problems
-Quantifier Ordering-

$$\overbrace{\exists x_{1,1},\ldots,\exists x_{1,c_1}}^{\text{first action}}\ \overbrace{\rotatebox[origin=c]{180}{R} w_{1,1},\ldots,\rotatebox[origin=c]{180}{R} w_{1,c_2}}^{\text{first observation}}\cdots\overbrace{\rotatebox[origin=c]{180}{R} w_{n-1,1},\ldots,\rotatebox[origin=c]{180}{R} w_{n-1,c_2}}^{\text{last observation}}\ \overbrace{\exists x_{n,1},\ldots,.\exists x_{n,c_1}}^{\text{last action}}$$

$$\overbrace{\rotatebox[origin=c]{180}{R}^{\rho_1} z_1,\ldots,\rotatebox[origin=c]{180}{R}^{\rho_{c_4}} z_{c_4}}^{\text{domain uncertainty}}\ \overbrace{\exists y_1,\ldots,.\exists y_{c_3}}^{\text{states encountered}}$$

(Goal: $\max E[$ (initial/goal conditions (y,z)-clauses)

(action exclusion (x)-clauses)

(action outcome (w,x,y,z)-clauses)$]$)

c_1 = number of variables it takes to specify a single action step (the number of actions),
c_2 = number of variables it takes to specify a single observation,
c_3 = number of state variables (one for each proposition at each time step), and
c_4 = number of chance variables (one for each possible stochastic outcome at each time step).

INSTITUTO SUPERIOR TÉCNICO

PÓLO DO I.S.T

Problem Size

		Time Steps				
		1	**2**	**3**	**4**	**5**
Deterministic	**#Vars**	80	160	240	320	400
	#Clauses	1030	2030	3030	4030	5030
Probabilistic	**#Vars**	106	212	318	424	530
	#Clauses	1182	2334	3486	4638	5790