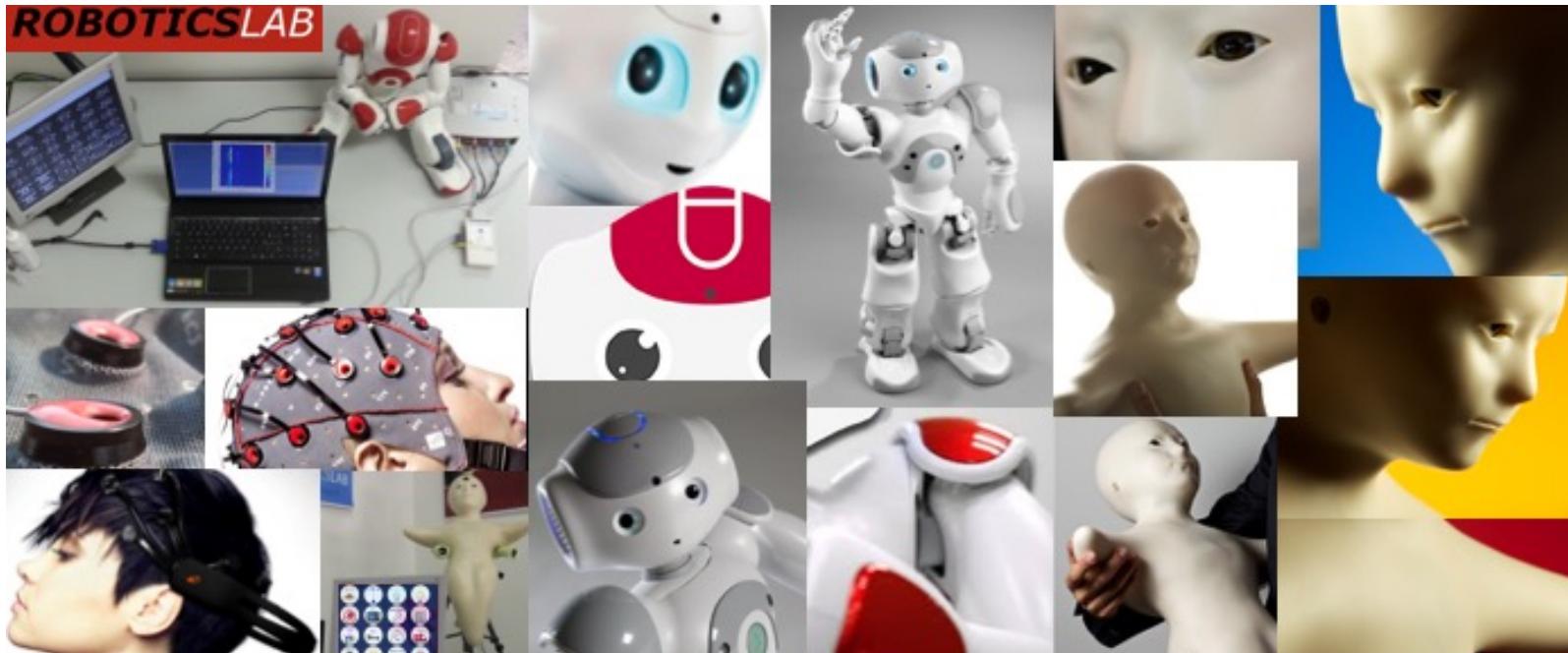




OSAKA UNIVERSITY



ATR

g-tec

## ANALISI ECONOMICA E SOCIALE SUI ROBOT INTERNATIONAL FEDERATION ROBOTICS

PROF. ROSARIO SORBELLO

2° LEZIONE - PALERMO 12 OTTOBRE 2022



## TASSONOMIA DEI ROBOT

- A **robot** is an actuated mechanism programmable in two or more axes with a degree of autonomy, moving within its environment, to perform intended tasks. Autonomy in this context means the ability to perform intended tasks based on current state and sensing, without human intervention.
- A **service robot** is a →robot that performs useful tasks for humans or equipment excluding industrial automation application. Note: The classification of a robot into industrial robot or service robot is done according to its intended application.
- A **personal service robot** or a service robot for personal use is a →service robot used for a non-commercial task, usually by lay persons. Examples are domestic servant robot, automated wheelchair, and personal mobility assist robot.
- A **professional service robot** or a service robot for professional use is a →service robot used for a commercial task, usually operated by a properly trained →operator. Examples are cleaning robot for public places, delivery robot in offices or hospitals, fire-fighting robot, rehabilitation robot and surgery robot in hospitals. In this context, an operator is a person designated to start, monitor and stop the intended operation of a →robot or a →robot system.
- A **robot system** is a system comprising →robot(s), end-effector(s) and any machinery, equipment, devices, or sensors supporting the robot performing its task.



# PRINCIPALI TIPOLOGIE DI ROBOT

Section I	Types of robots: Service robots for personal/domestic use
1-6	Robots for domestic tasks 1 Robot companions/assistants/humanoids 2 Vacuuming, floor cleaning 3 Lawn-mowing 4 Pool cleaning 5 Window cleaning 6 Others
7-10	Entertainment robots 7 Toy/hobby robots 8 Multimedia/remote presence 9 Education and research 10 Others
11-13	Elderly and handicap assistance 11 Robotized wheelchairs 12 Personal aids and assistive devices 13 Other assistance functions
14 15 16	Personal transportation (AGV for persons) Home security & surveillance Other Personal / domestic robots

PROF. ROSARIO SORBELLO



# PRINCIPALI TIPOLOGIE DI ROBOT

<b>Section II</b>		<b>Types of robots: Service robots for professional use</b>
17-23		Field robotics
17		Agriculture
18		Milking robots
19		Other robots for livestock farming
20		Forestry and silviculture
21		Mining robots
22		Space robots
23		Other field robotics
24-28		Professional cleaning
24		Floor cleaning
25		Window and wall cleaning (incl. wall climbing robots)
26		Tank, tube and pipe cleaning
27		Hull cleaning (aircraft vehicles etc.)
28		Other cleaning tasks
29-31		Inspection and maintenance systems
29		Facilities, plants
30		Tank, tubes, pipes and sewers
31		Other inspection and maintenance systems
32-35		Construction and demolition
32		Nuclear demolition & dismantling
33		Building construction
34		Robots for heavy/civil construction
35		Other construction and demolition systems
36-39		Logistic systems

PROF. ROSARIO SORBELLO



# PRINCIPALI TIPOLOGIE DI ROBOT

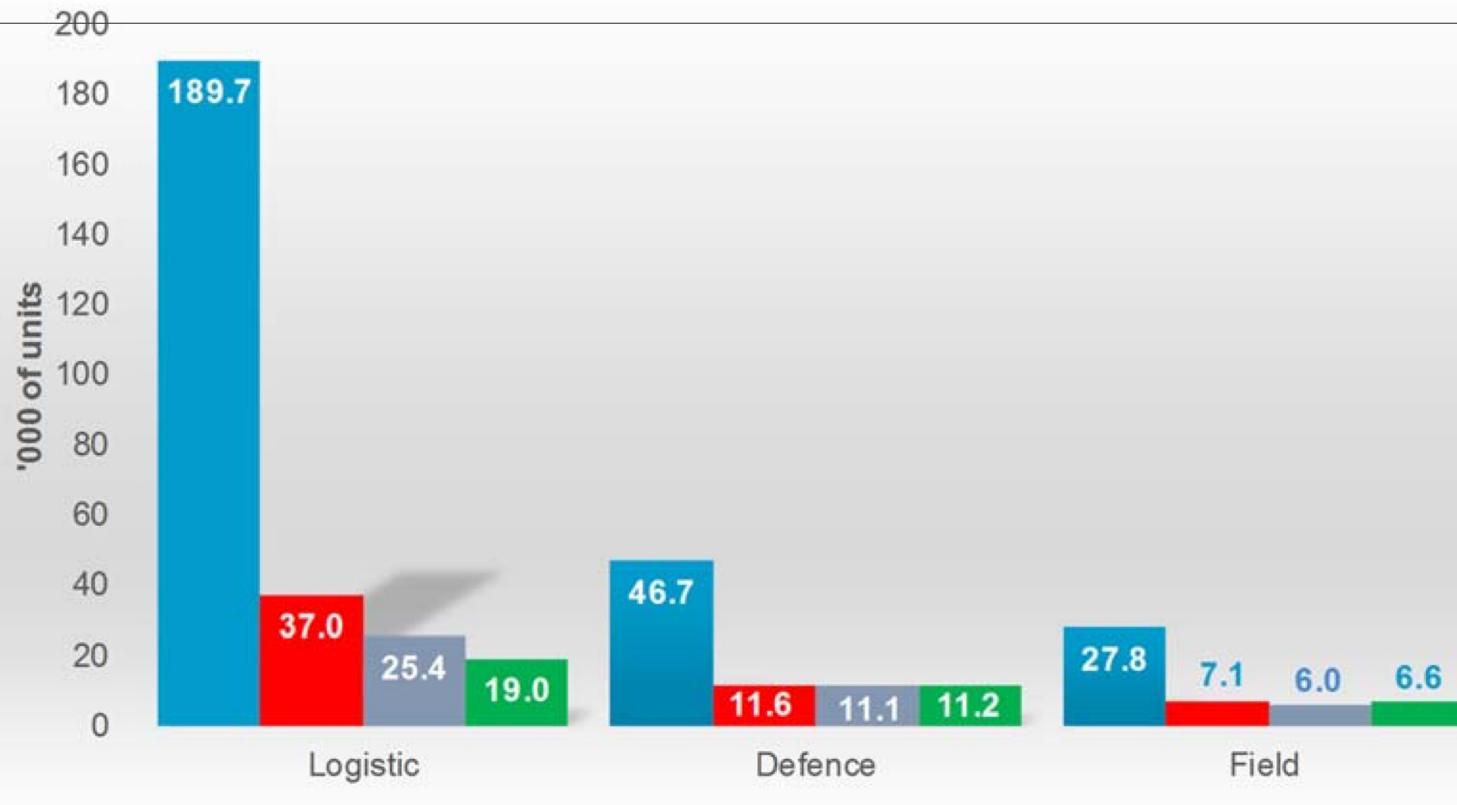
36-39	Logistic systems
36	Automated guided (AGV) vehicles manufacturing environments
37	AGVs non-manufacturing environments (indoor)
38	Cargo handling, outdoor logistics
39	Other logistic systems
40-43	Medical robotics
40	Diagnostic systems
41	Robot assisted surgery or therapy
42	Rehabilitation systems
43	Other medical robots
44-46	Rescue & security applications
44	Fire and disaster fighting robots
45	Surveillance / security robots
46	Other rescue and security robots
47-50	Defense applications
47	Demining robots
48	Unmanned aerial vehicles
49	Unmanned ground based vehicles
50	Unmanned underwater vehicles
51	Other defense applications
52	Underwater systems (civil / general use)
53	Powered Human Exoskeletons
54	Unmanned aerial vehicles (general use)
55	Mobile Platforms in general use
56-60	Underwater systems (civil / general use)
56	Hotel & restaurant robots
57	Mobile guidance, information robots
58	Robots in marketing
59	Robot joy rides
60	Others (i.e. library robots)
61	Other professional service robots not specified above

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## SERVICE ROBOT FOR PROFESSIONAL USE



Service robots for professional use in main applications. Unit sales 2015 and 2016, forecast 2017 and 2018-2020.

PROF. ROSARIO SORBELLO



ROBOTICS LAB

## SERVICE ROBOT FOR PROFESSIONAL USE

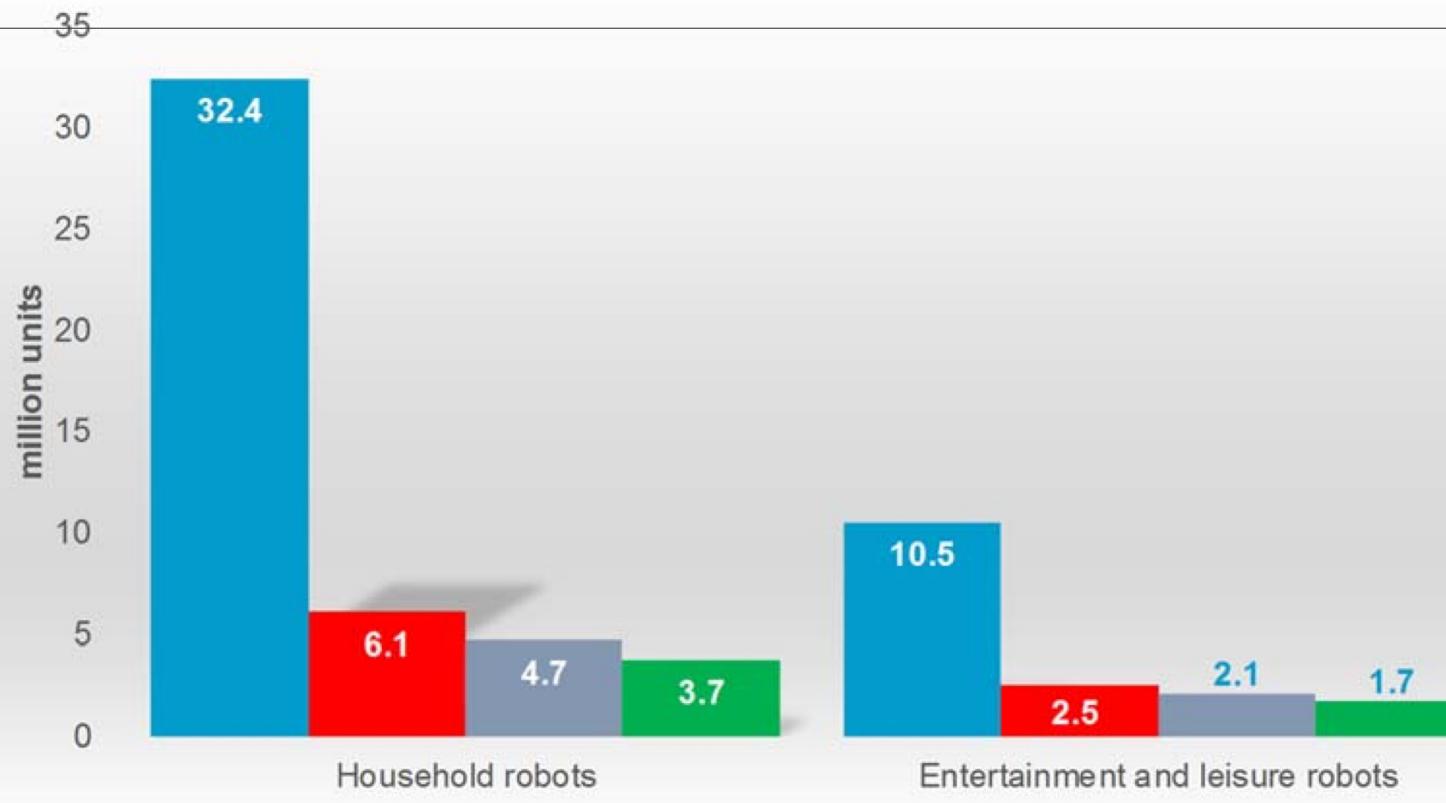


Service robots for professional use in more applications, unit sales 2015 and 2016, forecast 2017\* and 2018\*- 2020\* PROF. ROSARIO SORBELLO



ROBOTICS LAB

## SERVICE ROBOT FOR PROFESSIONAL USE



\*forecast

■ 2018\*-2020\* ■ 2017\* ■ 2016 ■ 2015

Source: World Robotics 2017

Service robots for personal/domestic use. Unit sales 2015 and 2016, forecast 2017\*  
and 2018\*- 2020\*

PROF. ROSARIO SORBELLO



## VENDITE DEI ROBOT PER CATEGORIE

### **Professional service robots:**

- 2016: 59,700 units, +24%
- 2017: 78,700 units, +17%
- 2018-2020: 397,000 units, 20% - 25% per year on average

### **Service robots for domestic/household tasks:**

- 2016: 4.7 million units, +25%
- 2017: 6.1 million units, +30%
- 2018-2020: 32.4 million units, between 30%-35% per year on average

### **Service robots for entertainment:**

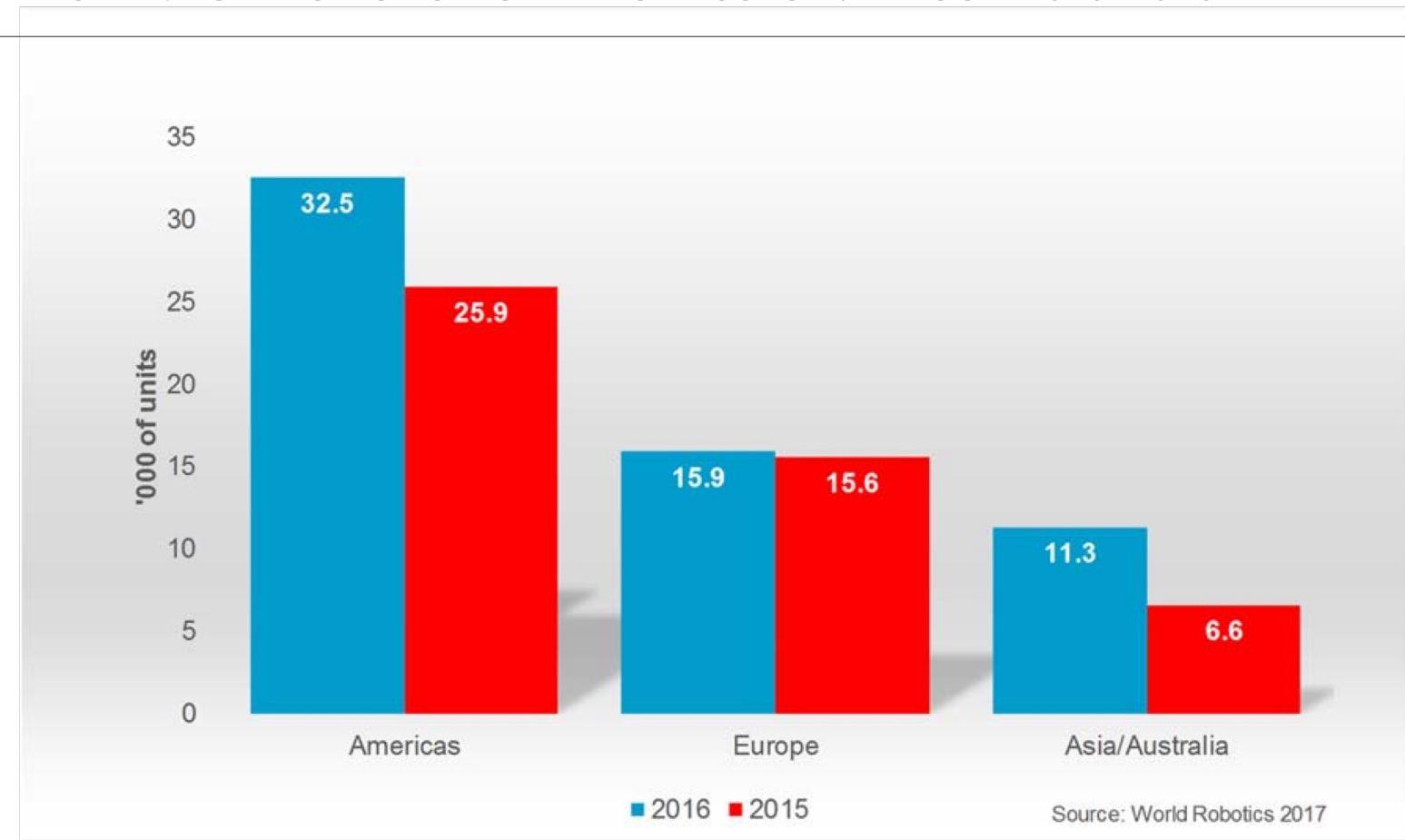
- 2016: 2.1 million units, +22%
- 2017: 2.6 million units, +22%
- 2018-2020: 10.5 million units, between 20%-25% per year on average

PROF. ROSARIO SORBELLO



ROBOTICS LAB

## SERVICE ROBOTS FOR PROFESSIONAL USE 2015-2016

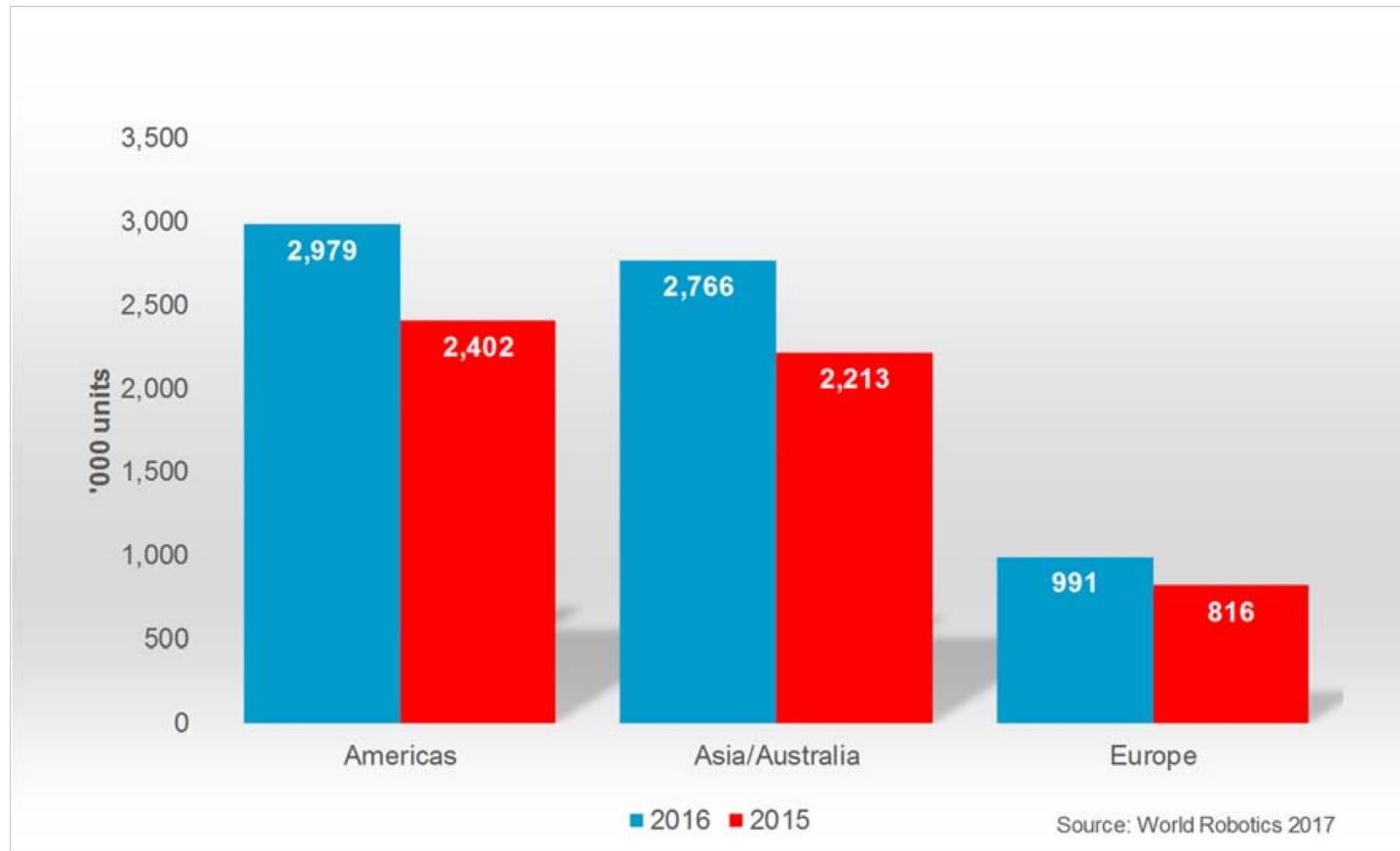


Service robots for professional use. Unit sales 2015 and 2016 by region of origin.

PROF. ROSARIO SORBELLO



## SERVICE ROBOTS FOR DOMESTIC USE 2015-2016



Service robots for personal/domestic use. Unit sales 2015 and 2016 by region of origin.

PROF. ROSARIO SORBELLO



## Robotics deals with robots

- Robots are artificial beings; two types of artificial beings exist:
  - Those that include biological material or parts capable of auto-organization
  - Those that do not contain any biological part
- The robots discussed here belong to the second type; they do not contain biological parts, but are able to
  - move autonomously in a partially known environment
  - interact with humans (supplying collaborative behaviour)
  - cooperate with other robots to execute specific tasks
  - learn from a teacher or from experience

PROF. ROSARIO SORBELLO

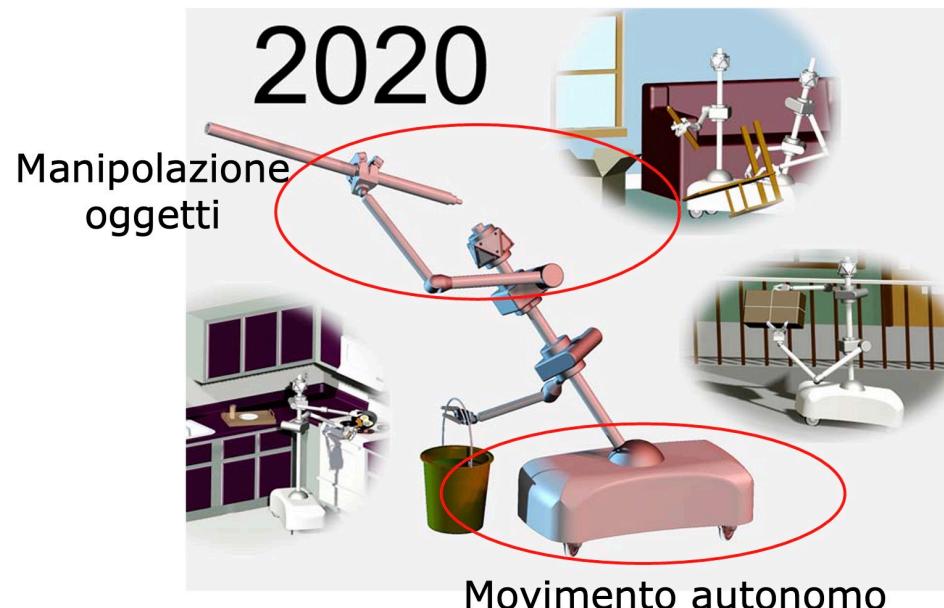


- 
- Robotics is a multidisciplinary approach; it includes:
    - mechanics (structures, ...)
    - electronics (sensors, actuators, control boards, ...)
    - information technologies (AI, algorithms, ...)
    - advanced control theory
    - electrical engineering (motors, power sources, ...)
    - telecommunications
  - But also
    - cognitive sciences, neurosciences
    - biomimesis, ethology
    - material sciences and nanotechnology
    - ergonomics, safety,
    - interfaces



---

## **Forse ci aspetta uno scenario diverso**



**Attualmente, manipolazione e movimento autonomo  
sono ancora scarsamente integrati**

PROF. ROSARIO SORBELLO



---

## Cosa ci riserva il futuro ?

Meglio essere cauti, considerando alcuni illustri precedenti

- “Posso affermare tranquillamente che macchine volanti più pesanti dell’aria sono impossibili”

William Thomson, Lord Kelvin, 1895

- “Gli aspirapolvere a energia nucleare saranno probabilmente una realtà entro dieci anni”

Alex Lewyt, inventore e filantropo, 1955

- “Non c’è nessun motivo per cui qualcuno possa volere un computer a casa sua”

Ken Olson, presidente e fondatore, Digital Equipment Corp., 1977



**ROBOTICS LAB**

---

## Cosa prevede l'Europa

- La piattaforma europea della robotica “EUROP” prevede tre settori chiave in cui concentrare gli sforzi di ricerca



Robotica Industriale



Robotica di Servizio



Robotica per  
Sicurezza e Spazio



**ROBOTICS LAB**

---

## Cosa prevede l'Europa

### Robotica Industriale

- *Gradualmente i robot diventeranno gli assistenti degli operai, fungendo da loro servitori sul posto di lavoro.*
- *Eseguiranno i compiti più ripetitivi e faticosi, e avranno il compito di prevenire gli infortuni, alleviando in questo modo le preoccupazioni di una crescente spesa sanitaria, particolarmente per le PMI.*



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## Cosa prevede l'Europa

### Robotica di Servizio

*La robotica di servizio si espanderà in ogni settore della nostra vita futura ...  
Ciò offrirà opportunità significative alle aziende europee.*

*La Robotica di servizio può essere convenientemente suddivisa nei tre segmenti di mercato seguenti:*

- *Divertimento e svago*



- *Usi domestici*



- *Applicazioni professionali*



PROF. ROSARIO SORBELLO



## Cosa prevede l'Europa

### Sicurezza

- *In the coming years, security enforcement ... will rely more and more on automated systems. In the first instance, civil security missions will include **surveillance of critical infrastructure** (telecommunication and power lines, water, gas and oil pipelines, etc.) and **extended border and coastal patrol** boosted by the shift of EU borders to the East and the need **to monitor illegal immigration fluxes.***
- *In the longer term, other major application drivers will include **maritime surveillance** (traffic control and monitoring of ship movements, supervision of illegal fishery), **crime monitoring**, **search and rescue operations**, **environmental monitoring** (fire detection and fire fighting, oil spill discovery, etc.) and **surveillance of hazardous materials.***



PROF. ROSARIO SORBELLO



## Cosa prevede l'Europa

### Spazio

**Robotica nello spazio:** i robot dovranno operare con una limitata assistenza umana, in ambienti incerti e ostili.



**Esplorazione spaziale e planetaria**

**Montaggio e manutenzione nello spazio**

- Solleveranno gli astronauti da compiti di ispezione e manutenzione in orbita. In prospettiva, assisteranno gli esseri umani nello sviluppo di strutture orbitali e di superficie con un gran numero di componenti.



## Quali saranno le applicazioni future

- **USA:** la ricerca è prevalentemente finanziata dal *Defense Advanced Research Projects Agency* (DARPA)
  - Sistemi per la sicurezza interna
  - UAV: velivoli tele- e auto-guidati per sorveglianza e attacco
  - Veicoli autonomi per ambienti anche complessi (urbani)
  - Robot per intervento sui campi di battaglia



Battlefield Extraction-Assist Robot



## Quali sono le applicazioni future

- **Giappone:** robot ludici per svago e compagnia, ma anche assistenza a malati e anziani



PROF. ROSARIO SORBELLO



## Quali sono le applicazioni future

- **Europa:** robotica di servizio
  - Assistenza domestica agli anziani e ai portatori di handicap
  - Supporto all'assistenza ospedaliera
  - Supporto alle squadre di soccorso
  - Sorveglianza, pulizia e servizi logistici di edifici pubblici e aree industriali
  - Servizi domotici



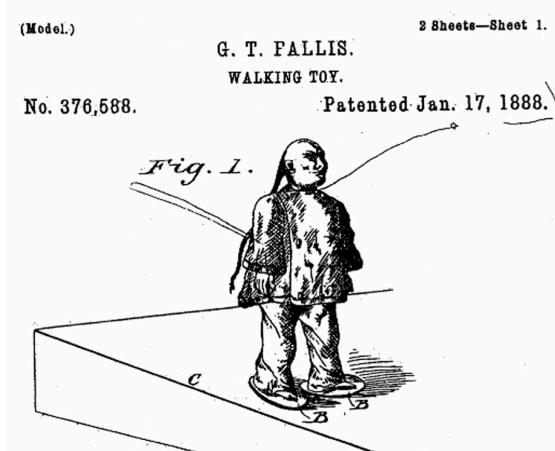
PROF. ROSARIO SORBELLO



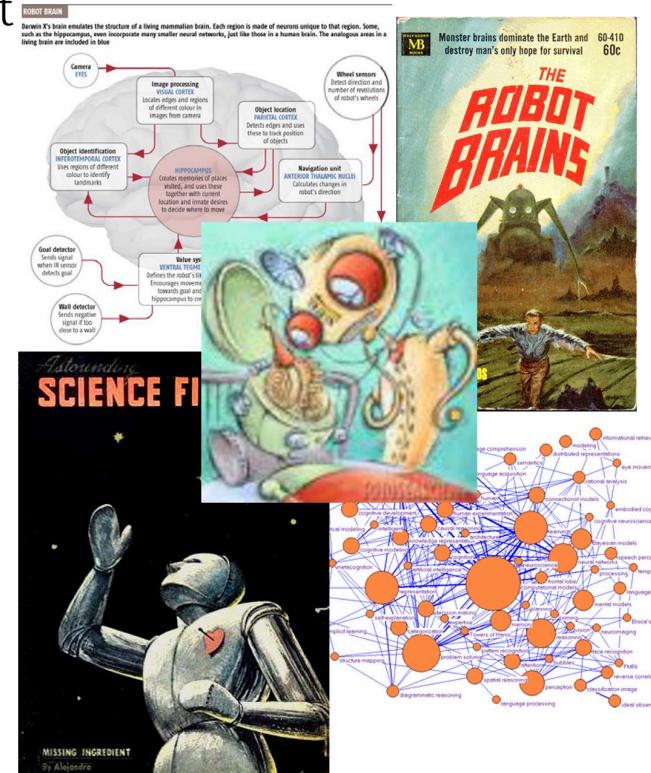
ROBOTICS LAB

## Per realizzare i robot futuri

Non basta dotare un robot di due braccia e due gambe



Occorre includere anche un cervello

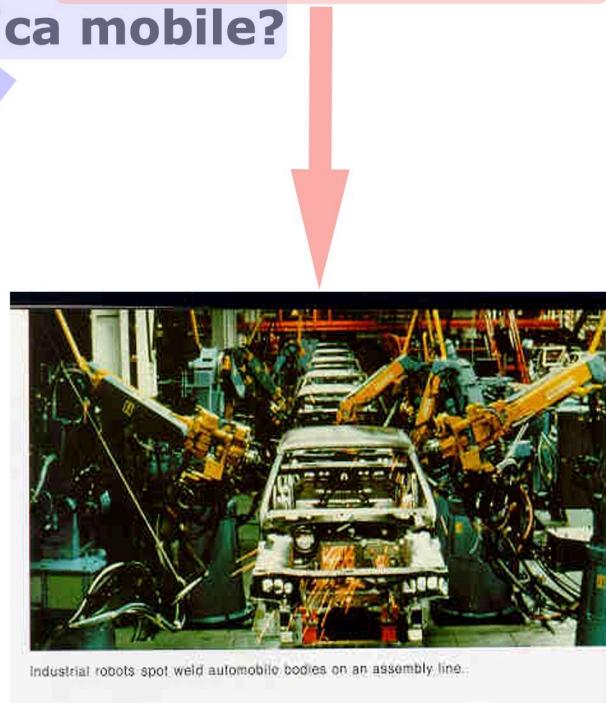
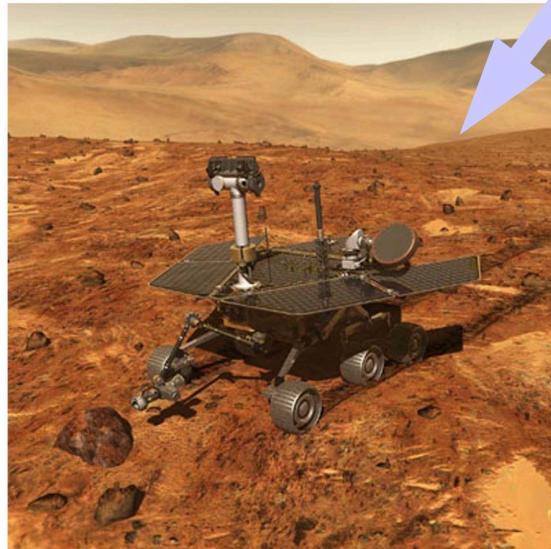


PROF. ROSARIO SORBELLO



---

**Che differenza c'è tra la robotica industriale  
e la robotica mobile?**



Industrial robots spot weld automobile bodies on an assembly line.

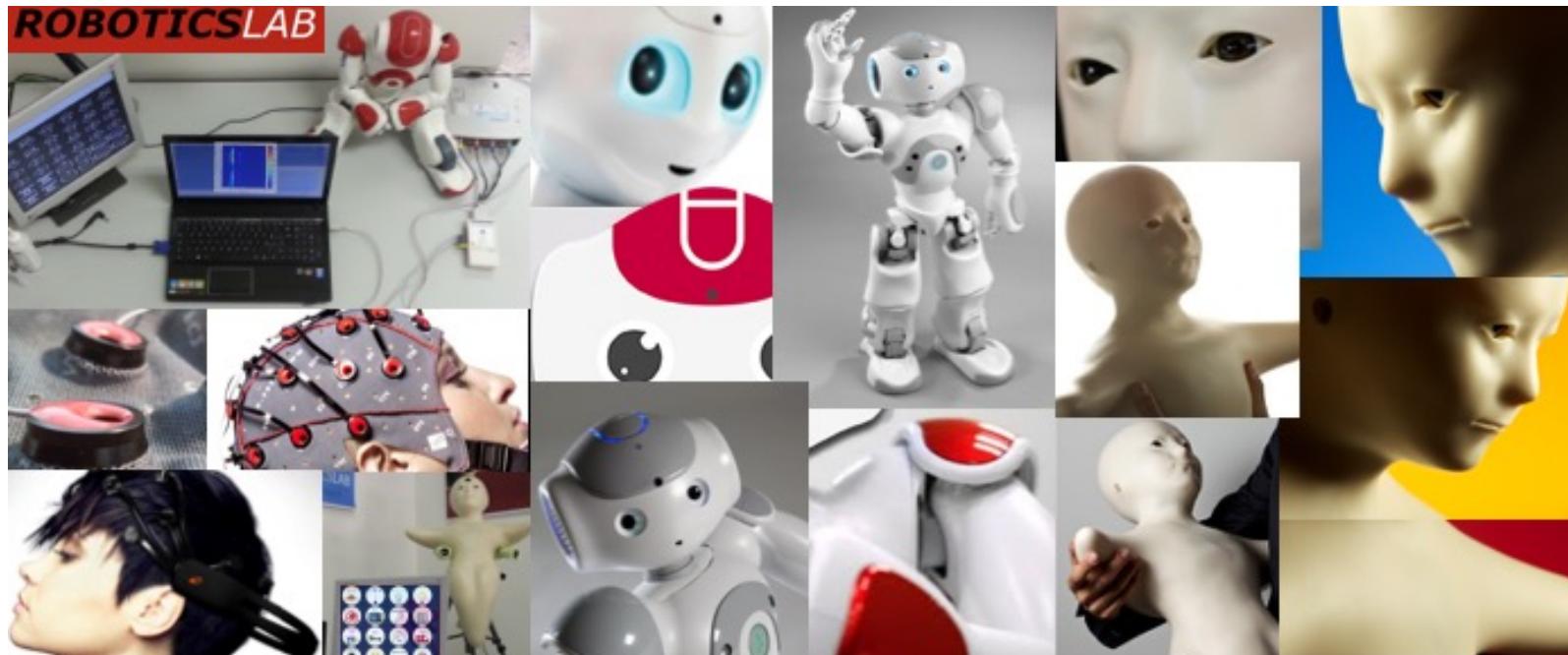
PROF. ROSARIO SORBELLO



- Robotica **industriale**: si concentra sulla **manipolazione**
  - settori tradizionali manifatturieri: per movimentazione, montaggi, saldatura, packaging, verniciatura, deposizione colle e sigillanti
  - diffusione crescente in settori non tradizionali (agricoltura, abbigliamento, alimentare ecc.)
- Robotica **mobile e di servizio**: si concentra sulla capacità di **movimento autonomo** e sull'**intelligenza**
  - sorveglianza, logistica e pulizia edifici
  - robotica domestica e assistiva
  - servizi ospedalieri, per disabili, per anziani
  - disastri naturali, bonifica aree “inquinate”, sminamento
  - esplorazione (spaziale, sottomarina, ecc.)
  - ... altro ...



OSAKA UNIVERSITY



ATR

g-tec

## ROBOTICA E LE SUE ORIGINI

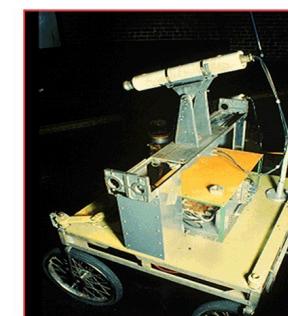
PROF. ROSARIO SORBELLO

2° LEZIONE - PALERMO 12 OTTOBRE 2022



## TIMELINE ROBOTICA

1. 1970: SRI Shakey
2. 1975: PUMA manipulator
3. 1979: Stanford cart
4. 1999: Sony AIBO
5. 2000: Honda Asimo
6. 2004: Mars rovers Spirit & Opportunity
7. 2006-7: DARPA Challenge



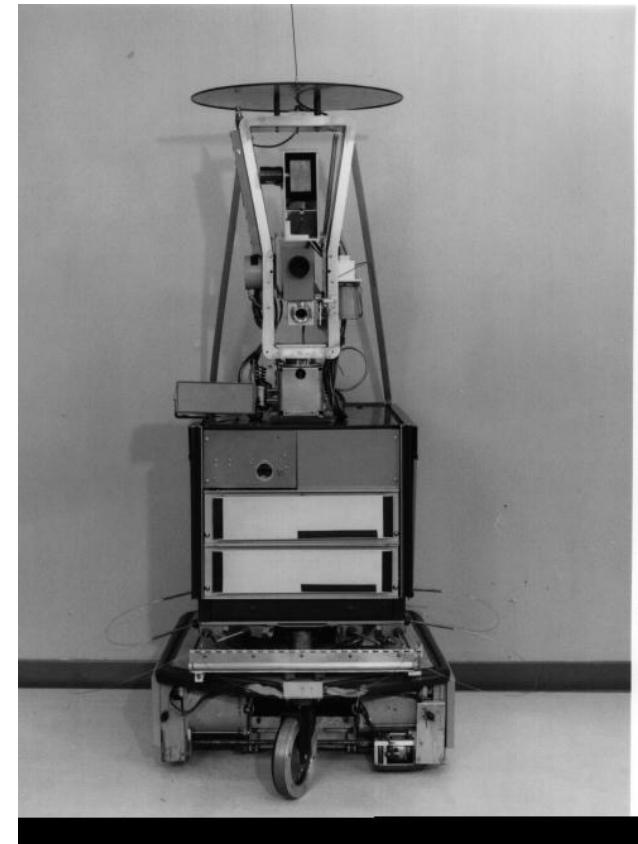
PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## PRIMI ROBOT: SHAKEY

- Tardi anni 1960
- Allo Stanford Research Institute
- Navigazione visiva in ambiente speciale
- Pianificatore STRIPS
- Sensori di visione e contatto



[VIDEO SHAKEY](#)

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

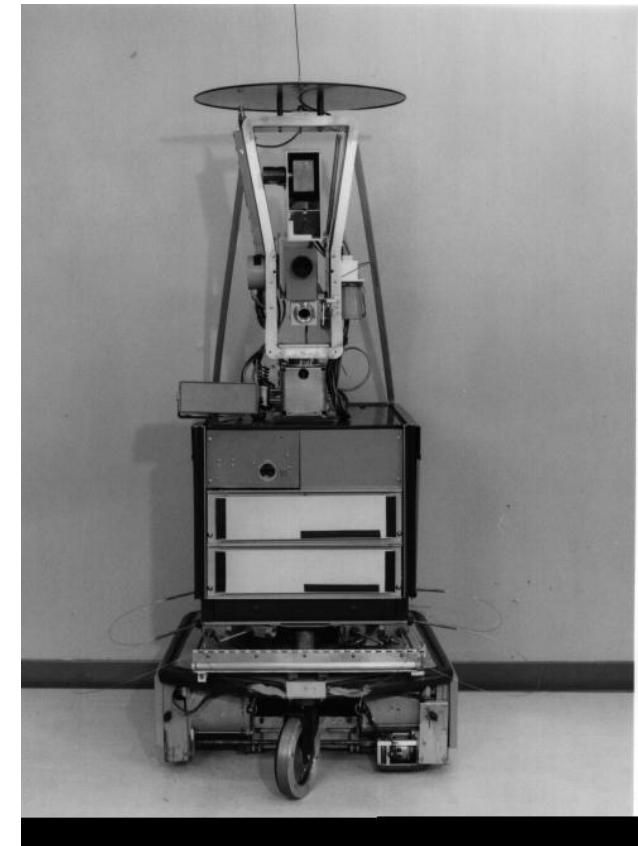
## PRIMI ROBOT: SHAKEY

- Pianificatore STRIPS:

**STANFORD RESEARCH INSTITUTE PROBLEM SOLVER**

STRIPS (STANFORD RESEARCH INSTITUTE PROBLEM SOLVER) È UN PIANIFICATORE AUTOMATICO SVILUPPATO NEL 1971.

IL SISTEMA SFRUTTA LA **LOGICA DELLE PROPOSIZIONI** ESPANSA CON I PREDICATI ED IL MECCANISMO DI RICERCA NELLO SPAZIO DEGLI STATI, PER OTTENERE UNA POSSIBILE SEQUENZA DI AZIONI CHE, SE ESEGUITE, PROVOCANO IL RAGGIUNGIMENTO DI UNO STATO FINALE DEL MONDO A PARTIRE DA UNO INIZIALE.



[VIDEO SHAKEY](#)

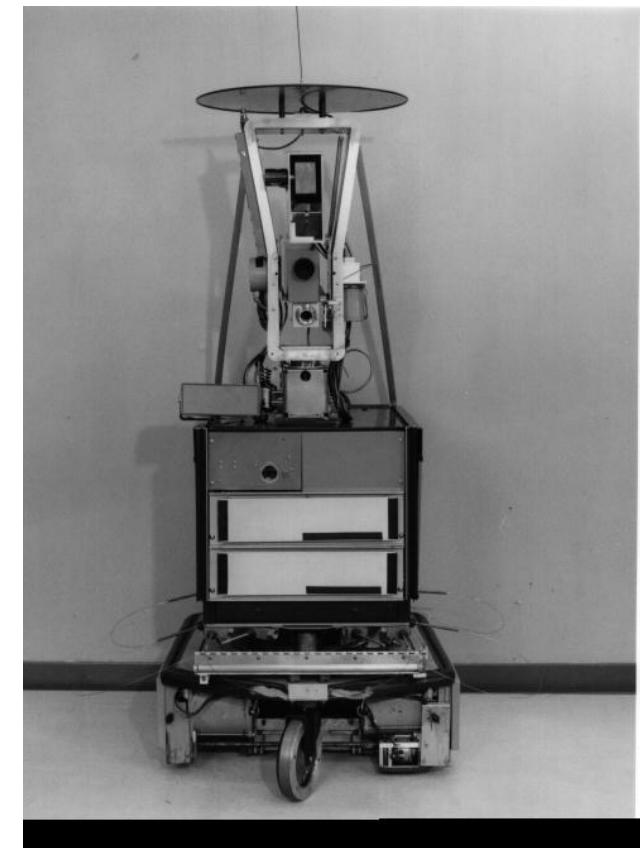
PROF. ROSARIO SORBELLO



- Pianificatore STRIPS

STRIPS È EFFICACE IN TUTTI QUEI CASI IN CUI IL **PROBLEMA E L'AMBIENTE SIANO NOTI A PRIORI** CON CERTEZZA MA HA ALCUNE LIMITAZIONI:

- IMPOSSIBILITÀ DI GESTIRE DELLE DIFFERENZE TRA MODELLO E AMBIENTE;
- LE MODIFICHE AL SISTEMA DEVONO SEMPRE E SOLO AVVENIRE PER EFFETTO DELL'AGENTE INTELLIGENTE, QUINDI È INADATTO ALLA COOPERAZIONE O ALL'INTERAZIONE IN GENERALE;
- AUMENTANDO LA COMPLESSITÀ DEL MODELLO, INTRODUCENDO NUOVE VARIABILI E OPERAZIONI, LA QUANTITÀ DI TEMPO DI CALCOLO NECESSARIO A PIANIFICARE UN'OPERAZIONE CRESCE ESPONENZIALMENTE



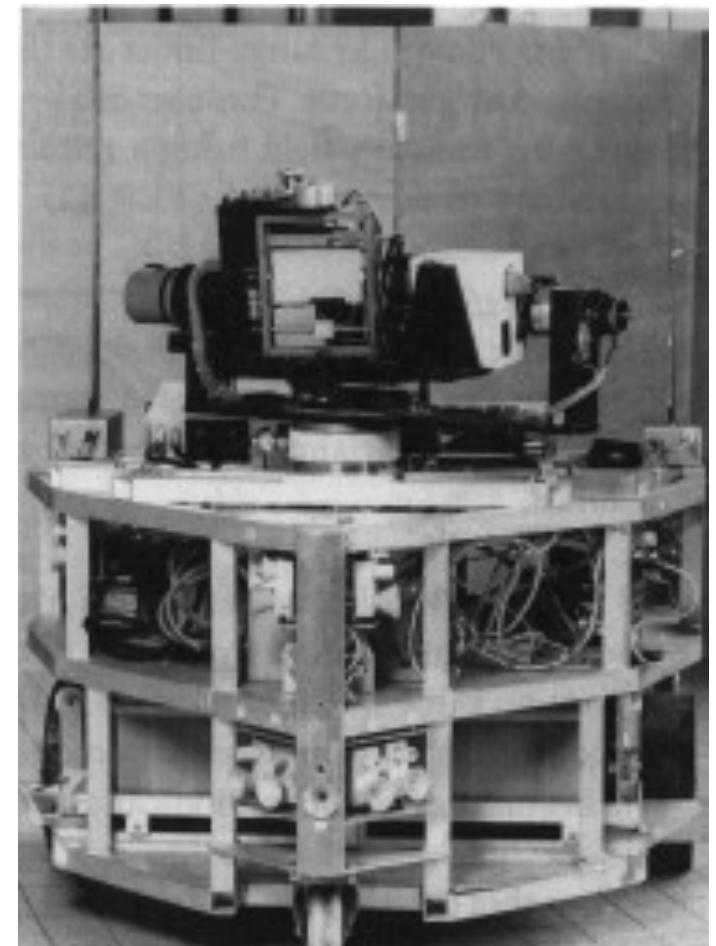
SHAKEY ROBOT

PROF. ROSARIO SORBELLO



- Tardi anni 1970 al LAAS a Tolosa
- Video Camera, 14 ultrasuoni,  
Rangefinder laser
- In uso per una decina di anni
- Rappresentazioni spaziali multi-livello

**H**EURISTIQUES  
**I**NTEGREES AUX  
**L**OGICIELS ET AUX  
**A**UTOMATISMES DANS UN  
**R**OBOT  
**E**VOLUTIF



PROF. ROSARIO SORBELLO

[HILARE ROBOT](#)



**ROBOTICS LAB**

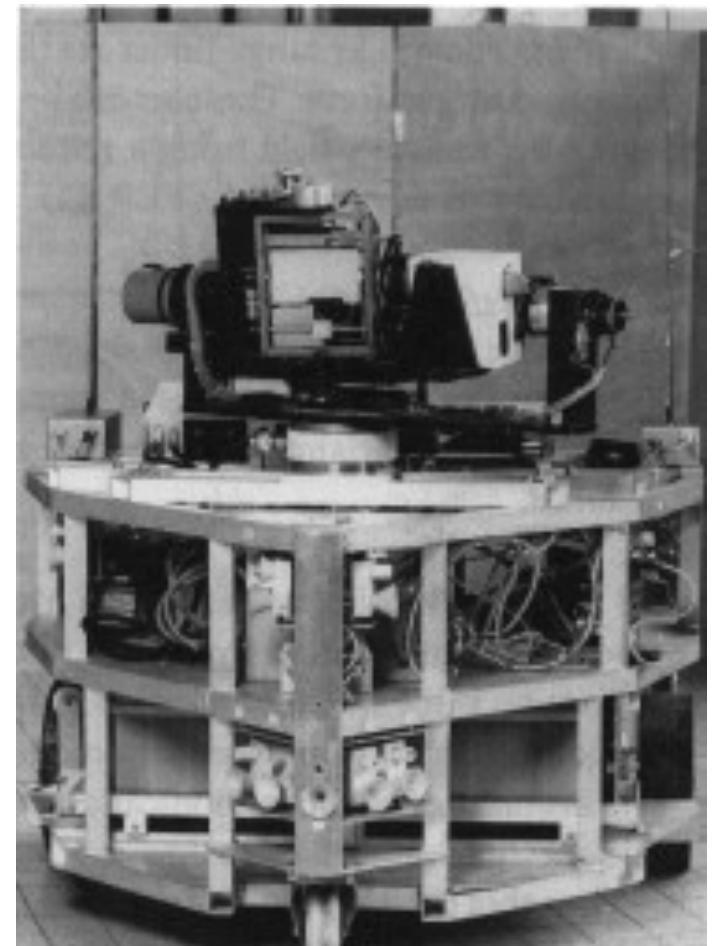
## PRIMI ROBOT: HILARE

- Tardi anni 1970 al LAAS a Tolosa
- Video, ultrasuoni, Rangefinder laser
- In uso per una decina di anni
- Rappresentazioni spaziali multi-livello

**1 LIVELLO: MODELLO GEOMETRICO  
DISTANZA DEL MONDO**

**2 LIVELLO: MODELLO RELAZIONALE  
PER CONNESSIONI TRA STANZA E  
CORRIDOI**

PROF. ROSARIO SORBELLO



[HILARE ROBOT](#)

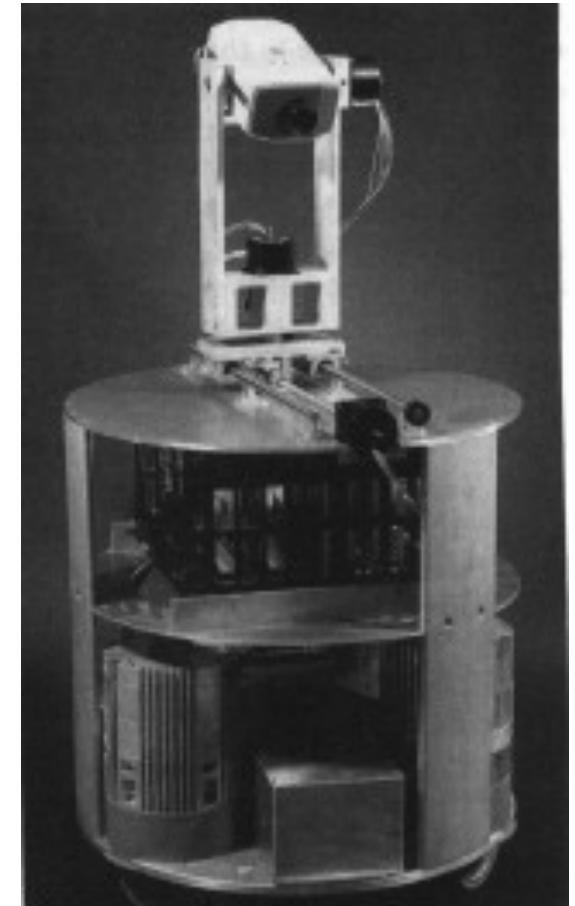


ROBOTICS LAB

## PRIMI ROBOT: CART E ROVER



- Primi robot di Moravec
  1. Stanford Cart (1977)
  2. CMU rover (1983)
- Sonar e Visione stereoscopica



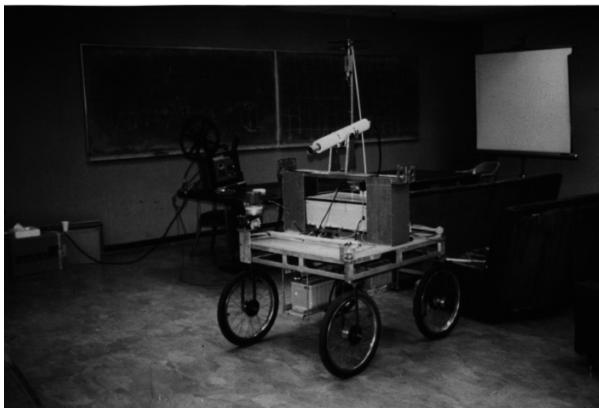
[CART-CMU ROVER ROBOTS](#)

PROF. ROSARIO SORBELLO



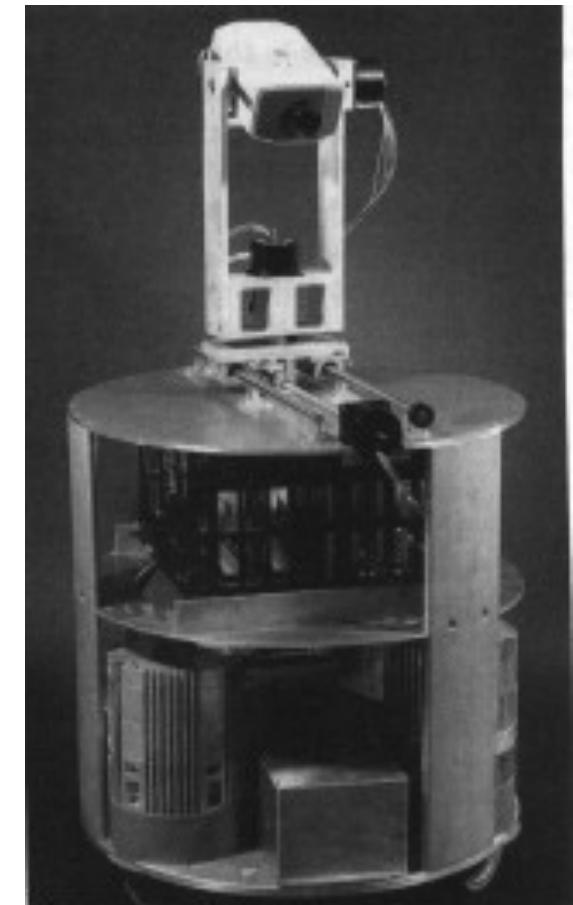
**ROBOTICS LAB**

## PRIMI ROBOT: CART E ROVER



MAPPA INTERNA DEGLI OSTACOLI RAPPRESENTATI COME SFERE

ALGORITMO GRAPH SEARCH PER IL PERCORSO PIU' BREVE



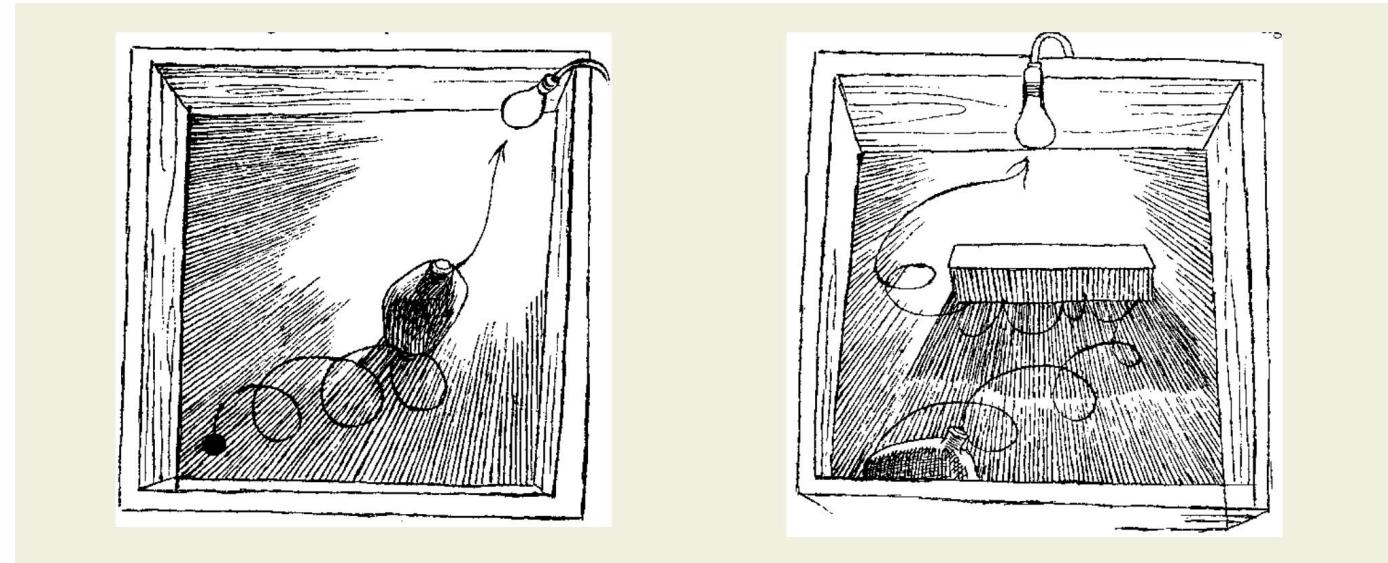
[CART-CMU ROVER ROBOTS](#)

PROF. ROSARIO SORBELLO



ROBOTICS LAB

## TARTARUGA DI GREY WALTER



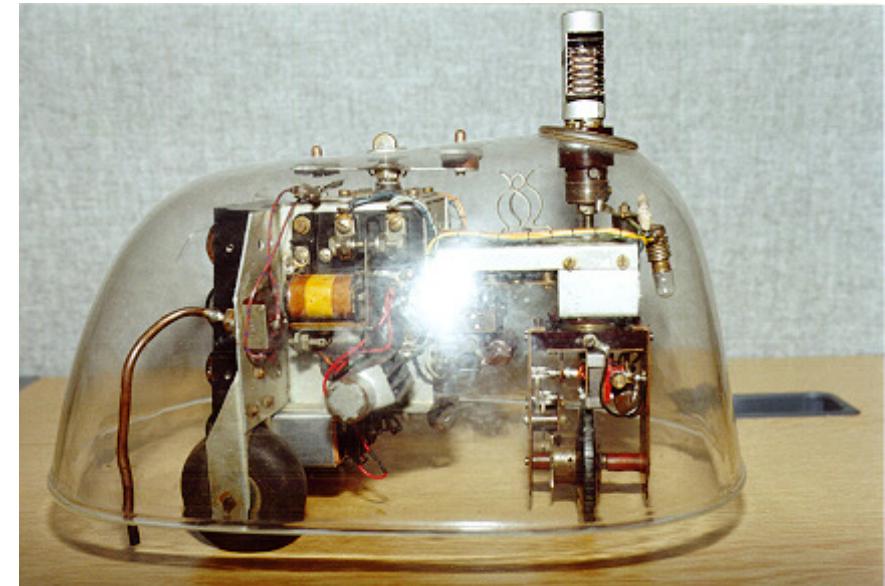
PRIMI ESEMPI DI MACCHINE AUTONOME MOBILI ("TARTARUGHE"), CON LO SCOPO DI ILLUSTRARE IL FUNZIONAMENTO DI ALCUNI MECCANISMI CEREBRALI.

PROF. ROSARIO SORBELLO



## TARTARUGA DI GREY WALTER

- Machina Speculatrix
- Comportamenti
- Cerca luce
- Orientati verso la luce debole
- Indietreggia dalla luce intensa
- Gira e spingi
- Ricarica la batteria
- Arbitraggio basato su priorità



GREY WALTER ROBOT 1

primi esempi di macchine autonome mobili ("*tartarughe*"),  
con lo scopo di illustrare il funzionamento di alcuni  
meccanismi cerebrali.

GREY WALTER ROBOT 2



**ROBOTICS LAB**

## VEICOLI DI BRAITENBERG

---

- Esperimenti immaginari
- Veicolo 1
  - Muovi avanti
- Veicolo 2
  - “paura” - Fuggi dalla luce
  - “Aggressione” - Attacca la luce
- Molti veicoli, anche con comportamenti complessi

PROF. ROSARIO SORBELLO



*The speed of the motor is controlled by a sensor whose signal depends on the distance to nearby sources.*

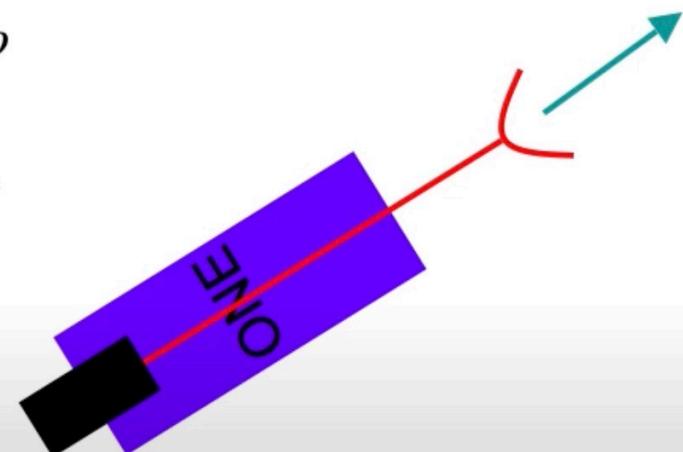
*Motion is always forward except for perturbations.*

*Species*

*1a – excitatory connection*

*1b – inhibitory connection*

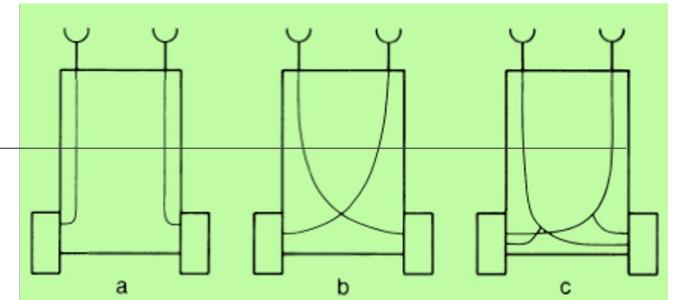
*1r – sources are other vehicles*





**ROBOTICS LAB**

## VEICOLO 2



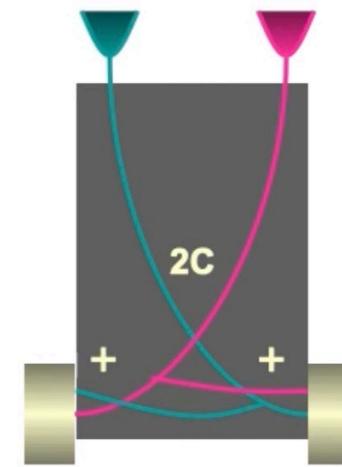
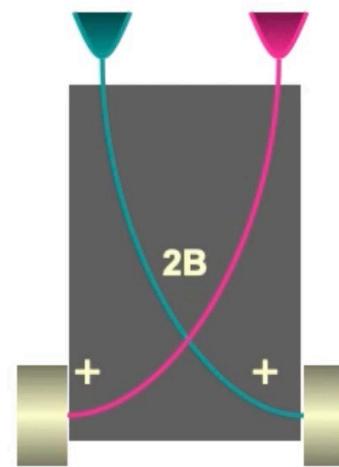
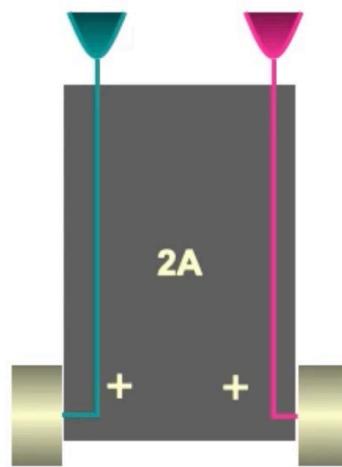
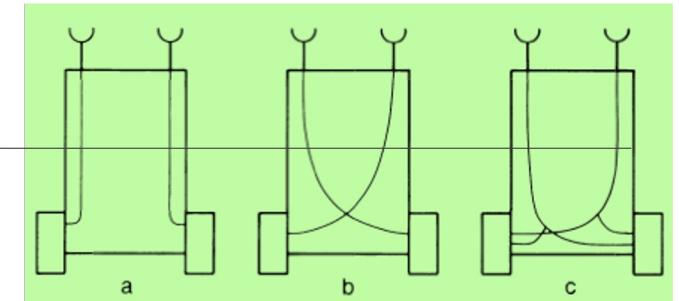
VEICOLO 2a	VEICOLO 2b	VEICOLO 3a	VEICOLO 3b
COLLEGAMENTO ECCITATORE DIRETTO	COLLEGAMENTO ECCITATORE INCROCIATO	COLLEGAMENTO INIBITORE DIRETTO	COLLEGAMENTO INIBITORE INCROCIATO
PAURA	AGGRESSIVITÀ	AMORE	CURIOSITÀ
il robot evita la fonte luminosa	il robot si dirige verso la fonte luminosa sino a sbatterci contro	il robot tende verso la fonte luminosa e si ferma in prossimità della luce	il robot tende verso la fonte luminosa, ma giunto vicino alla luce la sfugge

PROF. ROSARIO SORBELLO



ROBOTICS LAB

## VEICOLO 2



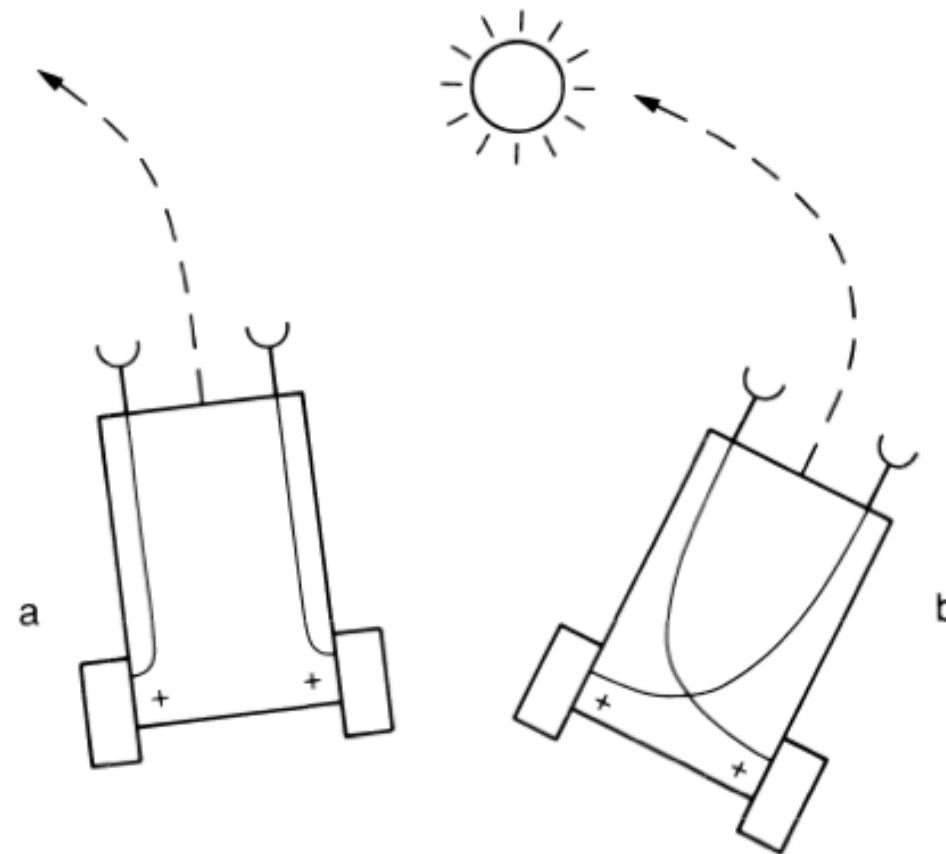
*Description:* With two motors and two sensors; otherwise like vehicle 1. The connections differ in 2A, 2B, 2C

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## PAURA E AGGRESSIVITÀ

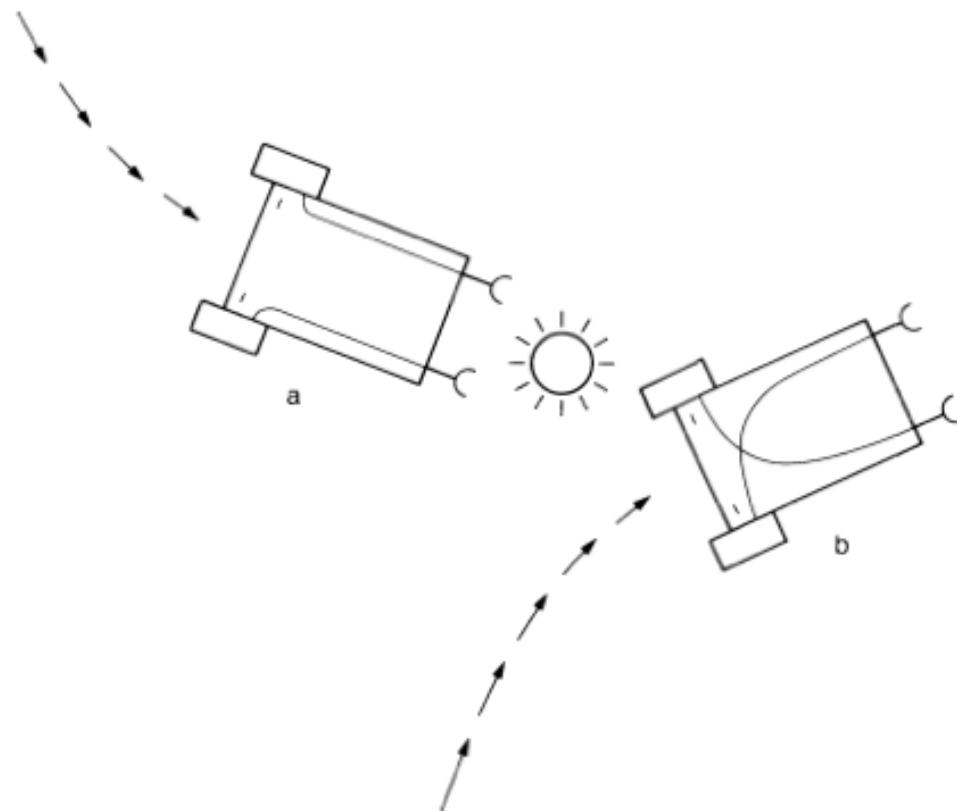


PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## ATTRAZIONE



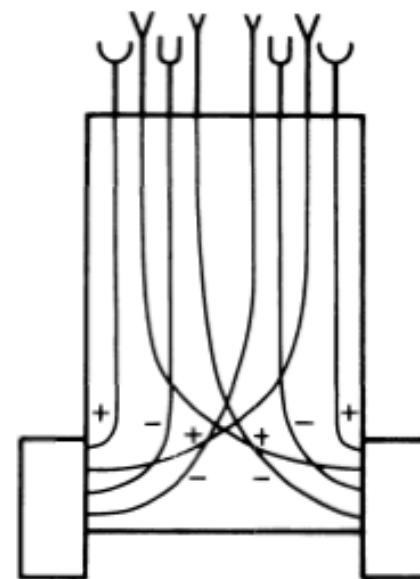
PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

---

## VEICOLO MULTI-SENSORIALE

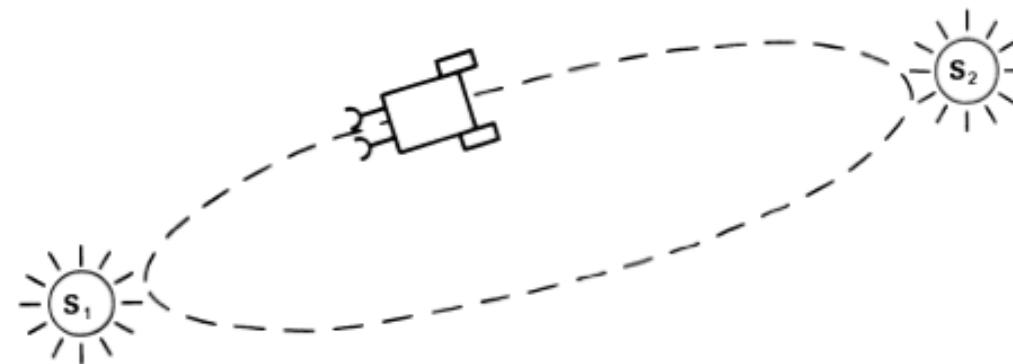


PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## TRAIETTORIE COMPLESSE

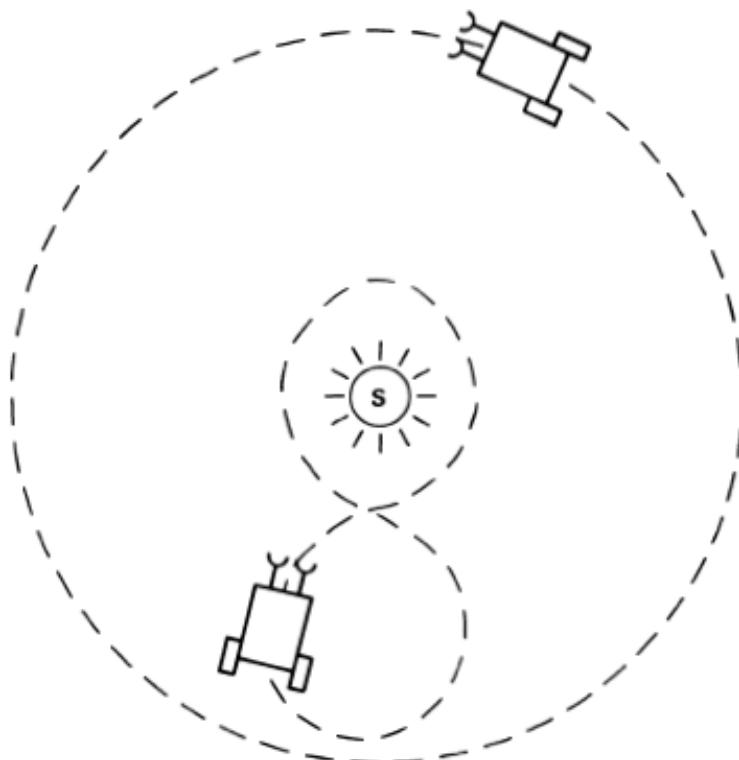


PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## TRAIETTORIE COMPLESSE



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## DEFINIZIONE DI ROBOT

---

- “A robot is a re-programmable, multi-functional, manipulator designed to move material, parts, or specialized devices through variable programmed motions for the performance of a task” (Robotics Industry Association)
- “Robotics is the intelligent connection of perception to action” (Brady)

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## DEFINIZIONE OPERATIVA DI ROBOT

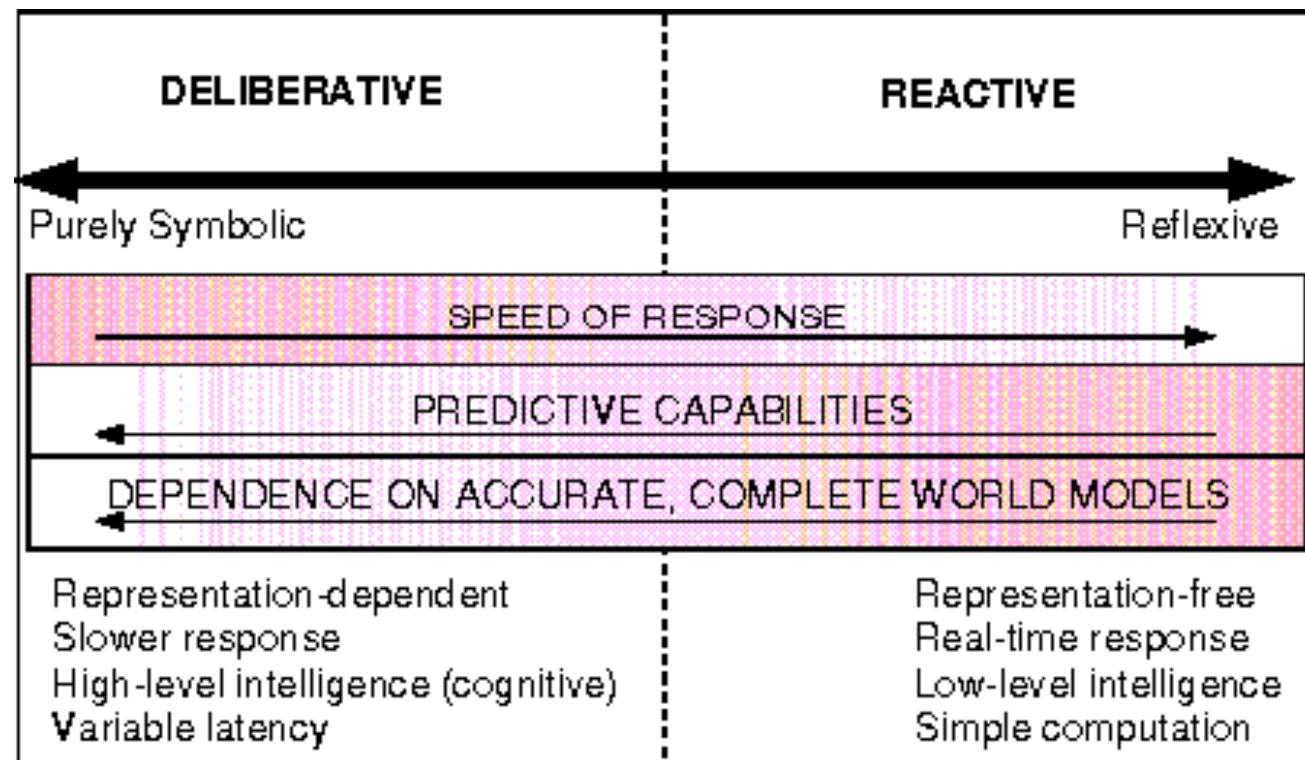
---

- Un robot intelligente è una macchina in grado di estrarre informazioni dal suo ambiente e di utilizzare la conoscenza sul mondo per agire in sicurezza in modo utile e significativo

PROF. ROSARIO SORBELLO



## SPETTRO DEL CONTROLLO DEL ROBOT



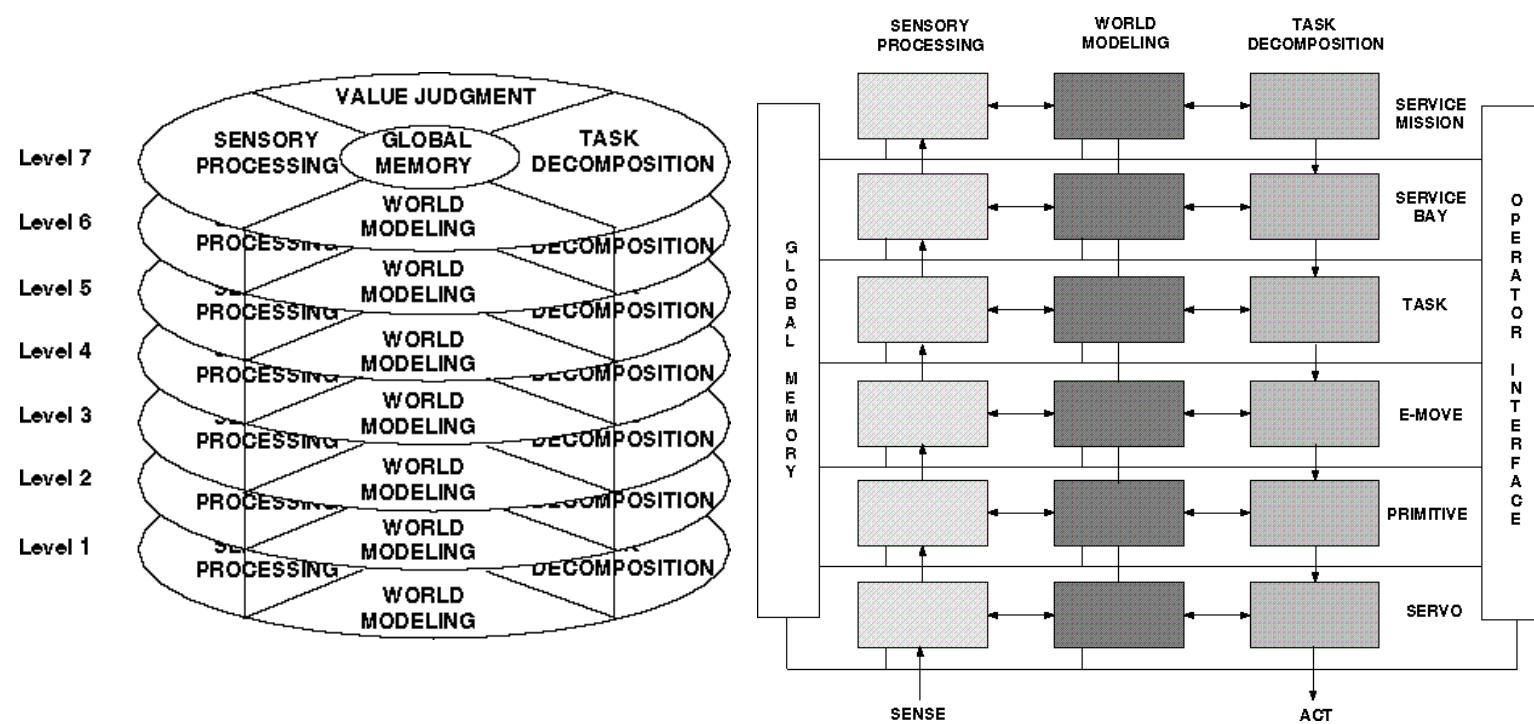
PROF. ROSARIO SORBELLO



- Struttura gerarchica
  - Chiara suddivisione di funzionalità
  - Relativa a strutture economiche/militari
- Comunicazione e controllo predeterminati e predicibili
- L'alto livello genera i sotto obiettivi per i livelli più bassi
- Basati su rappresentazioni simboliche
- Orizzonte spaziale e temporale varia solo gerarchicamente



## TEORIE DI ALBUS SULL'INTELLIGENZA

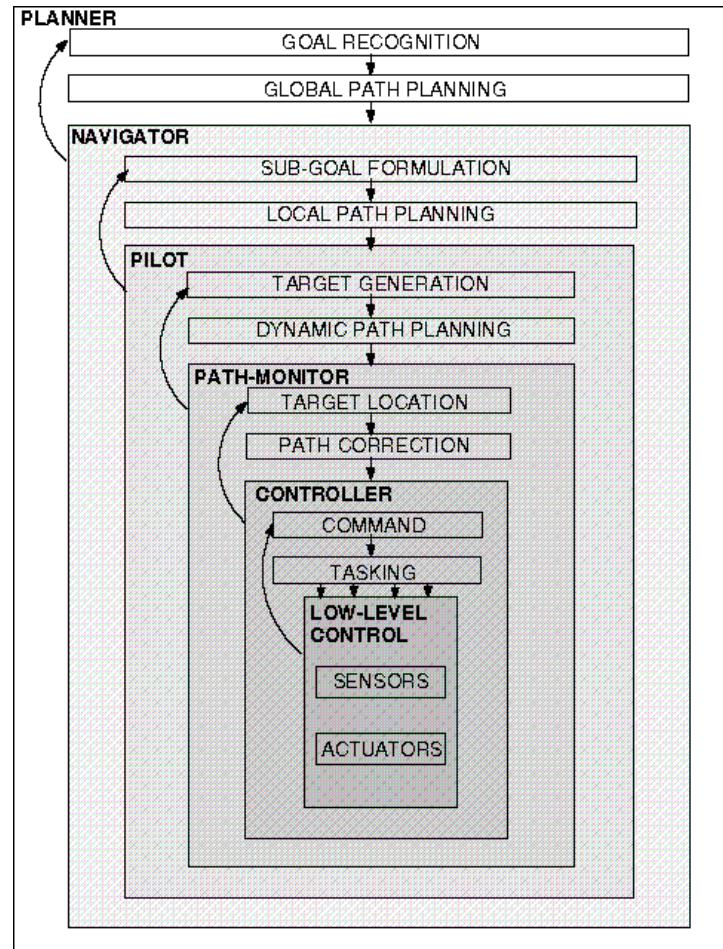


PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

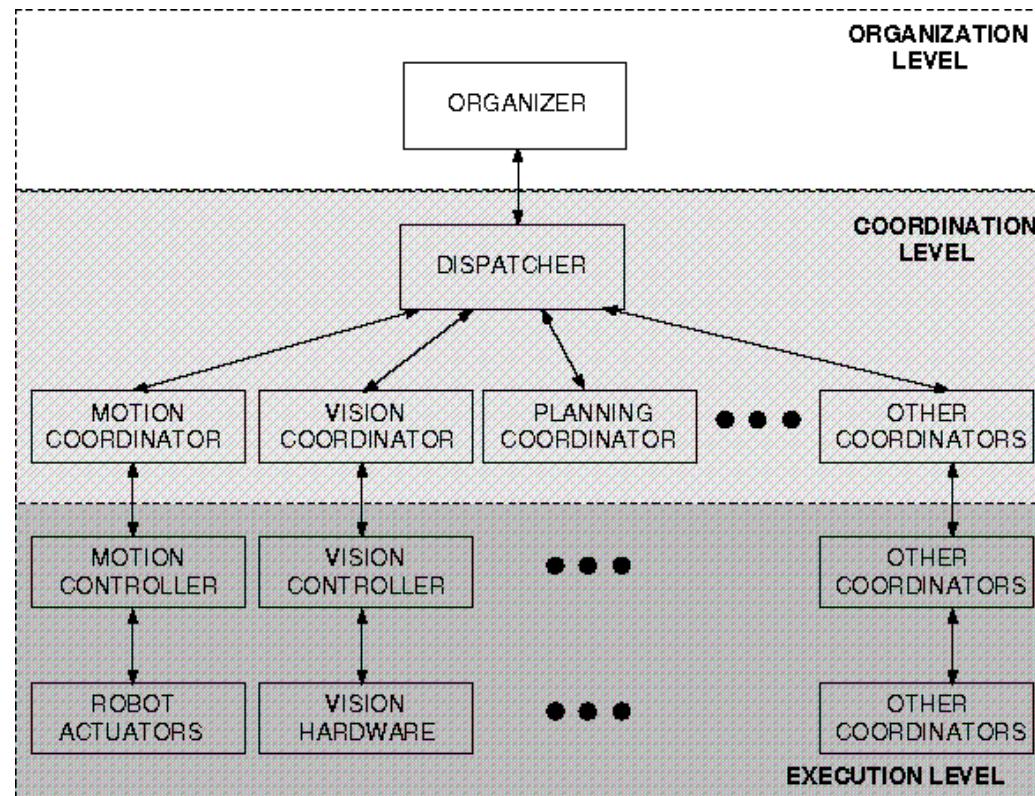
## CONTROLLO GERARCHICO DI MEYSTER



PROF. ROSARIO SORBELLO



# CONTROLLO GERARCHICO DI SARIDIS



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## SISTEMI REATTIVI

---

IL CONTROLLO REATTIVO È UNA TECNICA PER EFFETTUARE UN ACCOPPIAMENTO STRETTO TRA PERCEZIONE E AZIONE, TIPICAMENTE NEL CONTESTO DI COMPORTAMENTI MOTORI, PER PRODURRE RISPOSTE VELOCI IN AMBIENTI DINAMICI E NON STRUTTURATI

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## COMPORTAMENTO INDIVIDUALE E ATTENZIONE

---

- *Comportamento individuale*: una coppia stimolo/risposta per una data situazione ambientale modulata dall'attenzione e determinata dall'intenzione
- *Attenzione*: genera le priorità dei compiti e focalizza le risorse sensoriali; è determinata dal corrente contesto ambientale

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## INTENZIONE E COMPORTAMENTO EMERGENTE

---

- *Intenzione*: determina quale insieme di comportamenti dovrebbe essere attivo in base agli scopi e agli obiettivi del robot
- *Comportamento emergente*: il comportamento globale del robot come conseguenza dell'interazione dei singoli comportamenti attivi

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## COMPORTAMENTO PURAMENTE REATTIVO (RIFLESSO)

---

Comportamento generato da comportamenti reattivi cablati con stretta interazione sensore-effettore, dove l'informazione sensoriale non è persistente e non sono usati modelli del mondo.

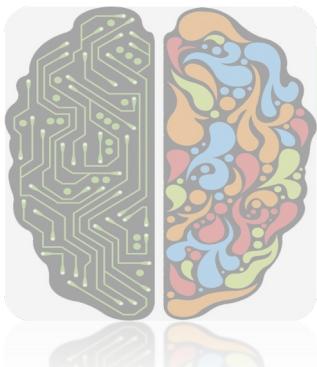
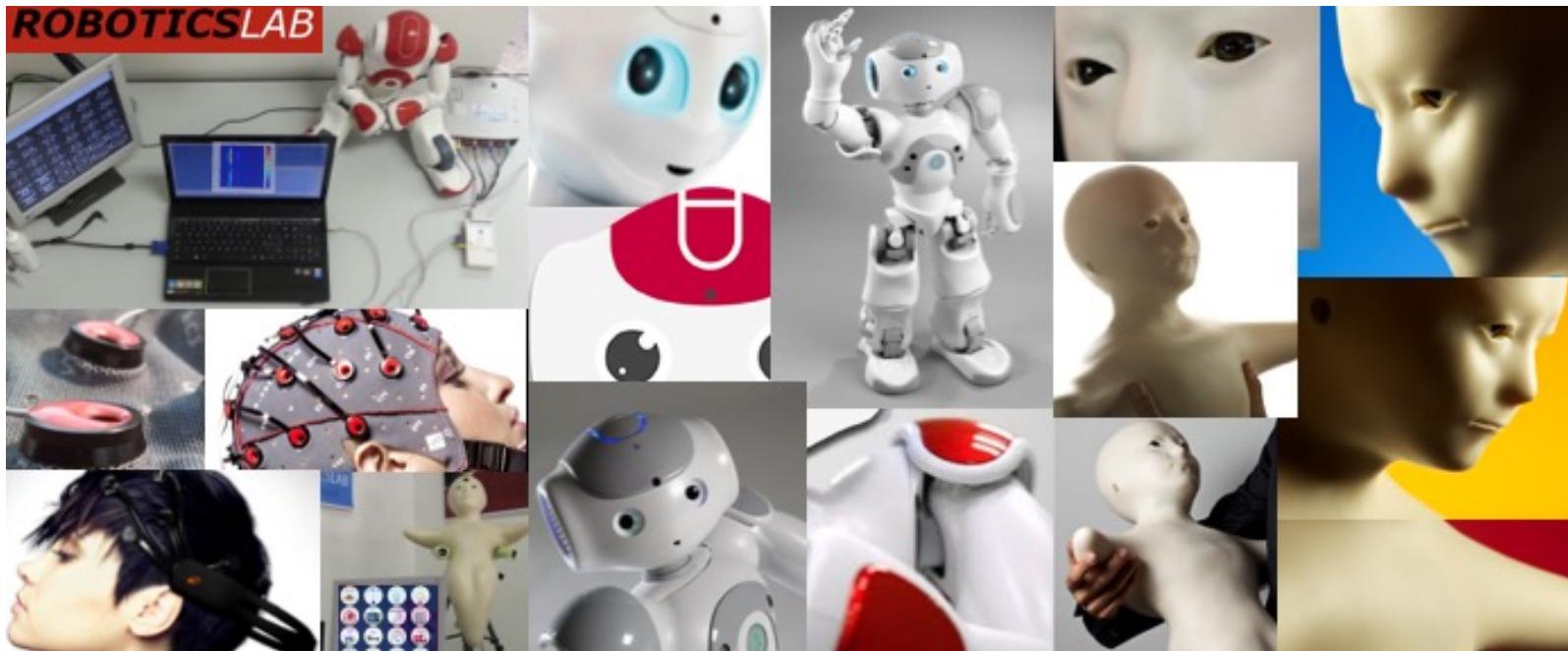
PROF. ROSARIO SORBELLO



- *Situatedness* - il robot è circondato dal mondo esterno, nessuna rappresentazione astratta
- *Embodiment* - il robot ha un corpo fisico nel mondo
- *Emergence* - intelligenza come risultato dell'interazione del robot con in mondo esterno



OSAKA UNIVERSITY

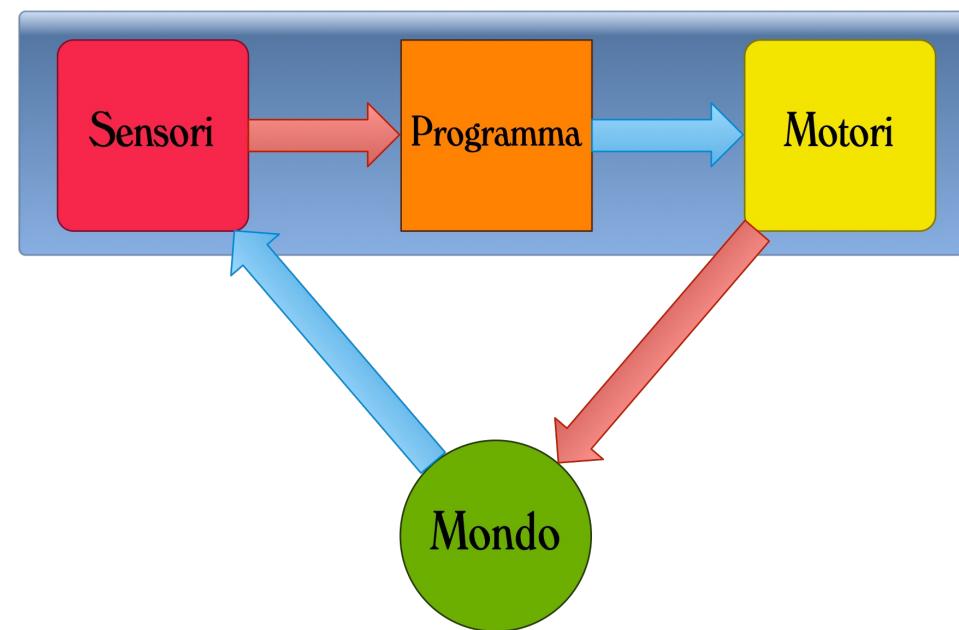
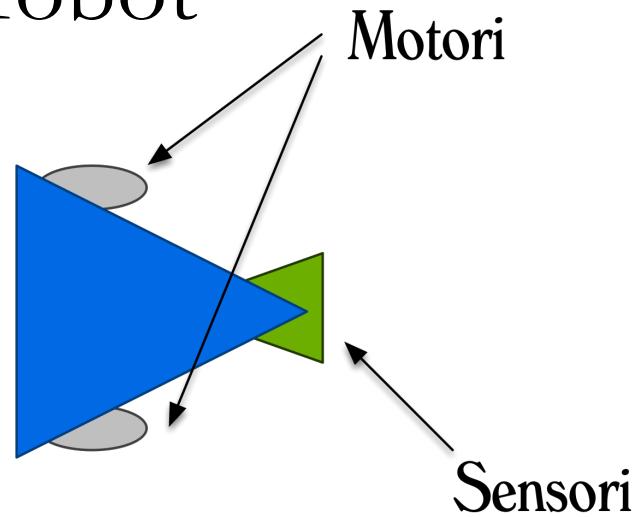


SENSORI

PROF. ROSARIO SORBELLO



Il robot





**ROBOTICS LAB**

## SENSORI PROPRIOCETTIVI

---

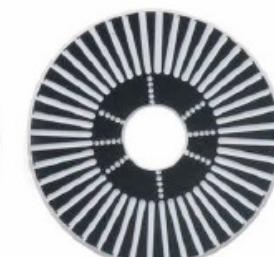
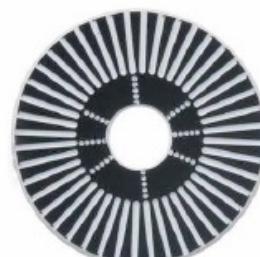
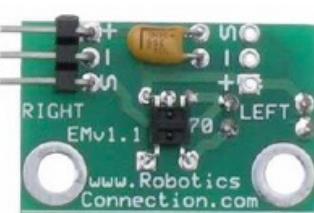
- PERCEZIONE DELLO STATO DEL ROBOT
  - TEMPERATURA
  - POSIZIONE (GPS)
  - VELOCITÀ, ACCELERAZIONE, ECC.

PROF. ROSARIO SORBELLO



## ODOMETRIA

- Deriva da  $l = 2\pi r$
- Noto il numero di rotazioni delle ruote, si risale allo spazio percorso
- Wheel encoders: segnalano la transizione da bianco a nero



PROF. ROSARIO SORBELLO

## MODELLO ODOMETRICO

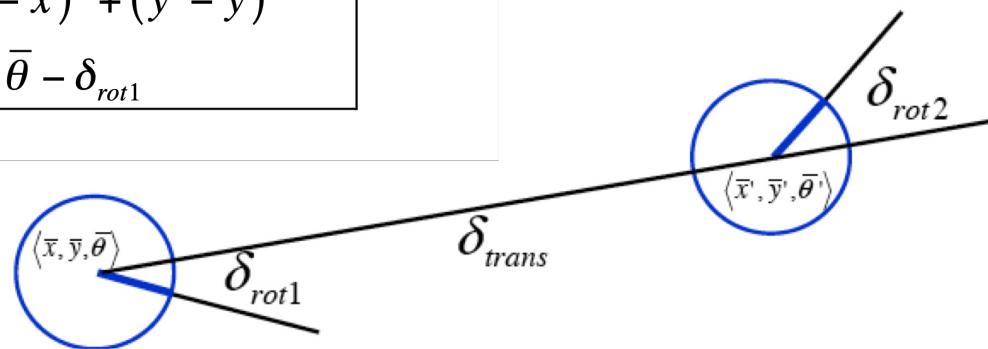
Il robot si muove da  $(\bar{x}, \bar{y}, \bar{\theta})$  a  $(\bar{x}', \bar{y}', \bar{\theta}')$

$$u = (\delta_{rot1}, \delta_{trl}, \delta_{rot2})$$

$$\delta_{rot1} = \text{atan} 2(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

$$\delta_{trl} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$



PROF. ROSARIO SORBELLO



- Estende l'arcotangente e aggiusta i segni

$$\text{atan}2(y, x) = \begin{cases} \text{atan}(y / x) & \text{if } x > 0 \\ \text{sgn}(y)(\pi - \text{atan}(|y / x|)) & \text{if } x < 0 \\ 0 & \text{if } x = y = 0 \\ \text{sgn}(y)\pi / 2 & \text{if } x = 0, y \neq 0 \end{cases}$$



**ROBOTICS LAB**

## DEAD RECKONING

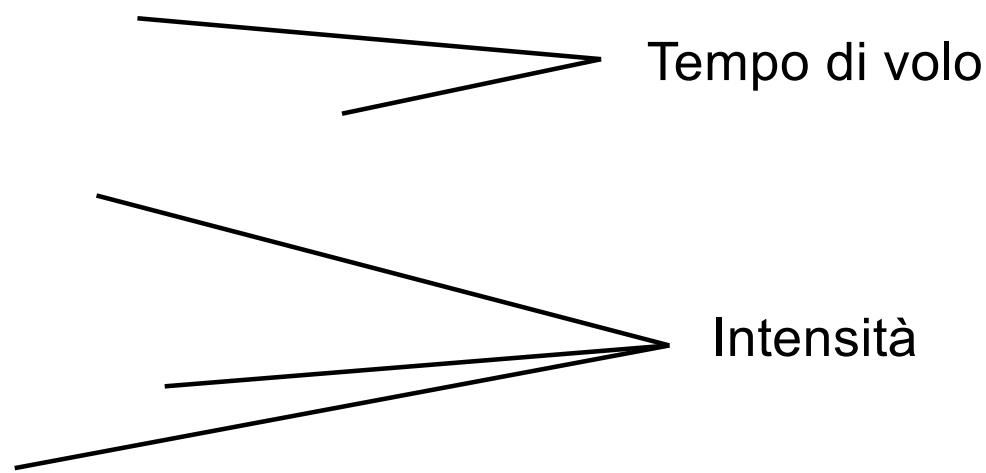
---

- Deduced reckoning
- Si deriva la posizione del robot in base alle velocità applicate e al tempo
- Ha origine dalla navigazione delle navi

PROF. ROSARIO SORBELLO



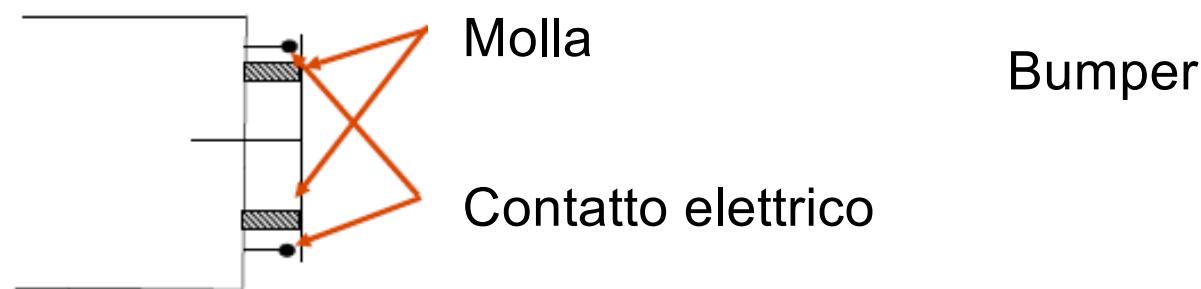
- Percezione dell'ambiente
- Attivi
  - Ultrasuoni
  - Laser range finder
  - Infrarossi
- Passivi
  - Telecamera
  - Tattili





**ROBOTICS LAB**

## SENSORI TATTILI



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## SENSORI AD ULTRASUONI

---

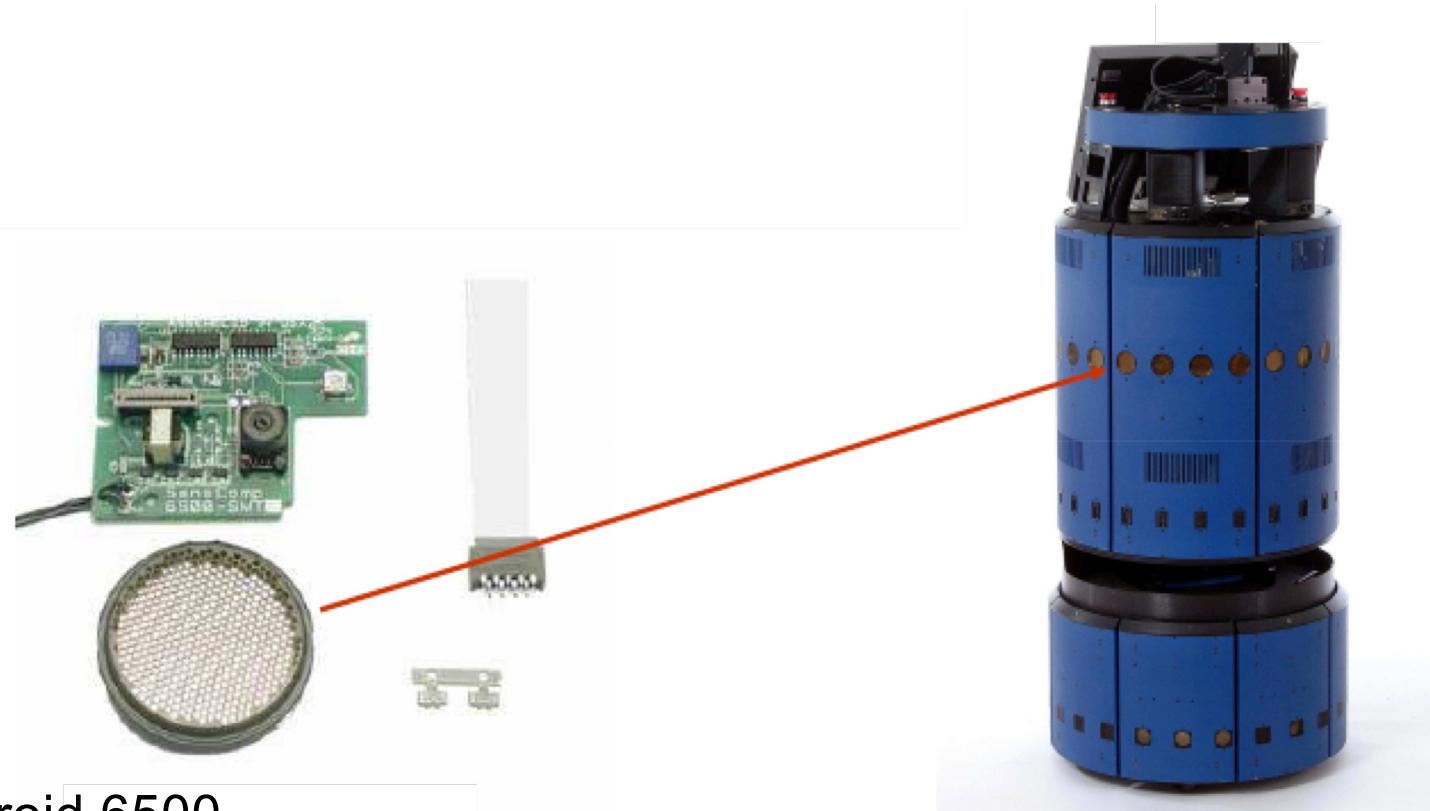
- Emettono un segnale ad ultrasuoni
- Aspettano l'eco del segnale
- Calcolano la distanza in base al tempo di volo

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## SENSORI AD ULTRASUONI



Polaroid 6500

PROF. ROSARIO SORBELLO



## TEMPO DI VOLO



$$d = \frac{1}{2} v \Delta t$$

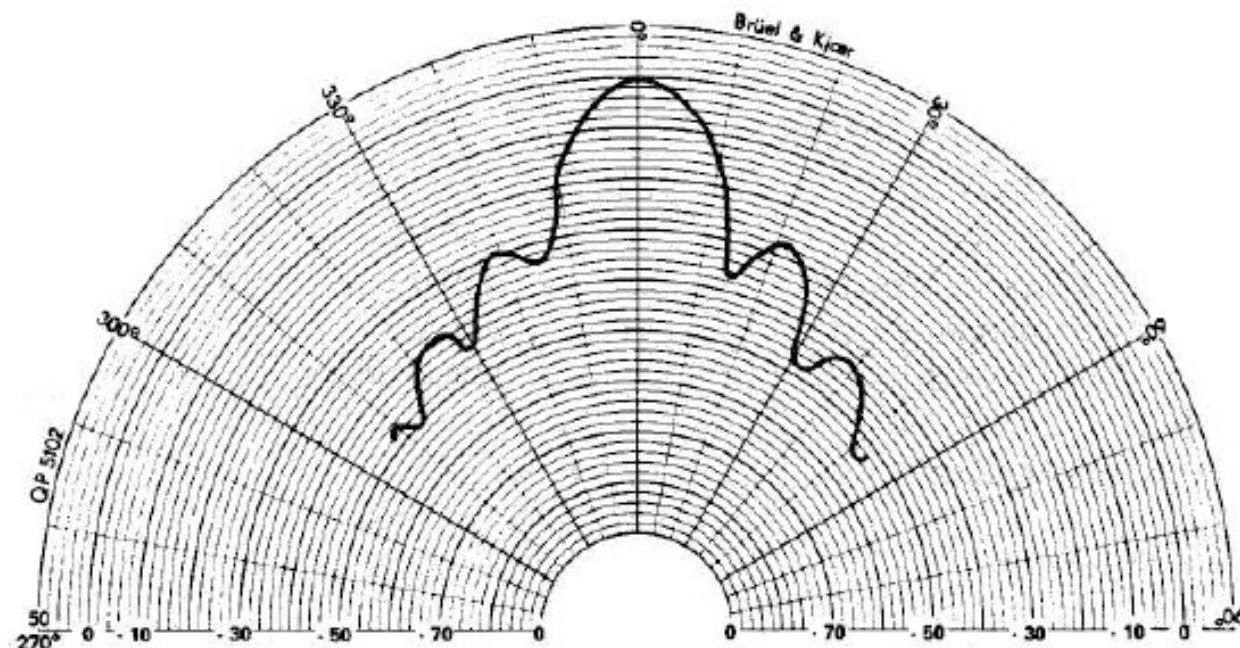
v: velocità del segnale

$\Delta t$ : intervallo di tempo tra l'invio del segnale e il ritorno dell'eco



**ROBOTICS LAB**

## PROPRIETA' DEI SENSORI AD ULTRASUONO



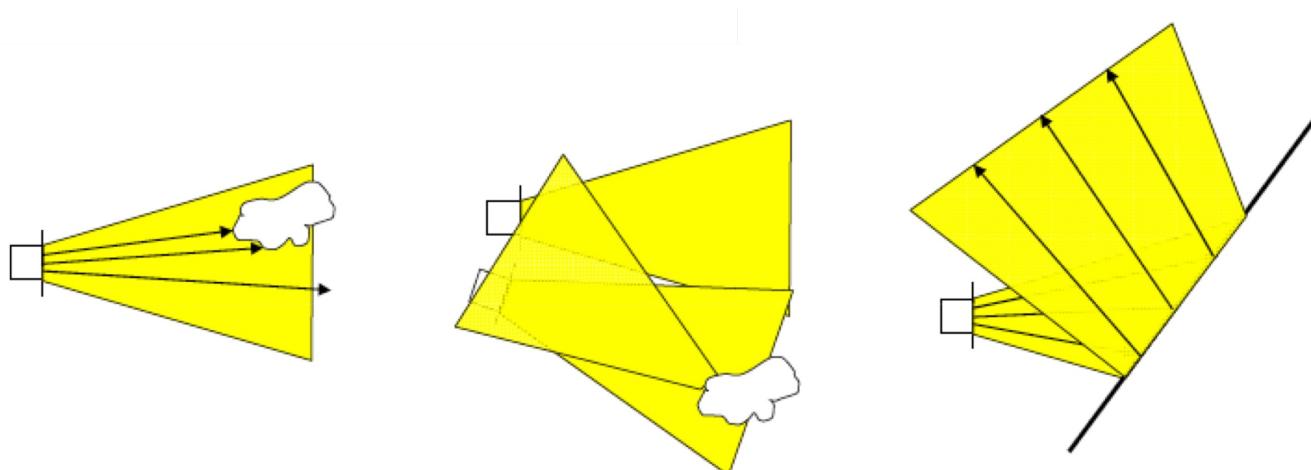
Profilo del sensore Polaroid

PROF. ROSARIO SORBELLO



## CAUSE DI ERRORE

- Angolo di apertura
- Crosstalk
- Riflessioni speculari



PROF. ROSARIO SORBELLO



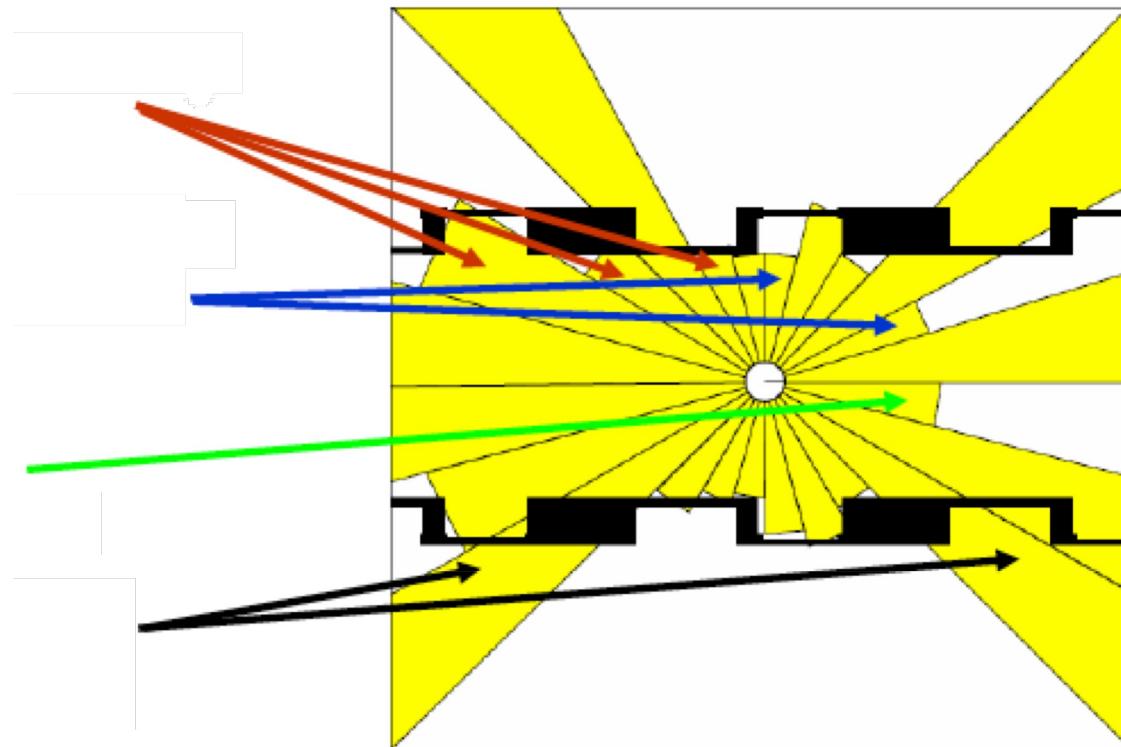
## SCANSIONE AD ULTRASUONI

Raggi riflessi da ostacoli

Raggi riflessi da persone / crosstalk

Misure casuali

Misure oltre il max



PROF. ROSARIO SORBELLO

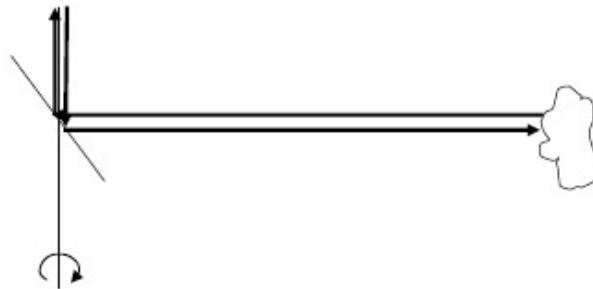


- Dato un angolo di apertura di 15° sono necessari 24 sensori per coprire i 360° attorno al robot
- Il massimo intervallo di misura è tipicamente 10m
- Il tempo di volo è quindi  $2*10/330=0.06\text{sec}$
- Una scansione completa richiede 1.45sec
- I sensori devono lavorare in parallelo
- Aumenta il rischio del crosstalk



**ROBOTICS LAB**

## SCANNER LASER



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## PROPRIETA'

---

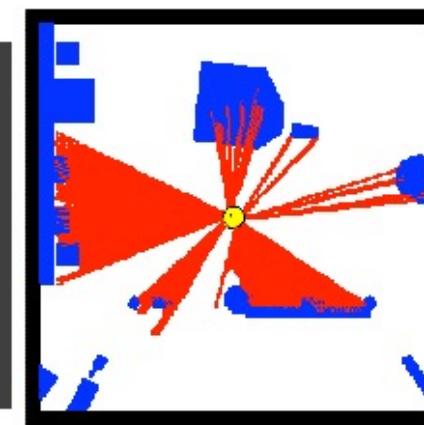
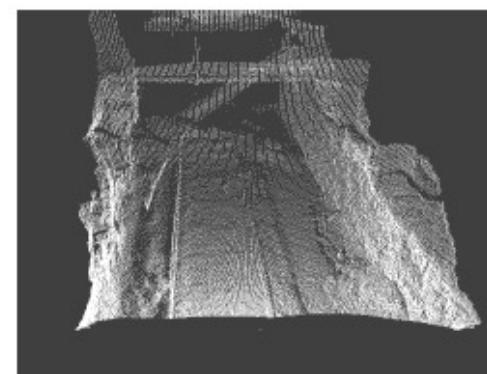
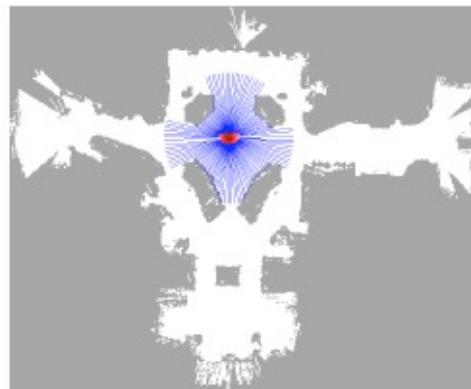
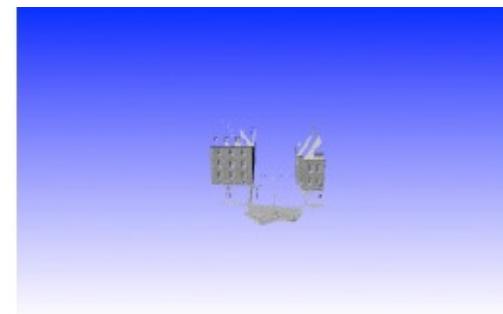
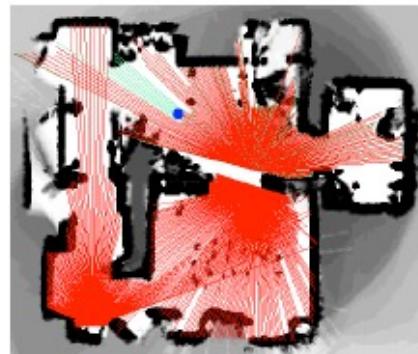
- PRECISIONE
- SICUREZZA
- E' PIÙ LENTO DEL SENSORE AD ULTRASUONI

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

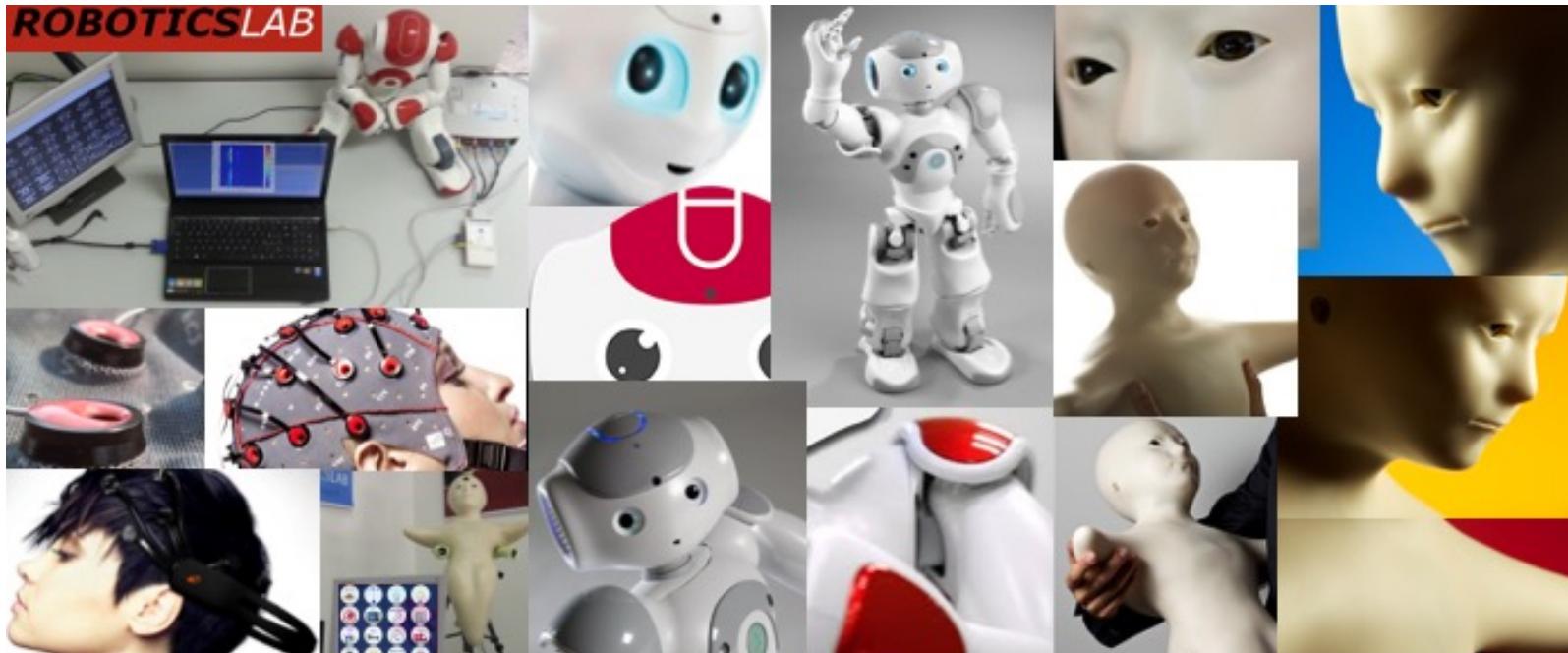
## SCANSIONI LASER



PROF. ROSARIO SORBELLO



OSAKA UNIVERSITY



ATR

g-tec

## ARCHITETTURE BASATE SUI COMPORTAMENTI

PROF. ROSARIO SORBELLO

3° LEZIONE - PALERMO 17 OTTOBRE 2022



**ROBOTICS LAB**

## SISTEMA REATTIVO

---

- UN SISTEMA REATTIVO EFFETTUÀ UN ACCOPPIAMENTO STRETTO TRA LA PERCEZIONE E L'AZIONE SENZA L'USO DI RAPPRESENTAZIONI ASTRATTE O STORIE TEMPORALI

PROF. ROSARIO SORBELLO



- COMPORTAMENTI COME BLOCCHI DI BASE
- NON SI UTILIZZANO RAPPRESENTAZIONI ASTRATTE DI CONOSCENZA
- SI UTILIZZANO SPESSO I MODELLI ISPIRATI DALLA NATURA
- ESTREMA MODULARITÀ PER AGGIUNGERE ALTRI COMPORTAMENTI

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## ESEMPIO DI SISTEMA DI NAVIGAZIONE PER STUDENTE DA UNA CLASSE AD UNA ALTRA

---

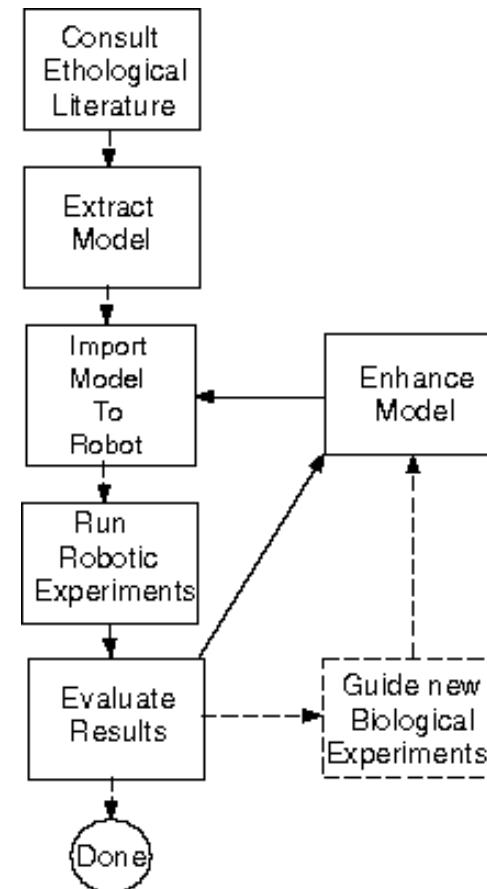
- ANDARE A DESTINAZIONE DA PUNTO DI PARTENZA
- NON SCONTRARSI CON NESSUN OSTACOLO
- NEGOZIARE CON ALTRI STUDENTI CHE HANNO SIMILI OBIETTIVI
- AFFRONTARE I CAMBIAMENTI E ATTUARE LE SCELTE NECESSARIE
- MOLTO DIFFICILE IN AMBIENTE NON CONTROLLABILE O FACILMENTE PREVEDIBILE
- SISTEMA BASATO SUI COMPORTAMENTI MEGLIO DELLA NAVIGAZIONE PURA SE OGNI COMPORTAMENTO OBIETTIVO E SI COORDINANO TRA DI LORO

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## PROGETTAZIONE GUIDATA DALL'ETOLOGIA



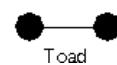
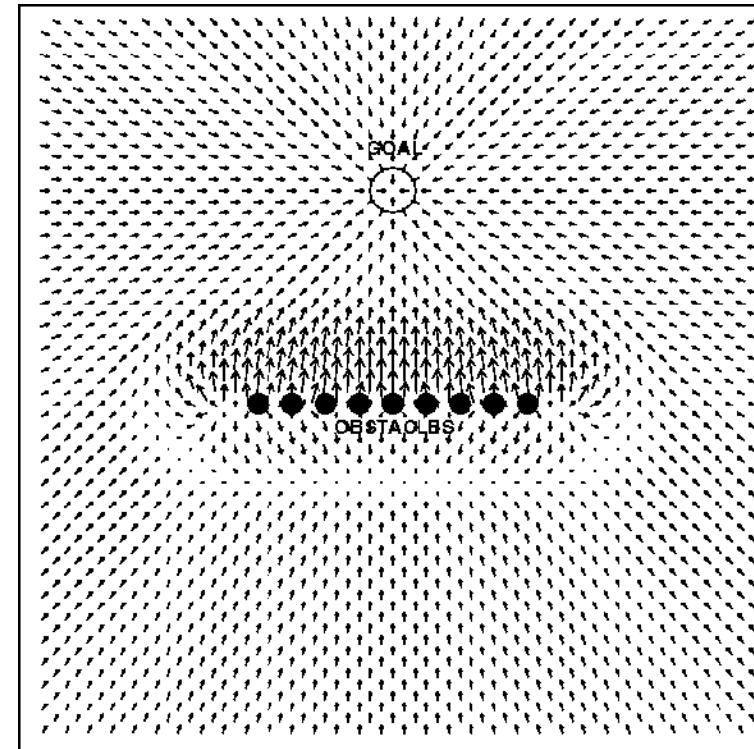
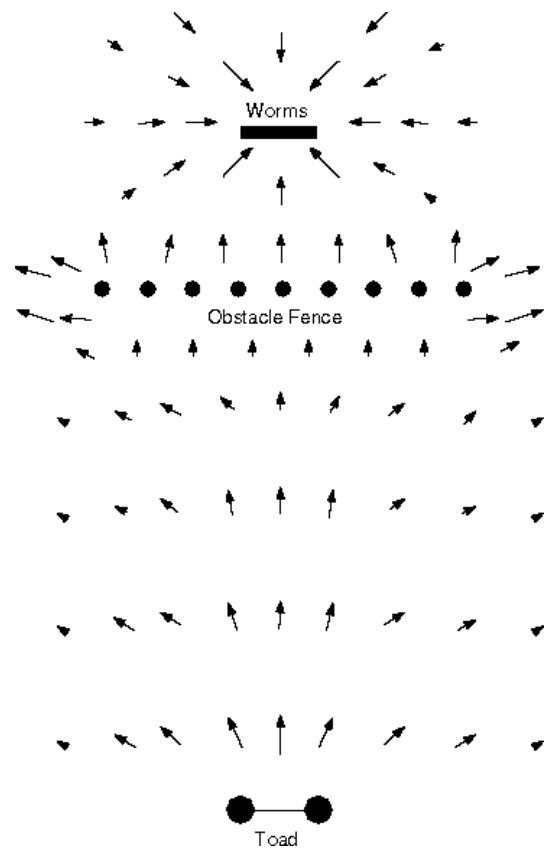
PROF. ROSARIO SORBELLO



ROBOTICS LAB

## MODELLO DI NAVIGAZIONE DELLA RANA

I VETTORI RAPPRESENTANO LE DIREZIONI DELL'ANIMALE IN OGNI PUNTO DELLO SPAZIO



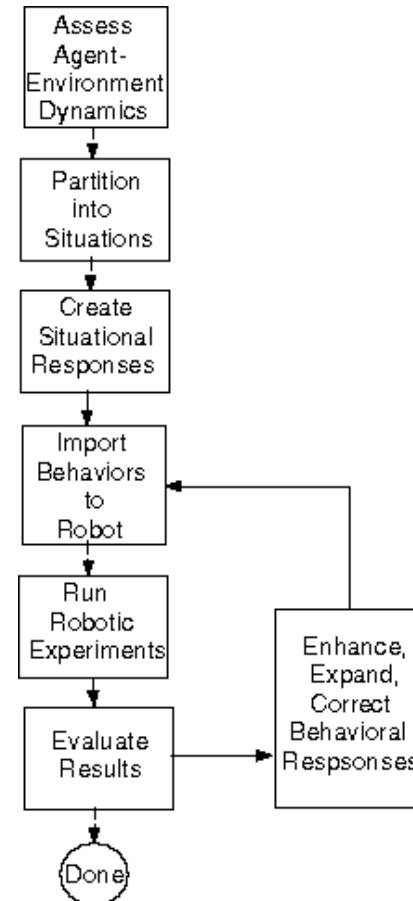
METODO DEL CAMPO POTENZIALE MODIFICATO  
PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## PROGETTAZIONE BASATA SU ATTIVITA' SITUATA

LE AZIONI DEL ROBOT SONO PREDETERMINATE DALLE SITUAZIONI IN CUI SI TROVA, IL PROBLEMA DELLA PERCEZIONE SI RIDUCE AL RICONOSCIMENTO DELLA SITUAZIONE IN CUI SI TROVA IL ROBOT E ALLA SCELTA DI UN'AZIONE DA INTRAPRENDERE.



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## SISTEMI BASATI SU ATTIVITA' SITUATA

---

- Pengi da Agre e Chapman (Indexical-functional aspects)

Una frase per ogni situazione

Esempio: the-block-i-need-to-kick-at-the-enemy-is-behind-me

PROF. ROSARIO SORBELLO



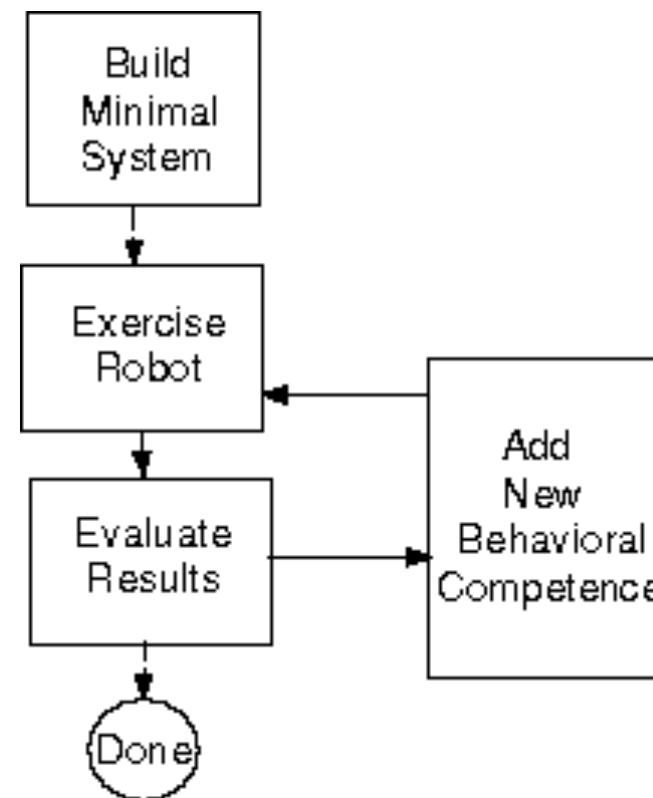
**ROBOTICS LAB**

## SISTEMI BASATI SU ATTIVITA' SITUATA

---

- Basata su videogiochi
- Microcomportamenti
- Piani Universali
- RAPs
  - Firby
  - Reactive Action Packages

PROF. ROSARIO SORBELLO



PROF. ROSARIO SORBELLO



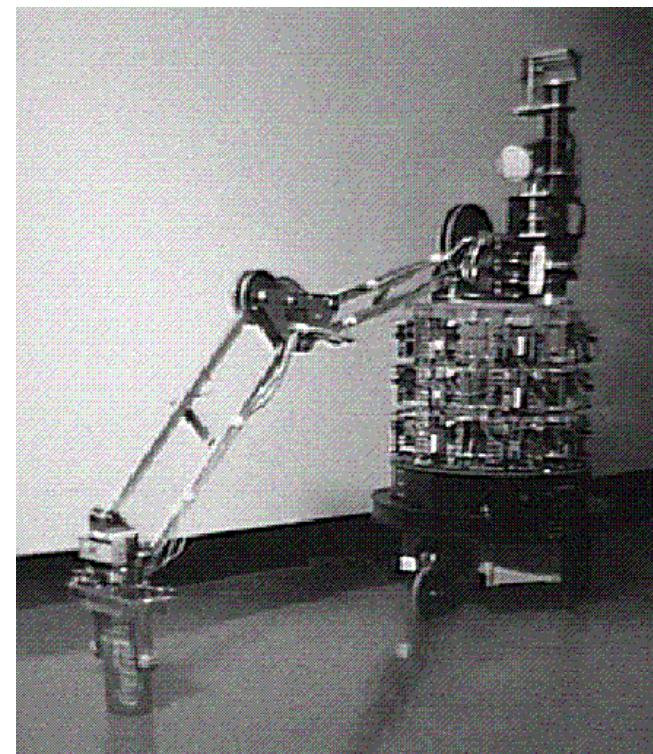
**ROBOTICS LAB**

## ROBOT PROGETTATI MEDIANTE ESPERIMENTI

ROBOT ESAPODO



HERBERT CERCA LATTINA



PROF. ROSARIO SORBELLO



## CLASSI DI COMPORTAMENTI

---

- ESPLORATIVI
- ORIENTATI AL GOAL
- PROTETTIVI/AVVERSI
- SEGUIRE CAMMINI
- POSTURALI
- SOCIALI/COOPERATIVI
- TELEAUTONOMIA
- PERCEZIONE
- CAMMINO
- SPECIFICHE DEL MANIPOLATORE
- PRESA DELLA MANO ROBOTICA

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## DESCRIZIONE DEI COMPORTAMENTI

---

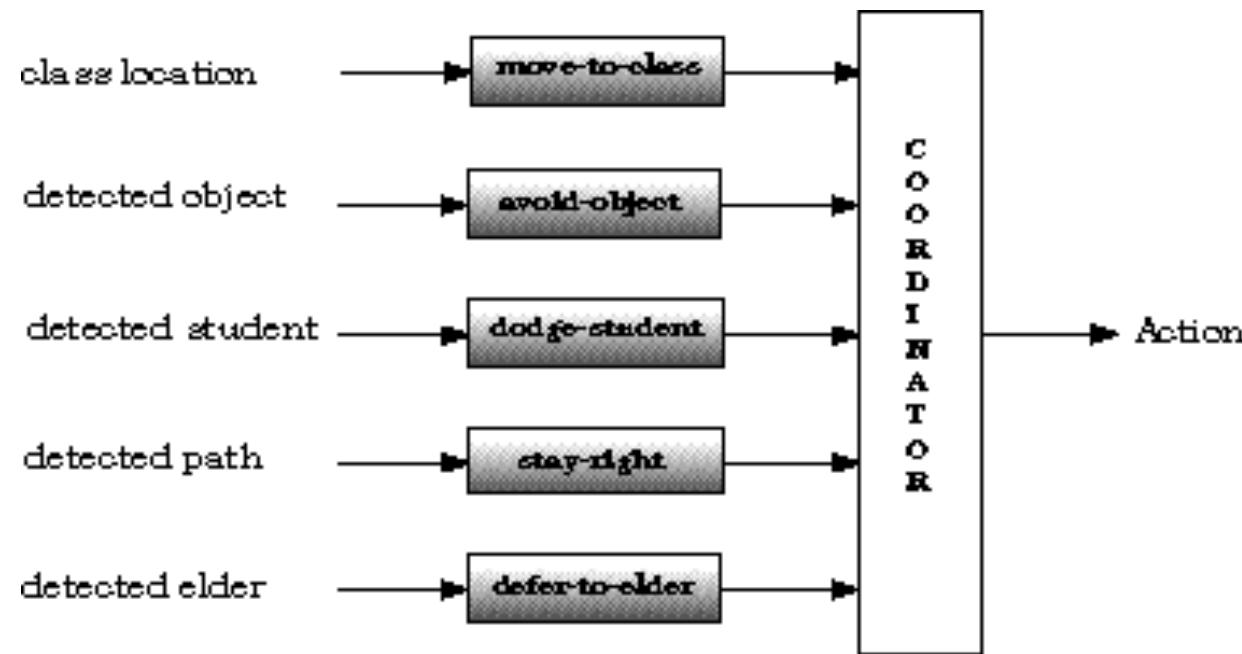
- DIAGRAMMI STIMOLO-RISPOSTA (SR)
- NOTAZIONE FUNZIONALE
- DIAGRAMMI DI ACCETTORI A STATI FINITI (FSA)

PROF. ROSARIO SORBELLO



## DIAGRAMMA SR

OGNI COMPORTAMENTO UNA RISPOSTA PER OGNI STIMOLO



PROF. ROSARIO SORBELLO



coordinate-behaviors [  
move-to-classroom (detect-classroom-location),  
avoid-objects (detect-objects),  
dodge-students (detect-students),  
stay-to-right-on-path (detect-path),  
defer-to-elders (detect-elders)  
] = motor-response

$$B(S)=R$$

PROF. ROSARIO SORBELLO



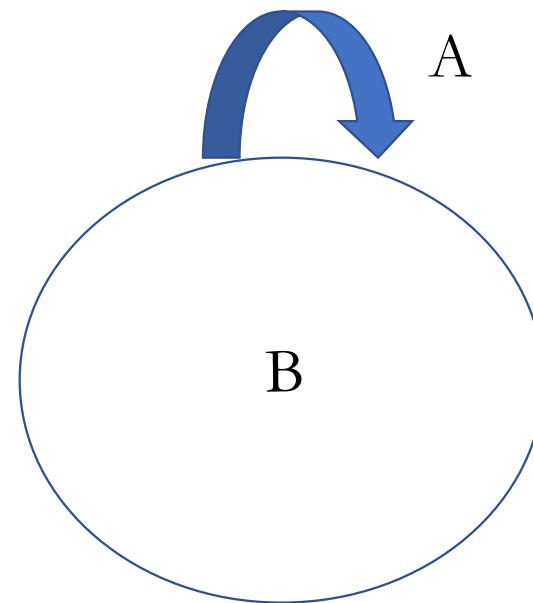
## DIAGRAMMA FSA

ESPLICITO AD OGNI ISTANTE IL COMPORTAMENTO E LA TRANSIZIONE TRA DI LORO

Q COMPORTAMENTI  
DELTA FUNZIONE DI  
TRANSIZIONE

Q<sub>0</sub> COMPORTAMENTO INIZIALE  
F INSIEME DI STATI TERMINALI

FSA = (Q, DELTA, Q<sub>0</sub>, F)



FSA = ( (B), DELTA, B, (B) )

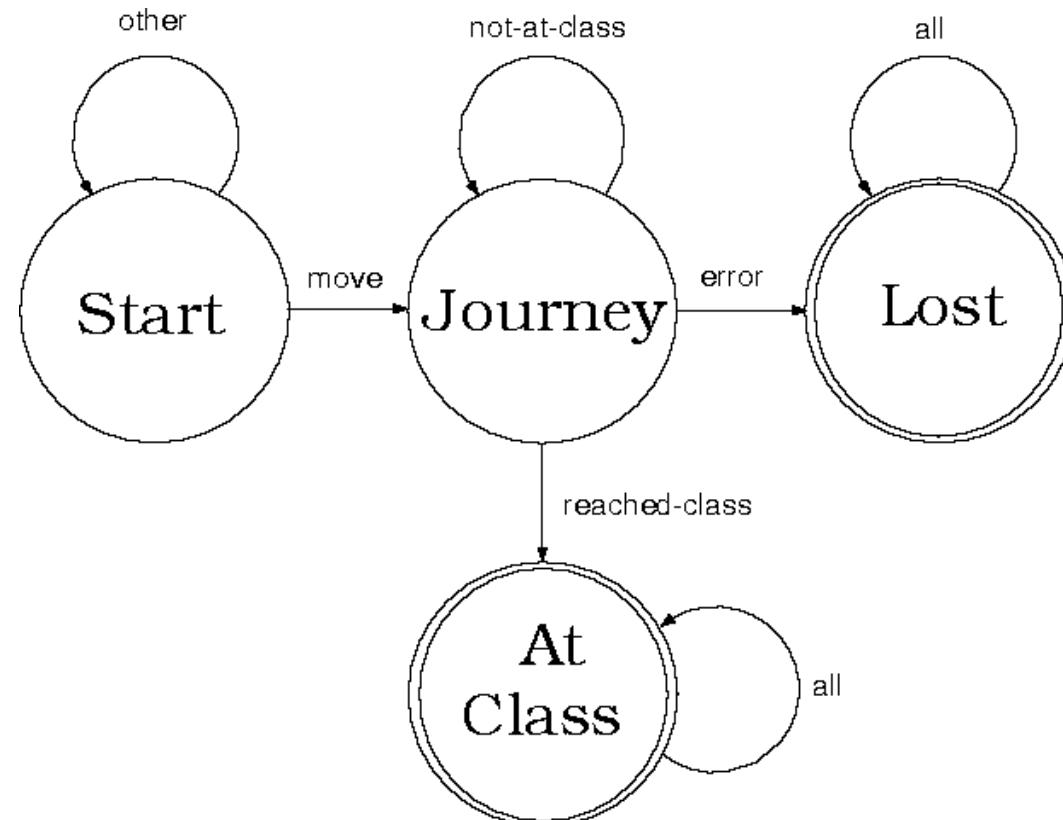
delta	q	input	Delta(q, input)
	b	a	b

PROF. ROSARIO SORBELLO



## DIAGRAMMA FSA

ESPLICITO AD OGNI ISTANTE IL COMPORTAMENTO E LA TRANSIZIONE TRA DI LORO

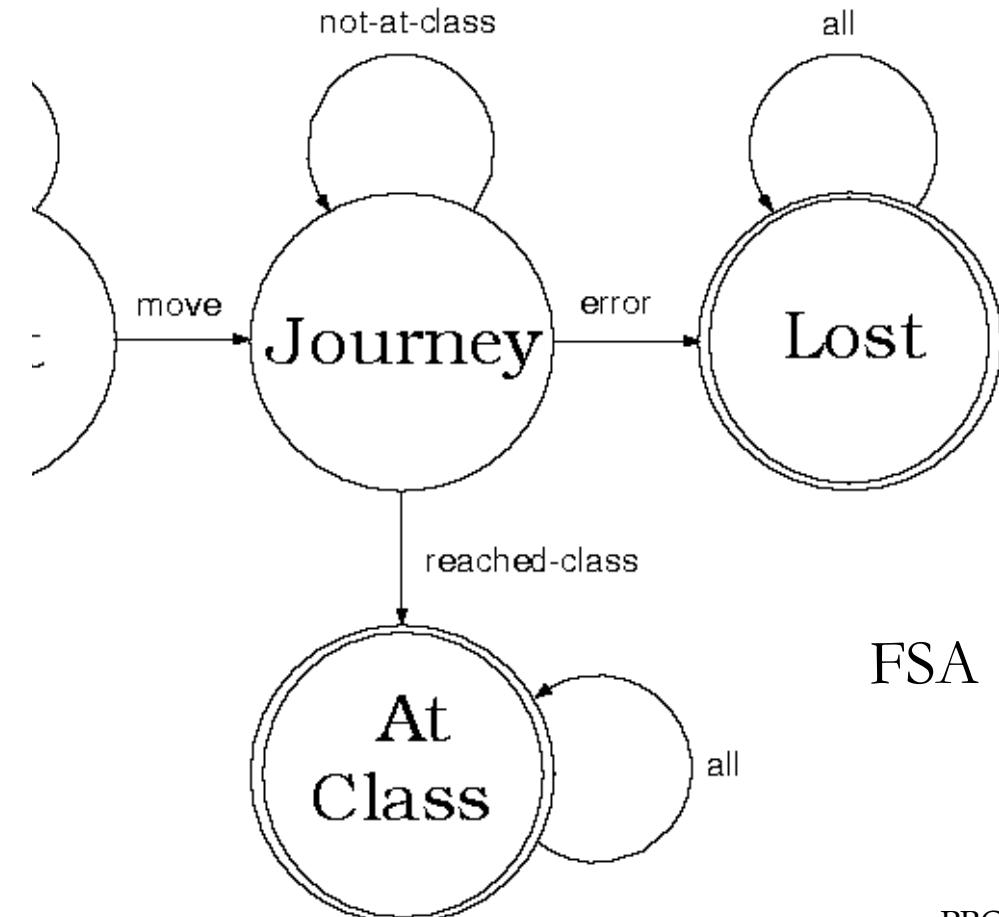


PROF. ROSARIO SORBELLO



ROBOTICS LAB

ESPLICITO AD OGNI ISTANTE IL COMPORTAMENTO



## DIAGRAMMA FSA

Q	INPUT
START	OTHER
START	MOVE
JOURNEY	NOT_AT_CLASS
JOURNEY	ERROR
JOURNEY	REACHED_CLASS
AT_CLASS	ALL
LOST	ALL

$$FSA = (Q, \Delta, Q_0, F)$$

PROF. ROSARIO SORBELLO

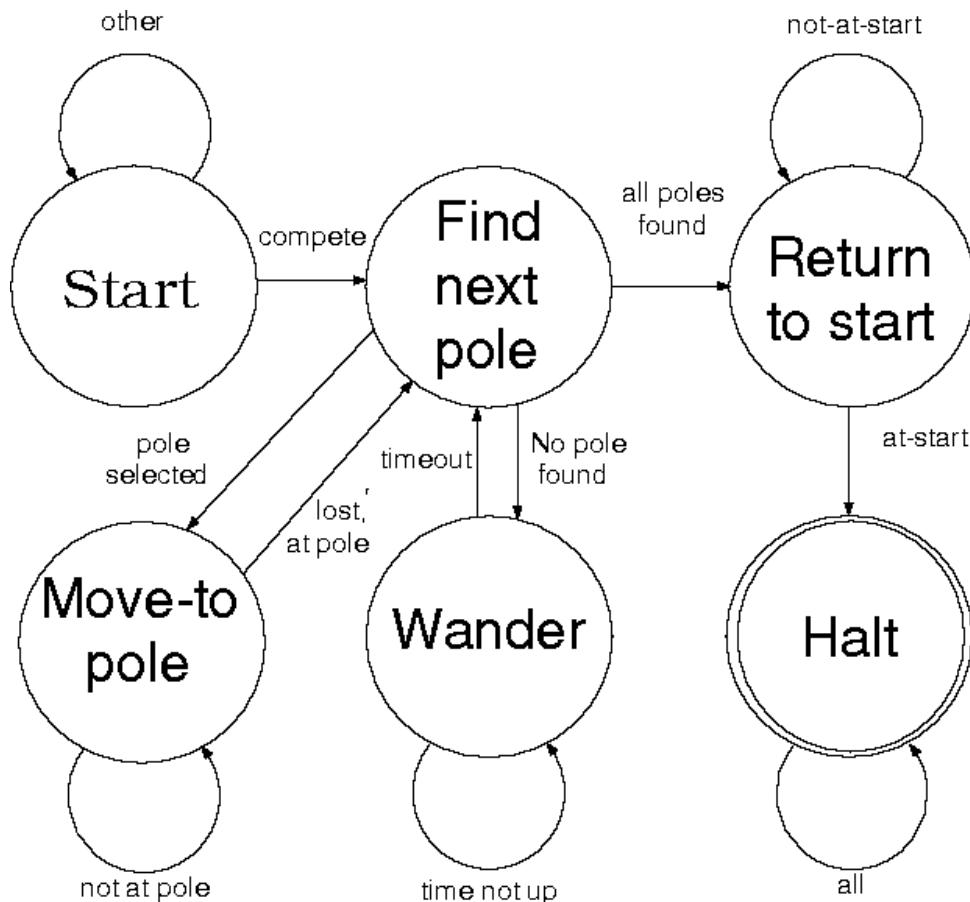
Q COMPORTAMENTI  
Δ FUNZIONE DI  
Q₀ COMPORTAMENTO  
F INSIEME DI STATI T]

$$FSA = (Q, \Delta, Q_0, F)$$

**ROBOTICS LAB**

$$FSA = (\{START, FNP, MTP, WANDER, R\_START, HALT\}, \Delta, START, \{HALT\})$$

## FSA DELLA COMPETIZIONE AAAI



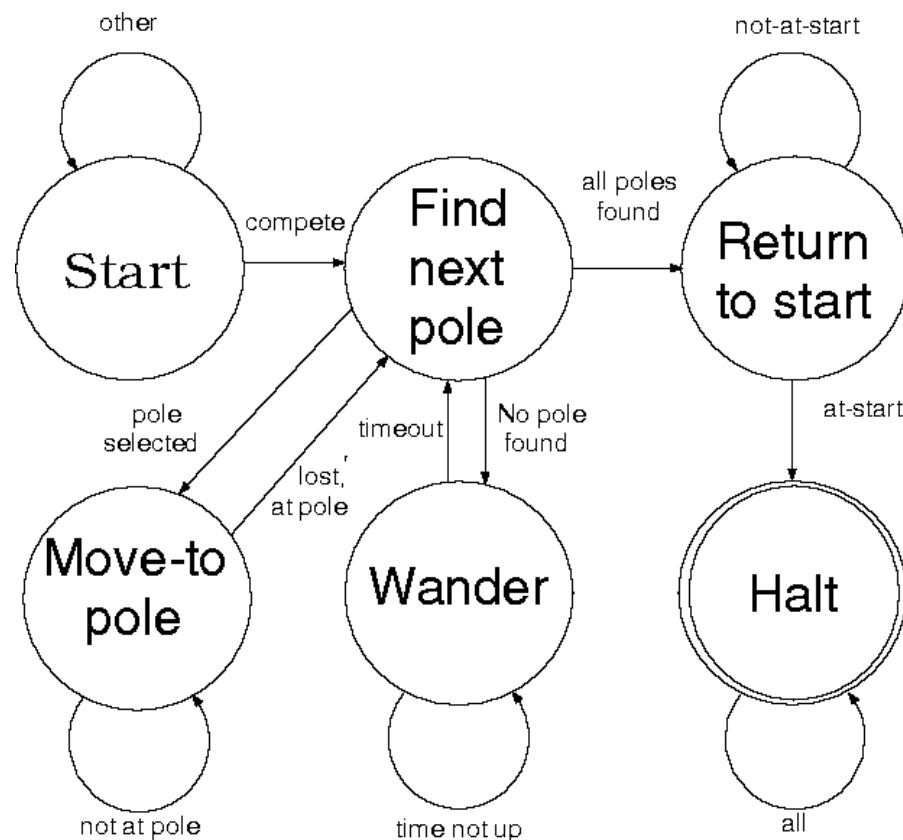
Q	INPUT	$\Delta(Q, INPUT)$
START	OTHER	START
START	COMPETE	FNP
FNP	POLE_SELECTED	MTP
FNP	ALL_POLES_FOUND	R_START
FNP	NO_POLE_FOUND	WANDER
MTP	NOT_AT_POLE	MTP
MTP	LOST_AT_POLE	FNP
WANDER	TIMEOUT	FNP
WANDER	TIME NOTUP	WANDER
R_START	NOT_AT_START	R_START
R_START	AT_START	HALT (STATO FINALE)
HALT	ALL	HALT

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## FSA DELLA COMPETIZIONE AAAI



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## METODI FORMALI

---

- Verifica delle intenzioni del progettista
- Generazione automatica del codice
- Linguaggio comune
- Supporto dell'analisi formale
- Supporto della programmazione ad alto livello

PROF. ROSARIO SORBELLO



- LA RISPOSTA POSSIEDE FORZA E ORIENTAZIONE
- COMPORTAMENTO ESPRESSO COME  $B(S) \rightarrow R$
- LO STIMOLO È CONDIZIONE NECESSARIA MA NON SUFFICIENTE AD EVOCARE RISPOSTE MOTORIE
- CATEGORIE DI RISPOSTE:
  - NULLA
  - DISCRETA
  - CONTINUA



**ROBOTICS LAB**

## CODIFICA DISCRETA

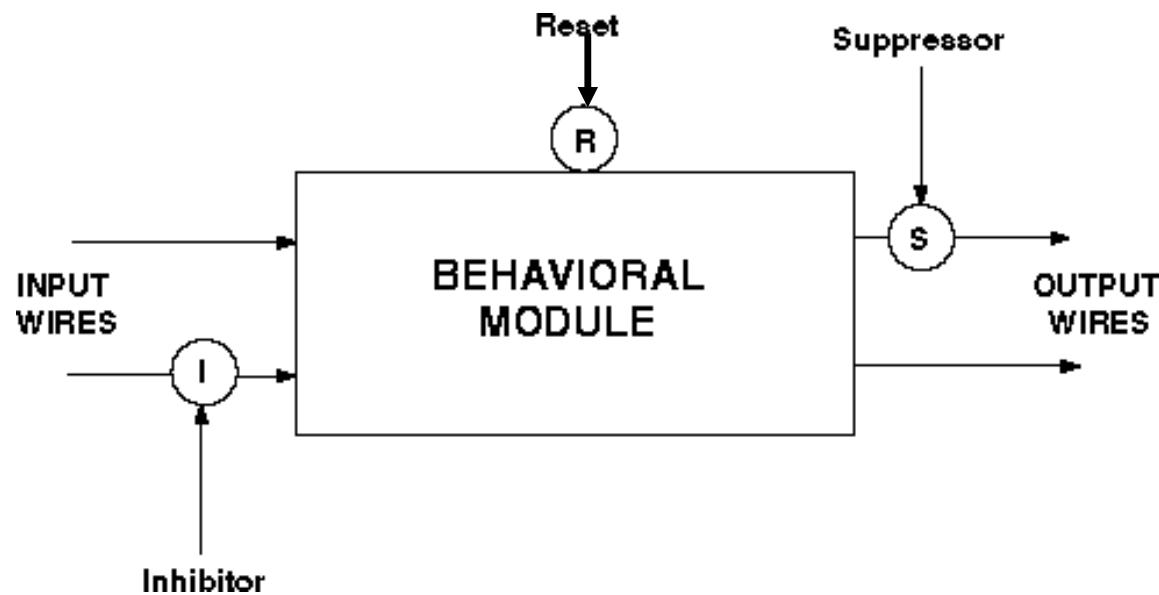
---

- INSIEME FINITO DI COPPIE SITUAZIONE/RISPOSTA
- TIPICAMENTE BASATA SU REGOLE
  - GAPS
  - NILSSON - REGOLE DI PRODUZIONE
  - SUBSUMPTION BEHAVIOR LANGUAGE

PROF. ROSARIO SORBELLO



## AFSM ORIGINALE



PROF. ROSARIO SORBELLO

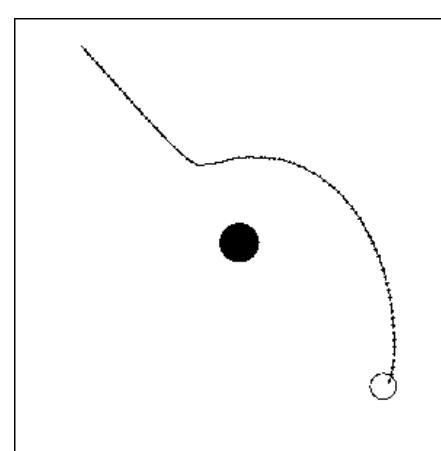
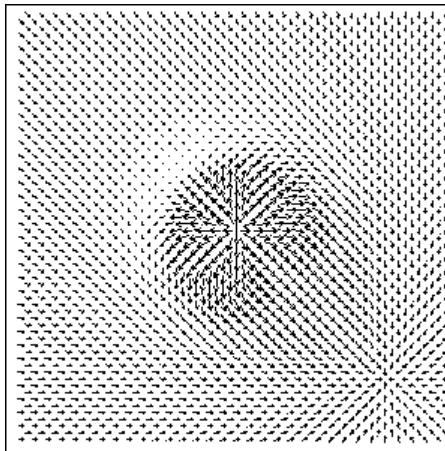
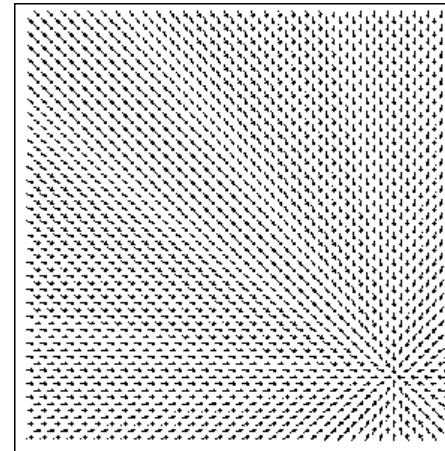
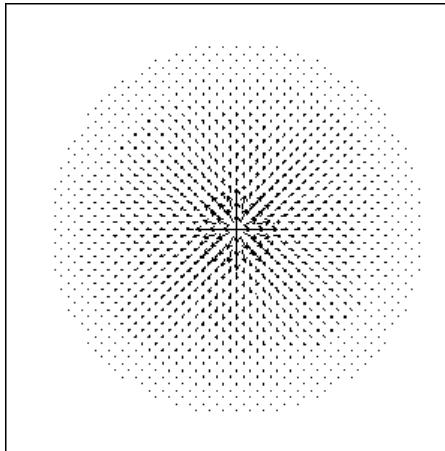


- Analogia con i campi di potenziale
- Semplici, tempovarianti, armonici, ecc.
- Problemi con minimi e massimi locali, possibili comportamenti oscillatori
- Gli schemi sono generalmente codificati in questo modo



**ROBOTICS LAB**

## ESEMPI DI CODIFICA DI SCHEMI



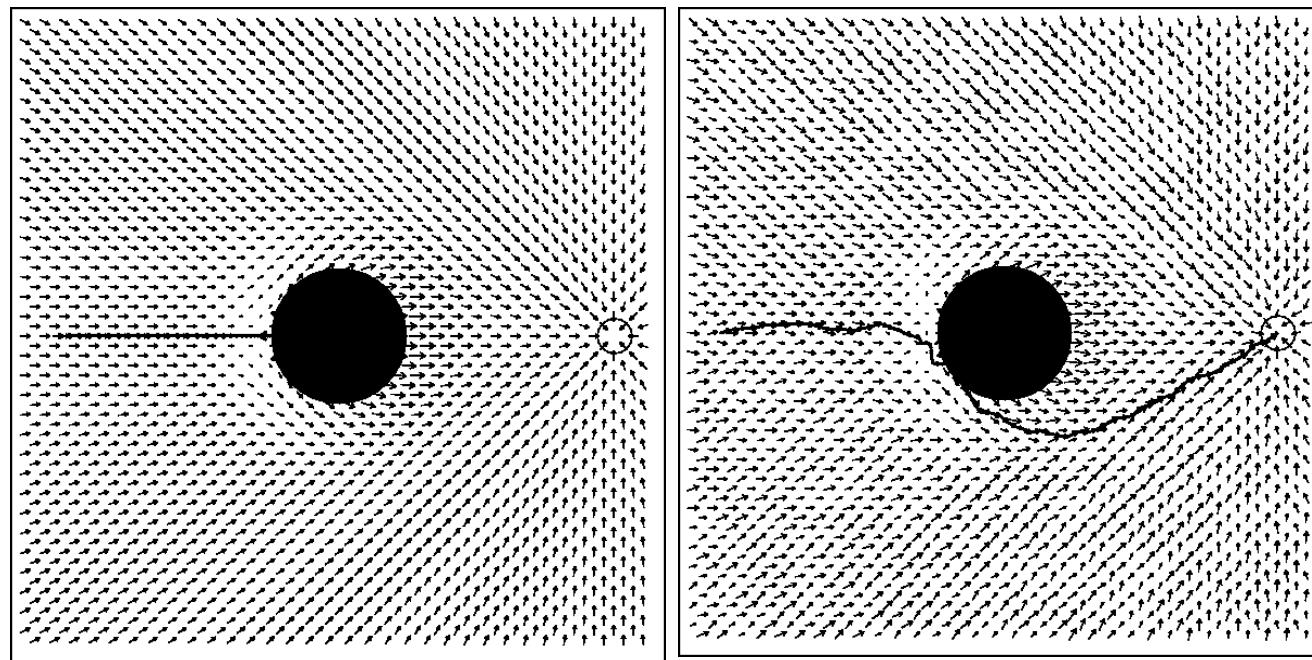
PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## RUMORE

---



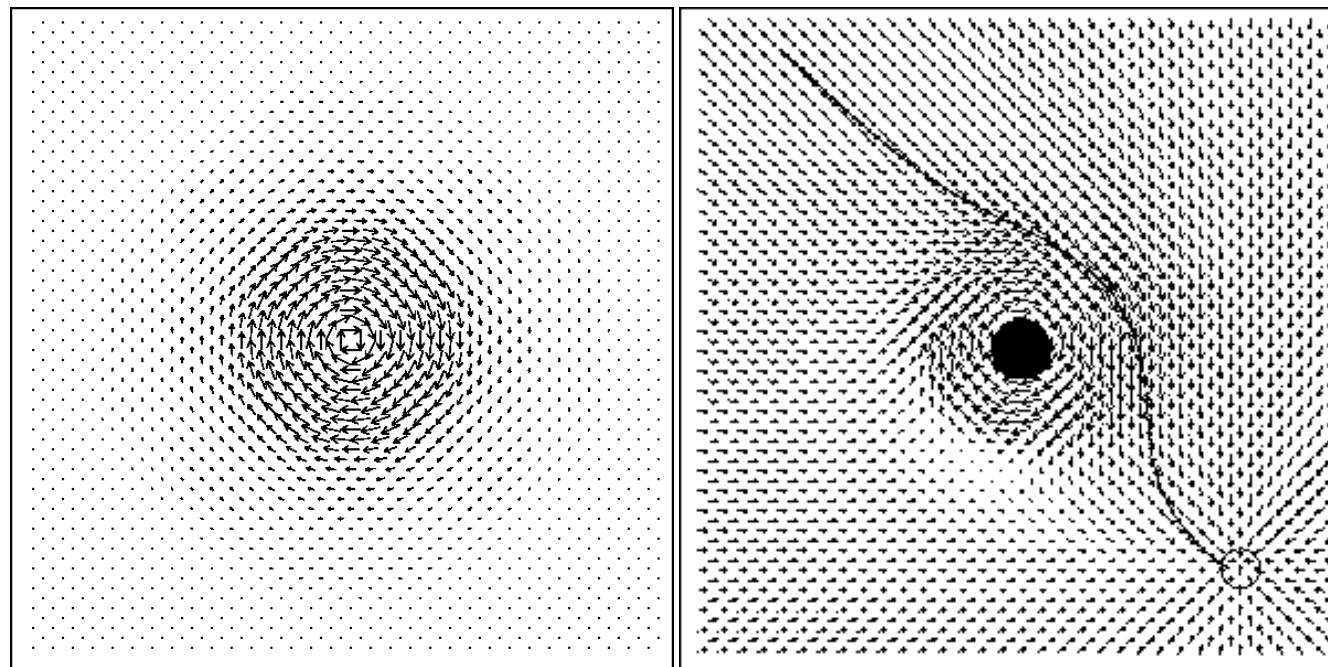
PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## SCHEMA BASATO SU ROTAZIONE (SWIRL)

---



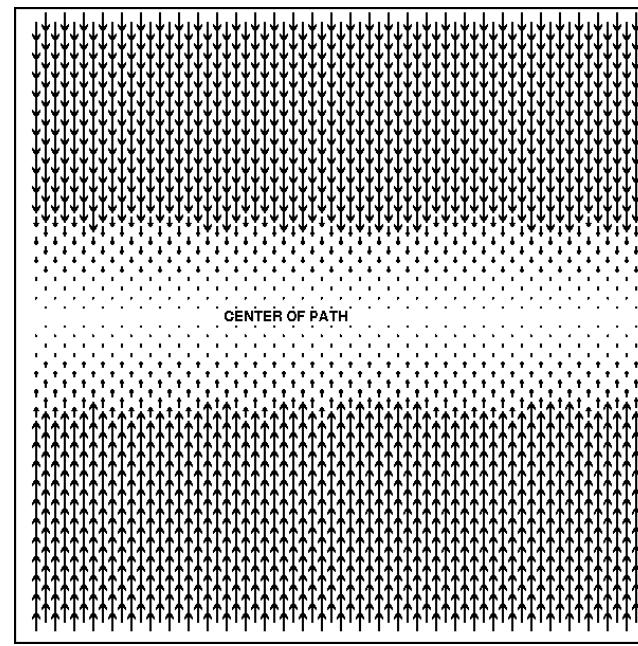
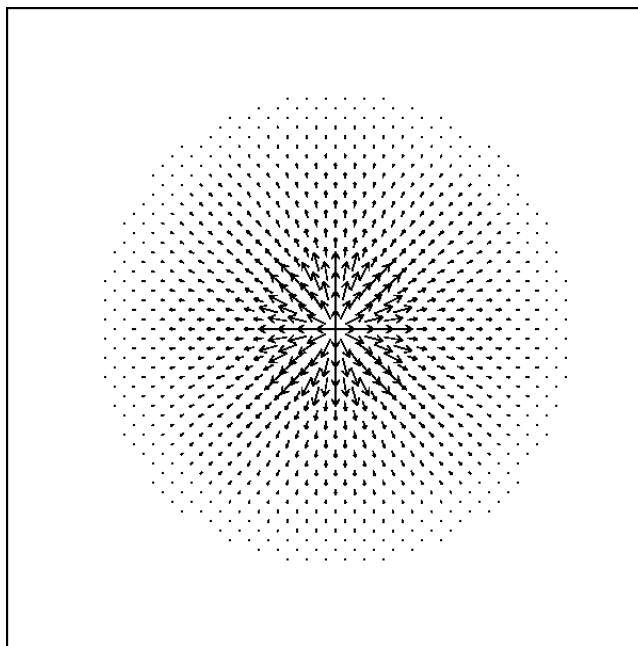
PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## SEGUIRE IL CAMMINO

---



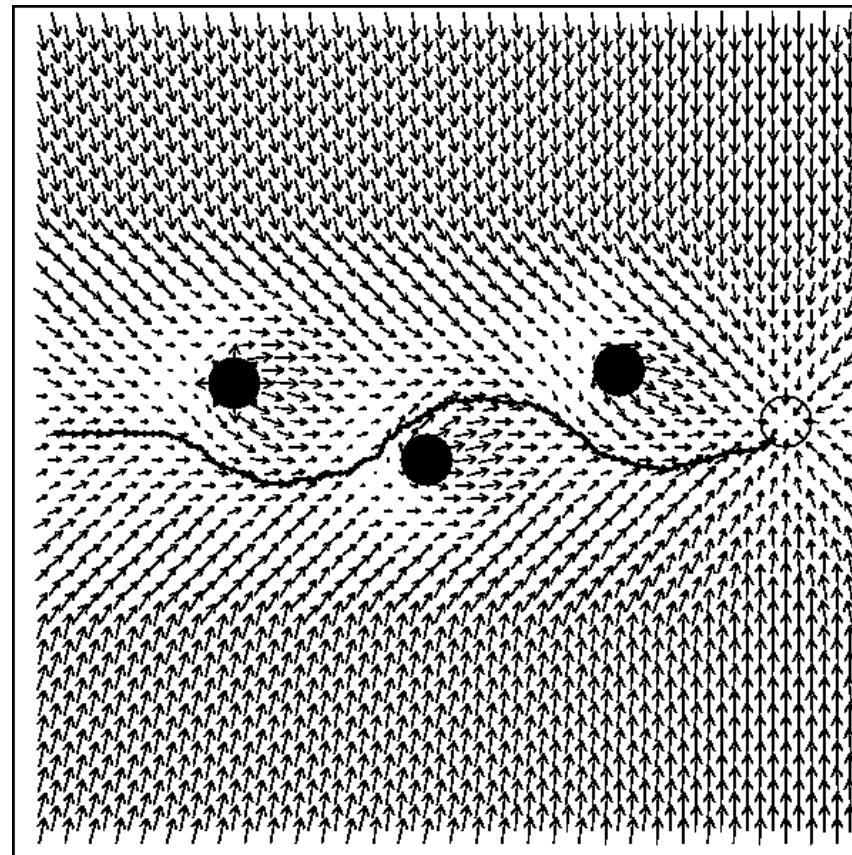
PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## ESEMPI DI TRAIETTORIE

---

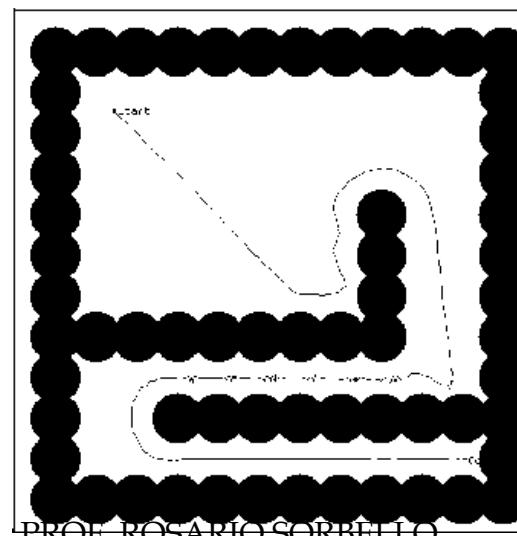
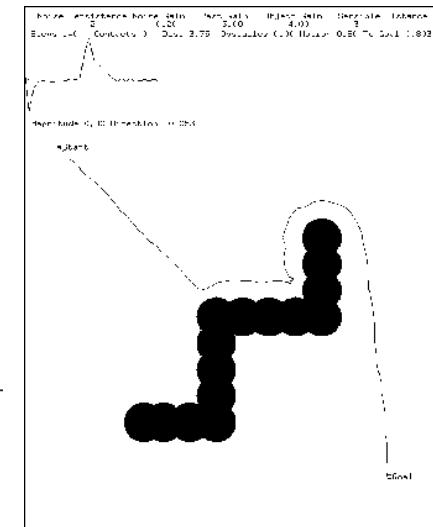
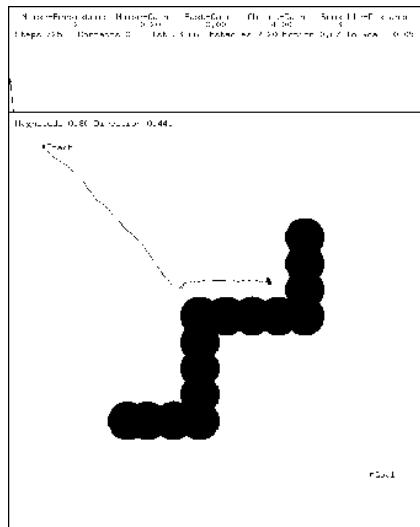


PROF. ROSARIO SORBELLO



**ROBOTICSLAB**

## COMPORTAMENTO EVITA PASSATO



PROF. ROSARIO SORBELO



**ROBOTICS LAB**

## MECCANISMI DI COORDINAMENTO

---

- Competitivi
- Cooperativi
- Combinazioni

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## NOTAZIONE

---

- Funzione di Coordinamento C
- Vettore di Guadagni G

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## COMPORTAMENTO EMERGENTE

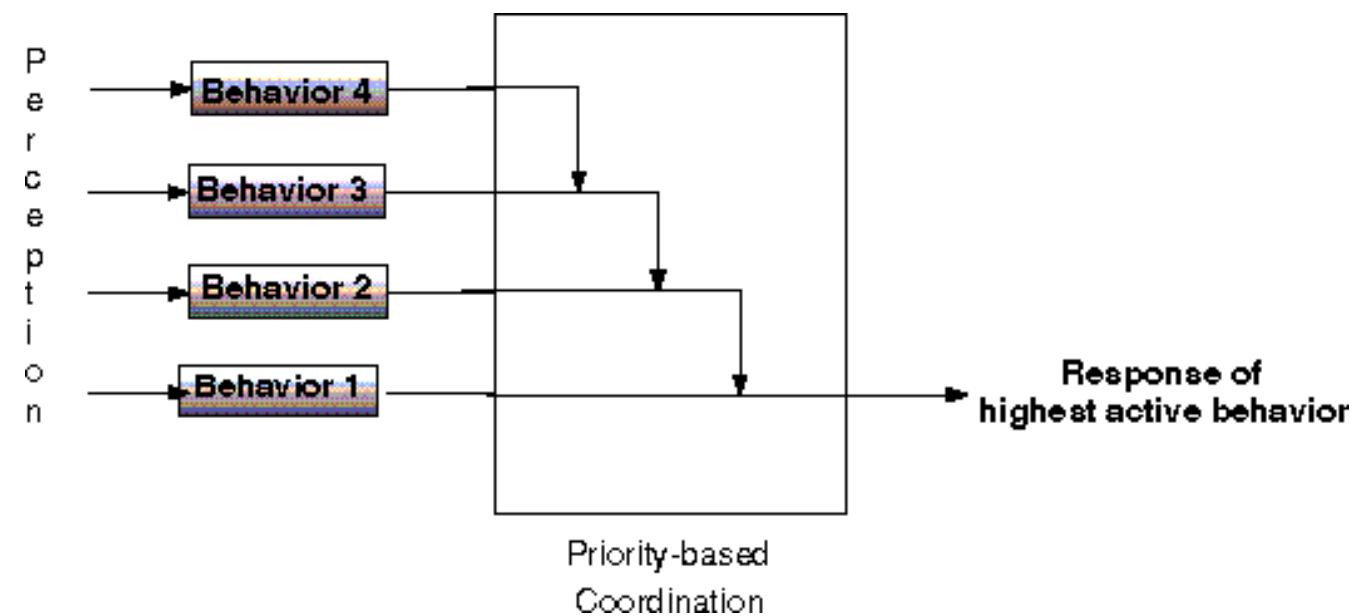
---

- Non compare magicamente
- Conseguenza della complessità del mondo in cui l'agente risiede e delle interazioni risultanti

PROF. ROSARIO SORBELLO



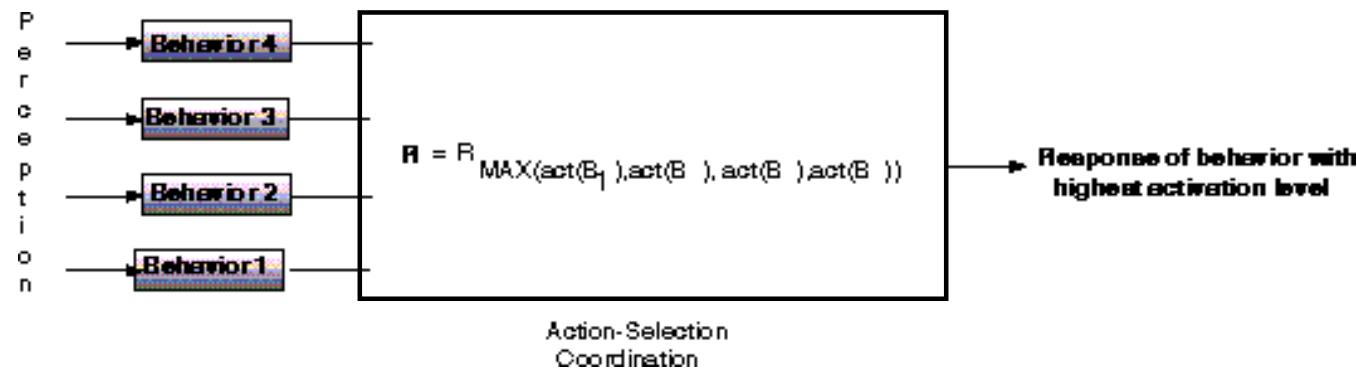
## ARBITRAGGIO CON SOPPRESSIONE



PROF. ROSARIO SORBELLO



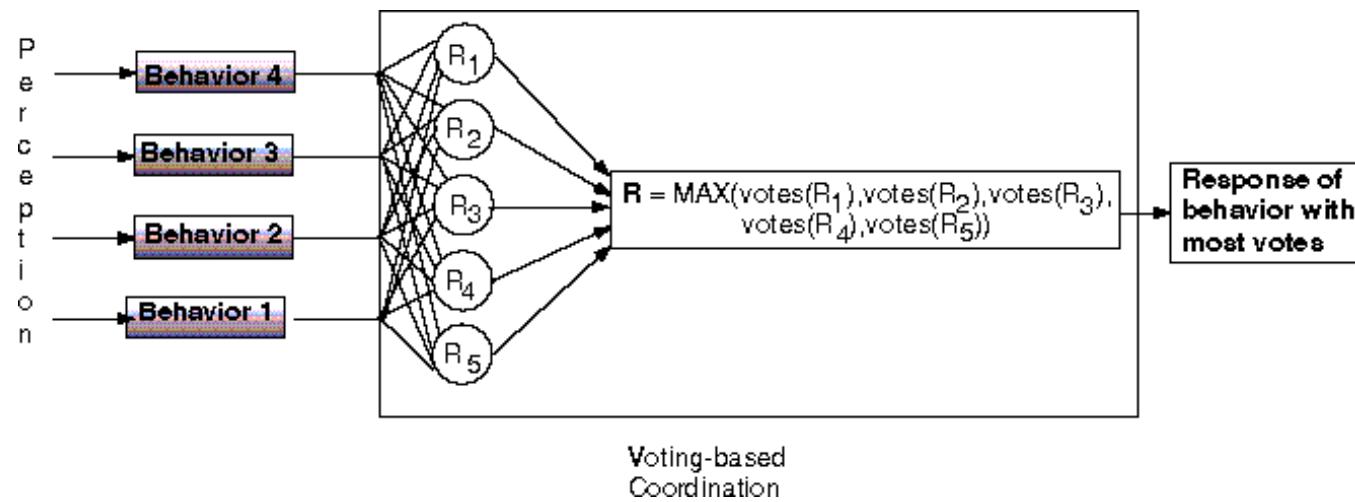
## ARBITRAGGIO CON SELEZIONE DI AZIONI



PROF. ROSARIO SORBELLO



## ARBITRAGGIO CON VOTAZIONE



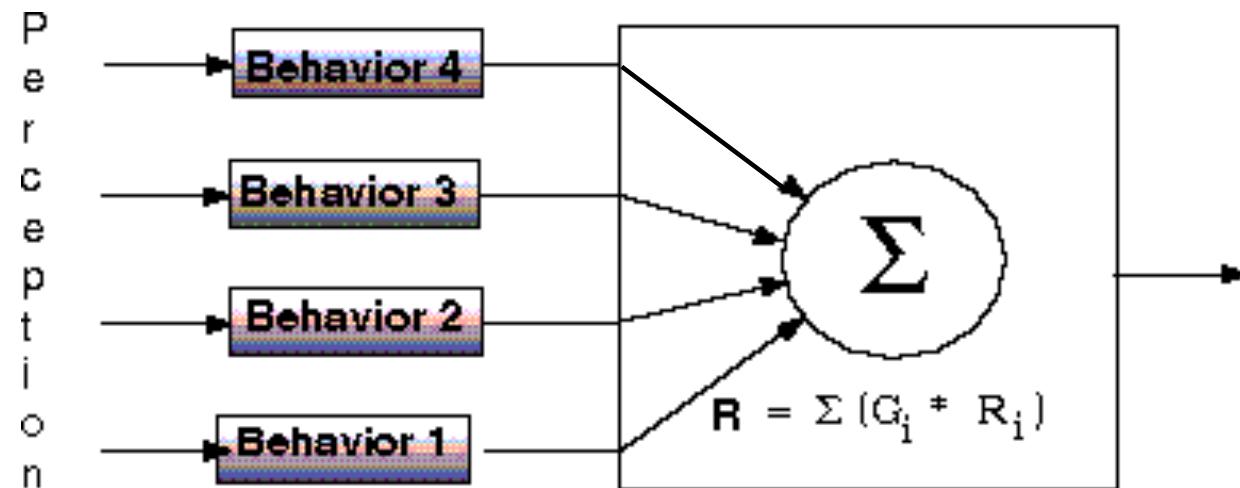
PROF. ROSARIO SORBELLO



ROBOTICS LAB

## FUSIONE DI COMPORTAMENTI CON SOMMA VETTORIALE

---



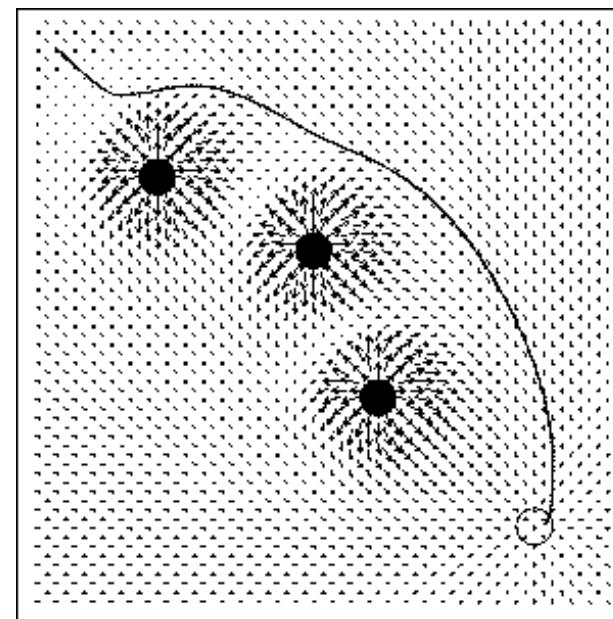
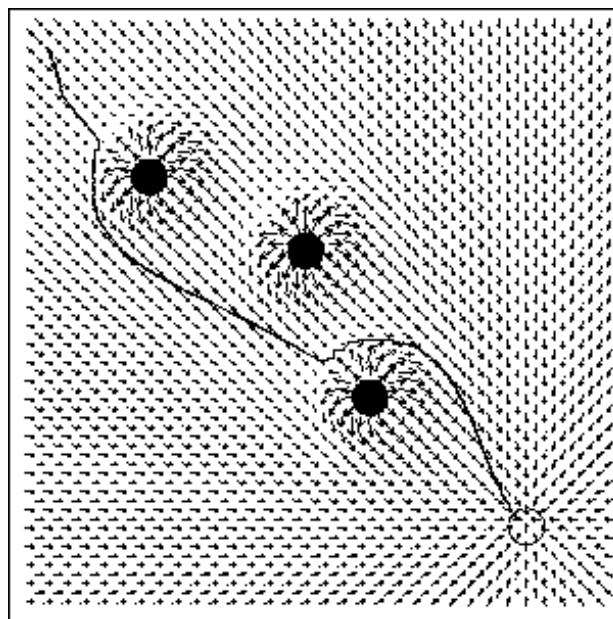
PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## FUSIONE DI COMPORTAMENTI - GUADAGNI

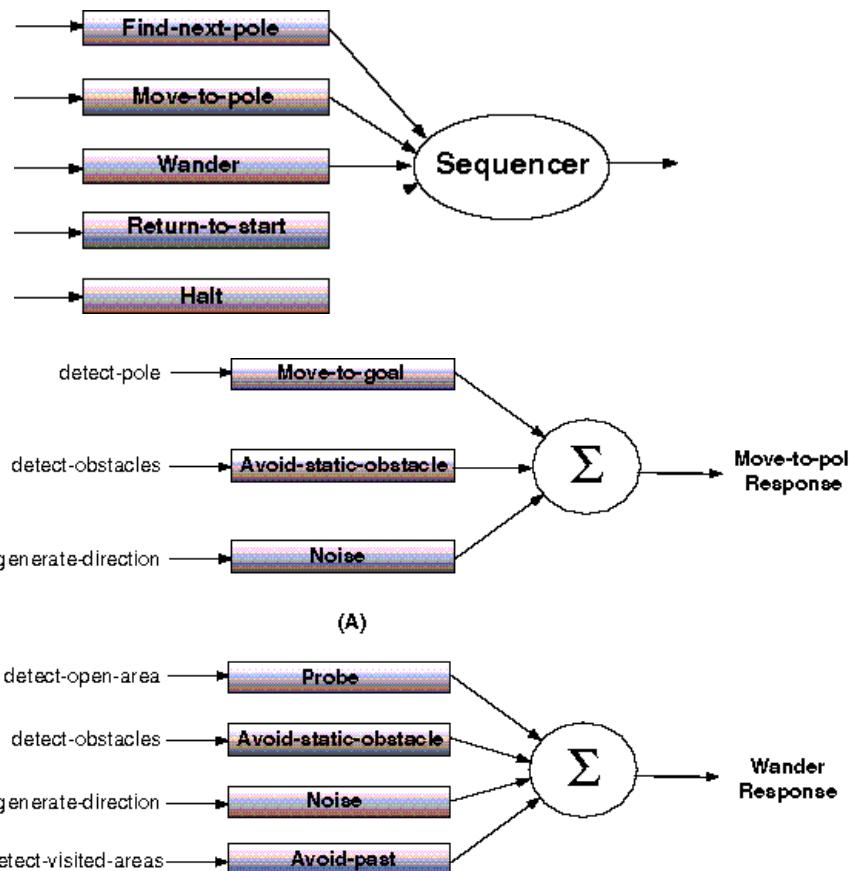
---



PROF. ROSARIO SORBELLO



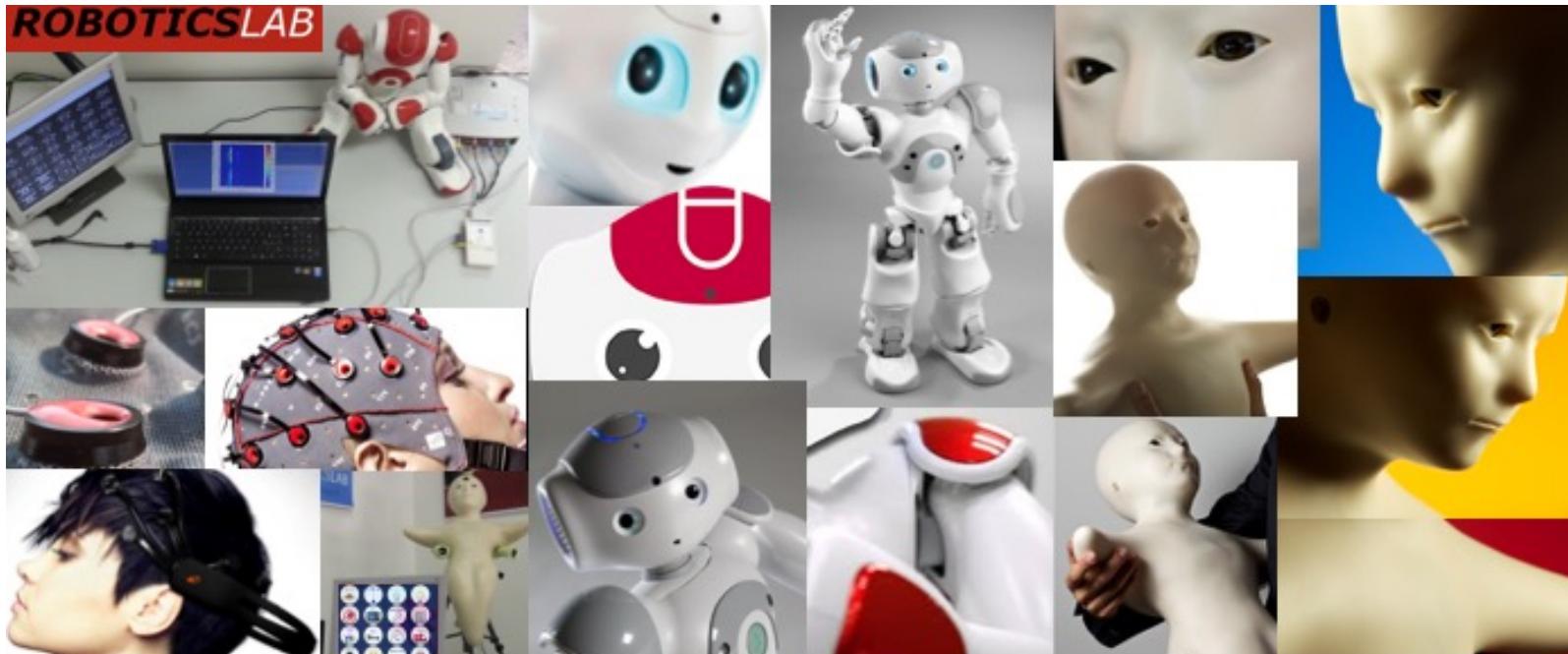
## DIAGRAMMI SR DI COORDINAMENTI



PROF. ROSARIO SORBELLO



OSAKA UNIVERSITY



ATR

g-tec

## CINEMATICA

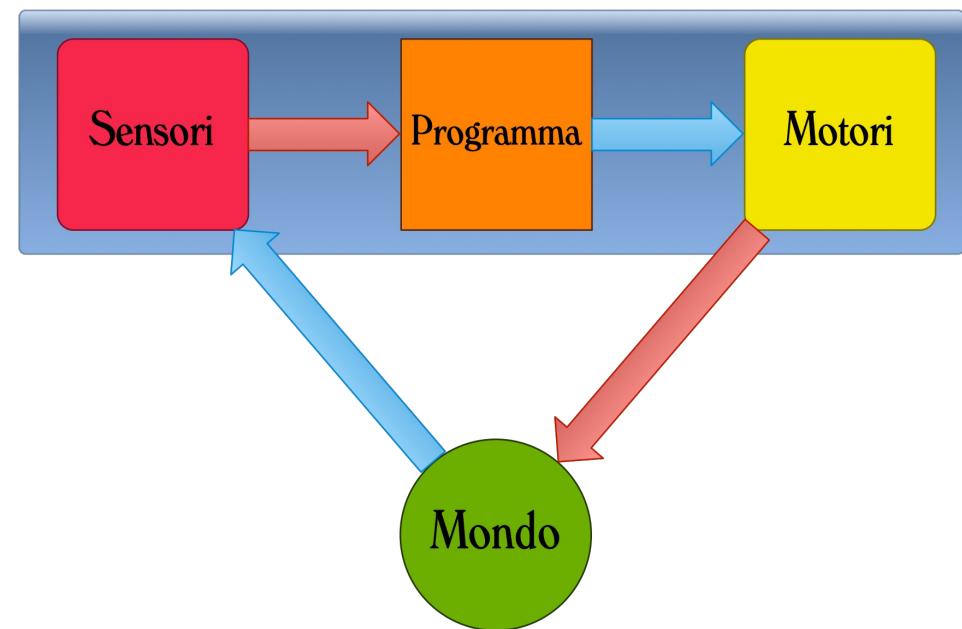
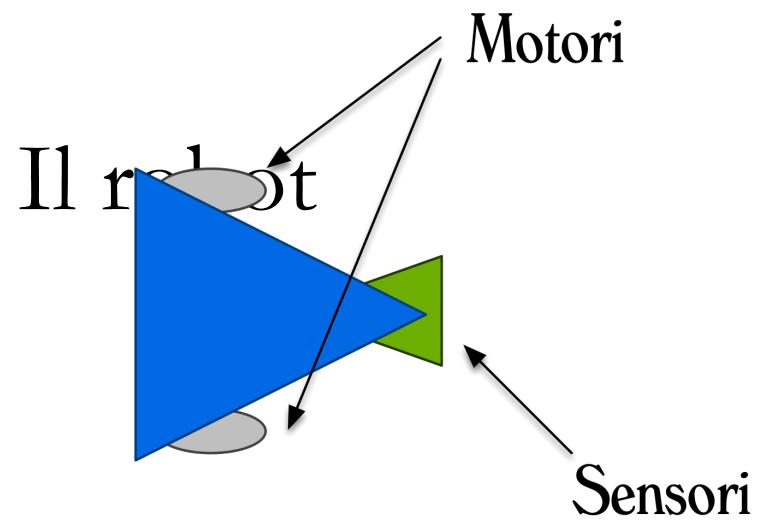
PROF. ROSARIO SORBELLO

3° LEZIONE - PALERMO 17 OTTOBRE 2022



**ROBOTICS LAB**

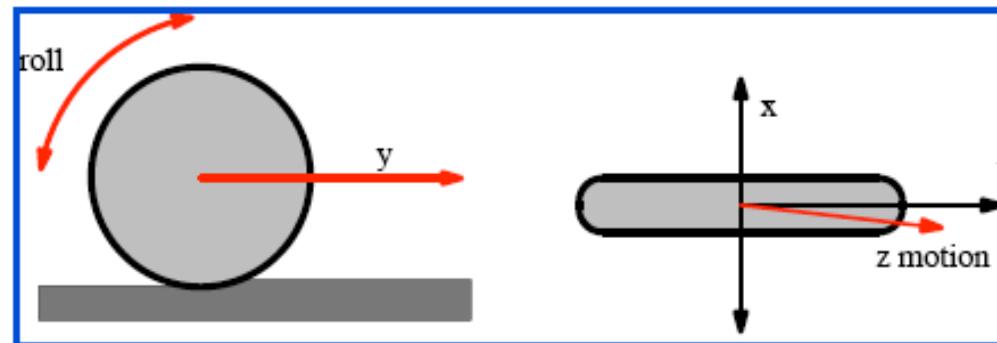
## IL ROBOT



PROF. ROSARIO SORBELLO

## LOCOMOZIONE DEI ROBOT

- Locomozione: azione di muoversi da un luogo all'altro
- Si considera il moto esclusivamente attorno l'asse di rotazione x

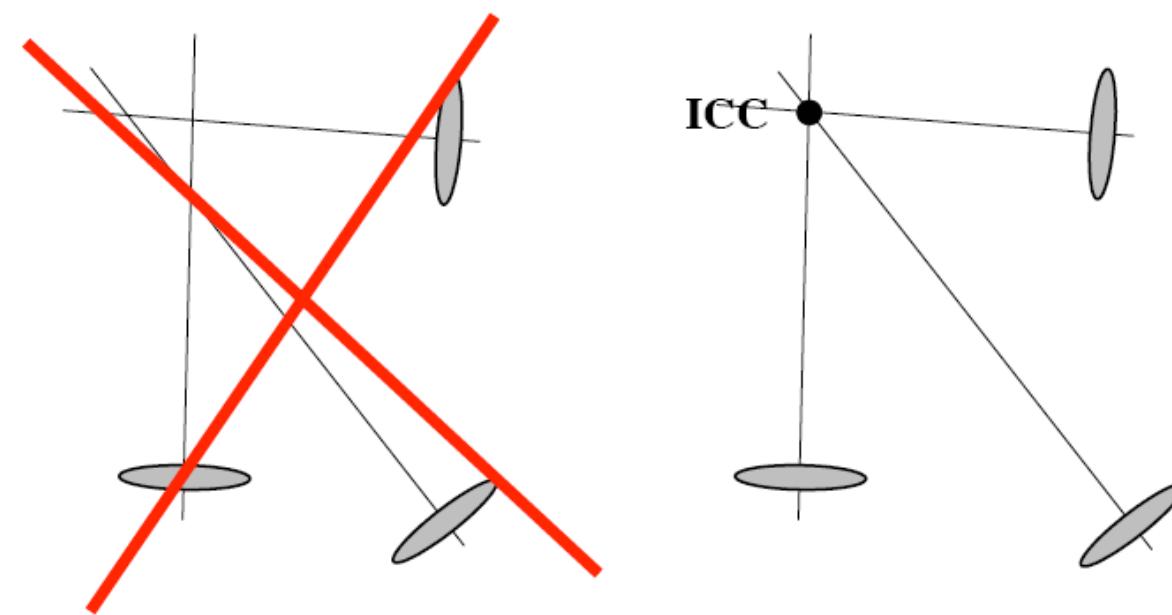


PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## CENTRO ISTANTANEO ROTAZIONALE (ICC)

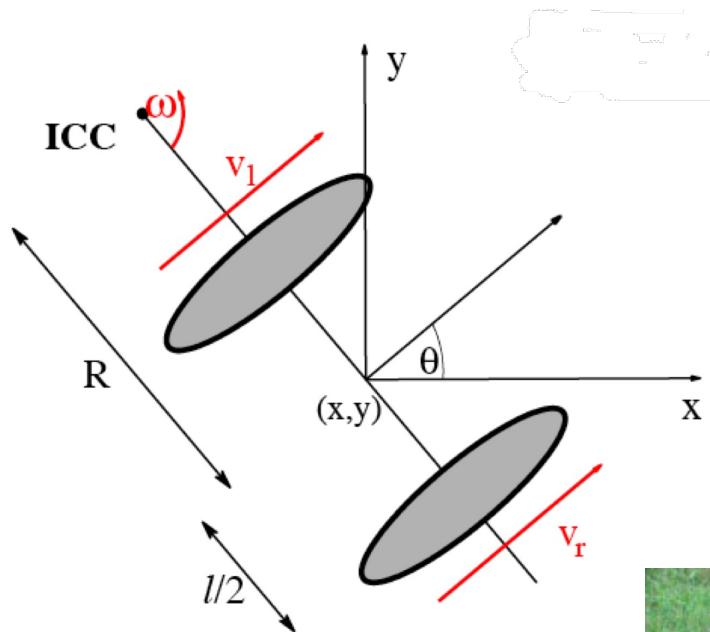


PROF. ROSARIO SORBELLO

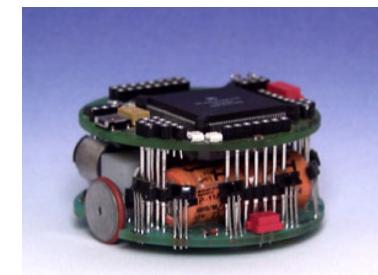


ROBOTICS LAB

## GUIDA DIFFERENZIALE



PROBOT  
Pioneer outdoor



Khepera



Pioneer 2



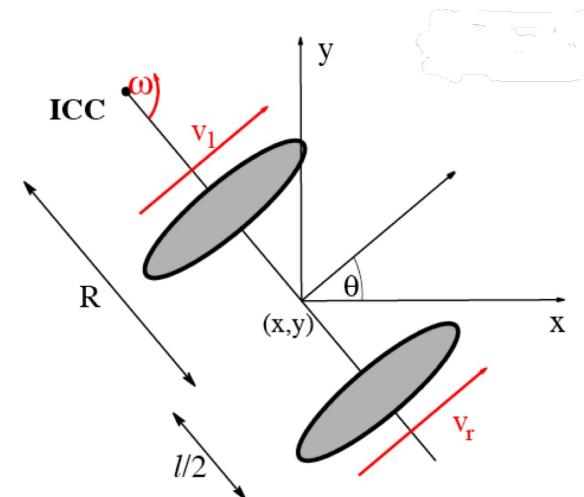
## GUIDA DIFFERENZIALE

$$\omega \left( R + \frac{l}{2} \right) = v_r$$

$$\omega \left( R - \frac{l}{2} \right) = v_l$$

$$R = \frac{l}{2} \frac{(v_l + v_r)}{(v_r - v_l)}$$

$$\omega = \frac{v_r - v_l}{l}$$



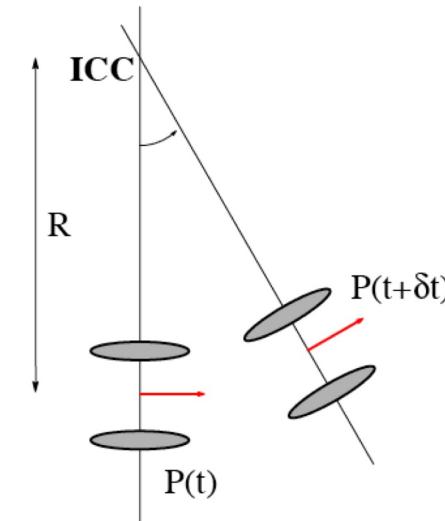
$$ICC = \begin{bmatrix} x - R \sin(\theta) \\ y + R \cos(\theta) \end{bmatrix}$$

PROF. ROSARIO SORBELLO



Posizione del robot a  $t + \delta t$  dopo  
una rotazione  $\omega\delta t$

Matematicamente si effettua una  
traslazione di ICC sull'origine,  
una rotazione di e una ri-  
traslazione su ICC.



$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix}$$

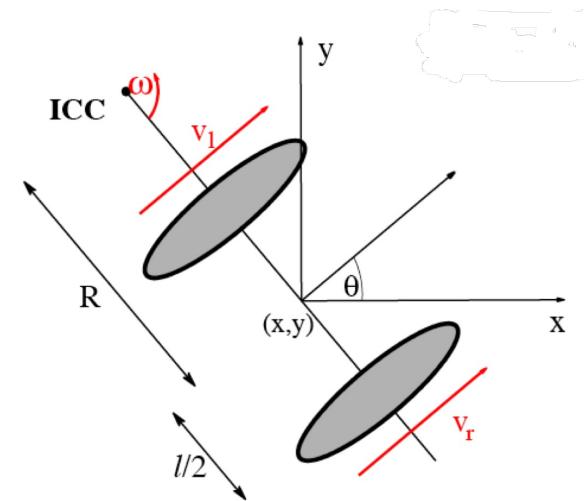


## GUIDA DIFFERENZIALE

$$x(t) = \int_0^t V(t) \cos [\theta(t)] dt$$

$$y(t) = \int_0^t V(t) \sin [\theta(t)] dt$$

$$\theta(t) = \int_0^t \omega(t) dt$$



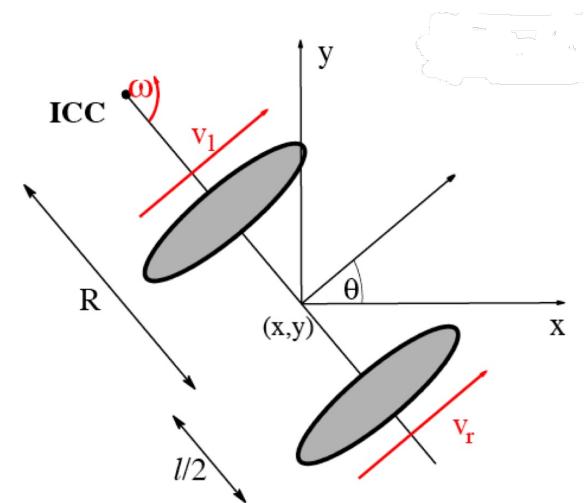
Equazioni generali della cinematica diretta per un veicolo in direzione  $\theta(t)$  e velocità  $V(t)$

## GUIDA DIFFERENZIALE

$$x(t) = \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \cos[\theta(t)] dt$$

$$y(t) = \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \sin[\theta(t)] dt$$

$$\theta(t) = \frac{1}{l} \int_0^t [v_r(t) - v_l(t)] dt$$



Equazioni della cinematica diretta per un veicolo a guida differenziale in direzione  $\theta(t)$  e velocità  $v_r(t)$ ,  $v_l(t)$

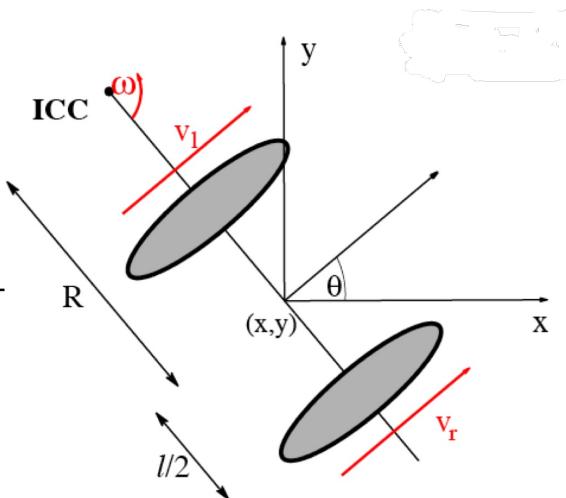
## GUIDA DIFFERENZIALE

$$x(t) = \frac{l}{2} \frac{v_r + v_l}{v_r - v_l} \sin \left[ \frac{t}{l} (v_r - v_l) \right]$$

$$y(t) = -\frac{l}{2} \frac{v_r + v_l}{v_r - v_l} \cos \left[ \frac{t}{l} (v_r - v_l) \right] + \frac{l}{2} \frac{v_r + v_l}{v_r - v_l}$$

$$\theta(t) = \frac{t}{l} (v_r - v_l)$$

$$(x, y, \theta)_{t=0} = (0, 0, 0)$$



Equazioni della cinematica inversa per un veicolo a guida differenziale in direzione  $\theta(t)$  e velocità  $v_r, v_l$ . Risolvendo si ottengono infinite velocità al variare di  $\theta$ .



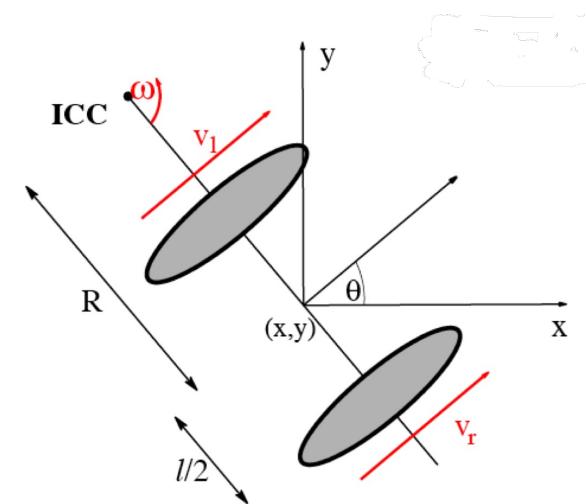
## GUIDA DIFFERENZIALE

$$v_l = v_r = v$$

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + v \cos(\theta) \delta t \\ y + v \sin(\theta) \delta t \\ \theta \end{pmatrix}$$

$$-v_l = v_r = v$$

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta + 2v \frac{\delta t}{l} \end{pmatrix}$$



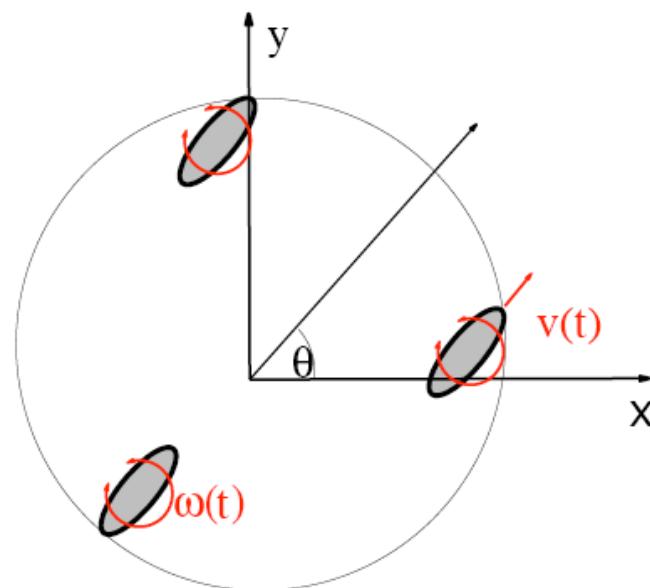
Piuttosto che invertire le equazioni della cinematica, si fa ruotare il robot su se stesso, lo si fa muovere lungo una retta fino ad arrivare a  $(x,y)$  finale e lo si fa ruotare ancora per assumere la configurazione finale  $(x,y,\theta)$ .

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## GUIDA SINCRONA



RWI B21

PROF. ROSARIO SORBELLO

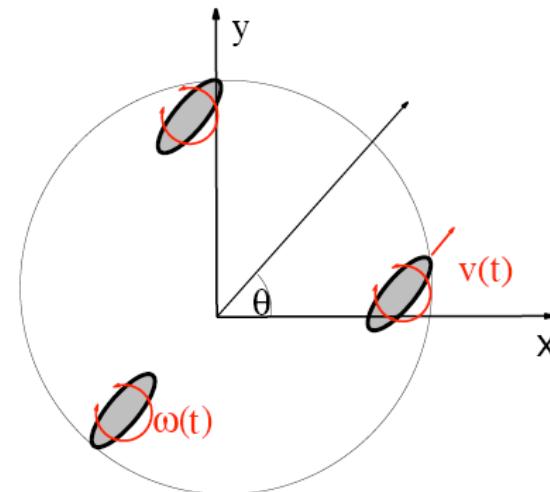


$$x(t) = \int_0^t V(t) \cos [\theta(t)] dt$$

$$y(t) = \int_0^t V(t) \sin [\theta(t)] dt$$

$$\theta(t) = \int_0^t \omega(t) dt$$

$$ICC \rightarrow \infty$$



Equazioni generali della cinematica diretta per un veicolo in direzione  $\theta(t)$  e velocità  $V(t)$ .



$$v(t) = 0$$

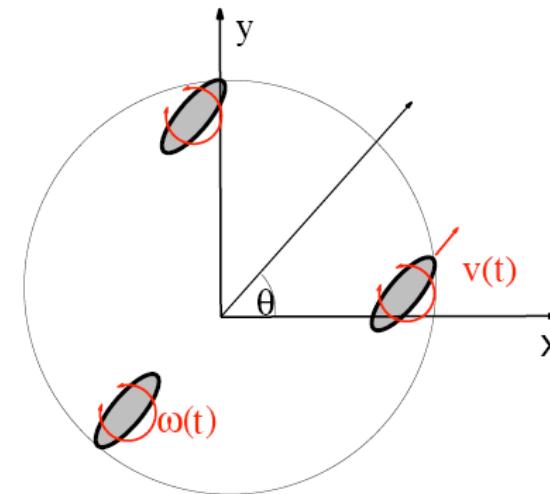
$$\omega(t) = \omega$$

Rotazione di  $\omega\delta t$

$$v(t) = v$$

$$\omega(t) = 0$$

Traslazione di  $v\delta t$



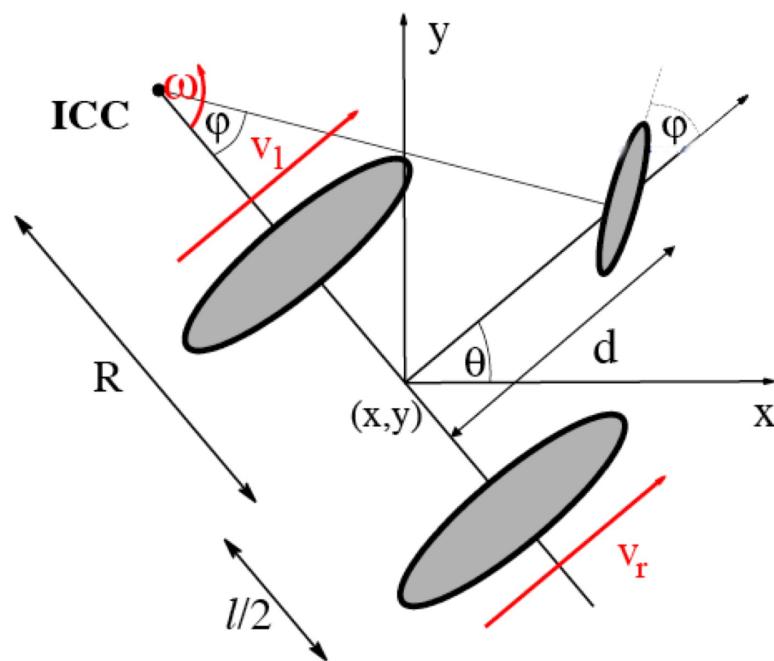
Anche in questo caso si fa ruotare il robot su se stesso, lo si fa muovere lungo una retta fino ad arrivare a  $(x,y)$  finale e lo si fa ruotare ancora per assumere la configurazione finale  $(x,y,\theta)$ .

PROF. ROSARIO SORBELLO



ROBOTICS LAB

## TRICICLO



Staffetta  
(GenovaRobot)

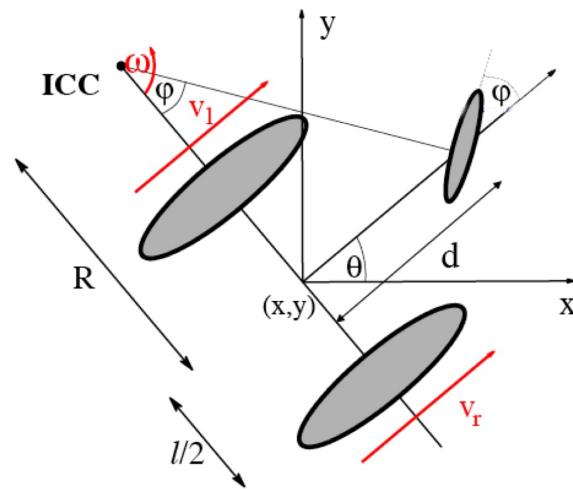
PROF. ROSARIO SORBELLO



## TRICICLO

$$R = \frac{d}{\tan(\varphi)}$$

$$\omega = \frac{v}{\sqrt{d^2 + R^2}}$$



$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix}$$

PROF. ROSARIO SORBELLO

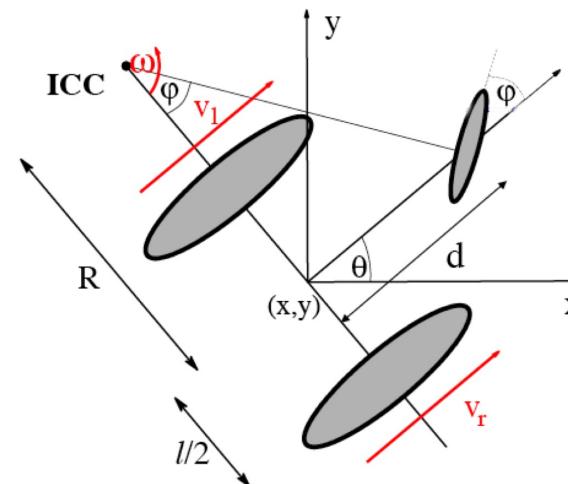
## TRICICLO

$$\varphi = 0$$

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + v \cos(\theta) \delta t \\ y + v \sin(\theta) \delta t \\ \theta \end{pmatrix}$$

$$\varphi = \pm \frac{\pi}{2}$$

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \pm v \frac{\delta t}{l} \end{pmatrix}$$



La cinematica inversa si risolve in maniera simile al caso della guida differenziale o sincrona

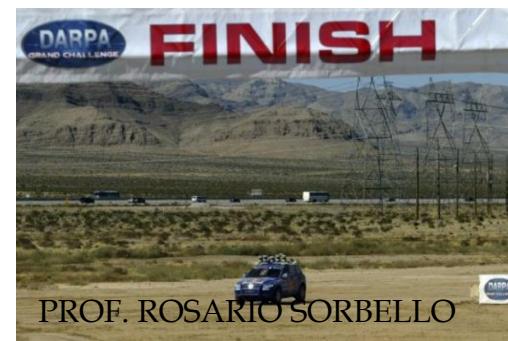


**ROBOTICS LAB**

## AUTOMOBILE ACKERMAN

---

- 2 ruote sterzanti
- Cinematica diretta e inversa molto complessa
- Automobile autonoma





**ROBOTICS LAB**

## RUOTE COMPLESSE



**Mecanum wheels**

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## RUOTE COMPLESSE



**Robot cingolati**

PROF. ROSARIO GONDELLO



**ROBOTICS LAB**

---

# Robot non tradizionali

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

- UAV : Unmanned Aerial Vehicle
- ASV : Autonomous Surface Vehicle
- ROV : Remotely Operated Vehicle



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Robovolc UNICT

Monitoraggio dell'Etna



PROF. ROSARIO SORBELLO

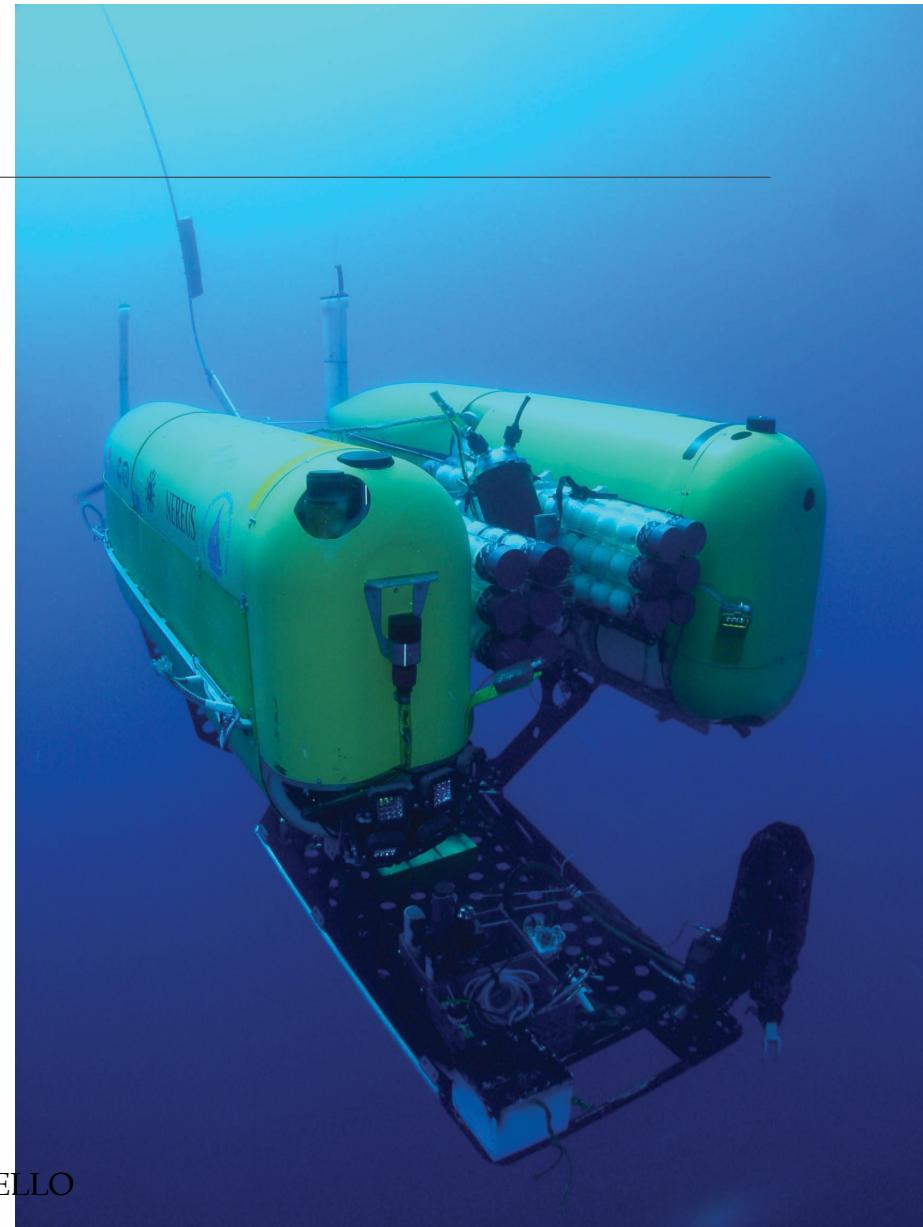


**ROBOTICS LAB**

# Nereus ROV

Nereus ROV ha esplorato l'oceano  
profondo  
a 10,903 m

PROF. ROSARIO SORBELLO





**ROBOTICS LAB**

---

# Aereosonda UAV

Esplorazione autonoma in Antartide nel  
2009



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

---

# Wave Glider ASV

Oceano Pacifico vicino le Hawaii, 2010



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

---

# UAV per monitoraggio di alghe



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Interazione tra robot e rete di sensori in acqua

Lake Wivenhoe, Australia. Monitoraggio  
della qualità dell'acqua del lago



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

---

# Robot Pesce - Robot Fish

PROF. ROSARIO SORBELLO



ROBOTICS LAB

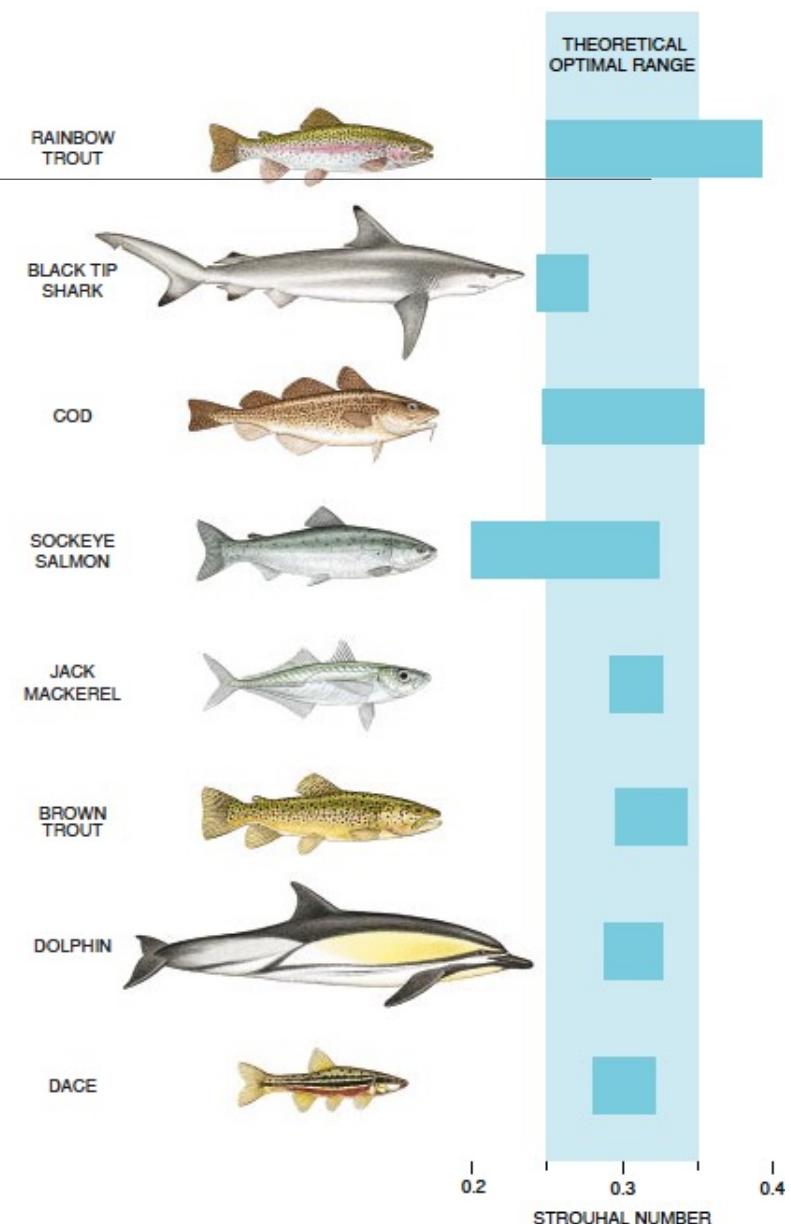


# Nuoto dei pesci

I pesci nuotano muovendo la coda per creare vortici che producono una spinta propulsiva di grande efficienza. Il numero di Strouhal indica l'efficienza del nuoto.



PROF. ROSARIO SORBELLO



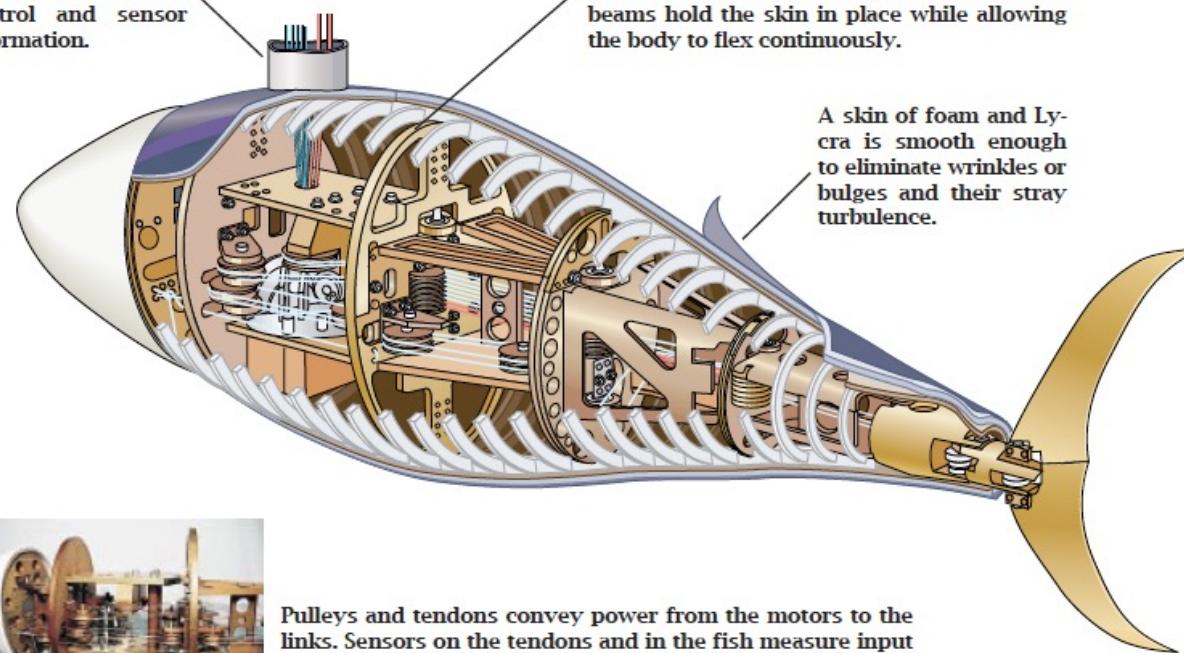


**ROBOTICS LAB**

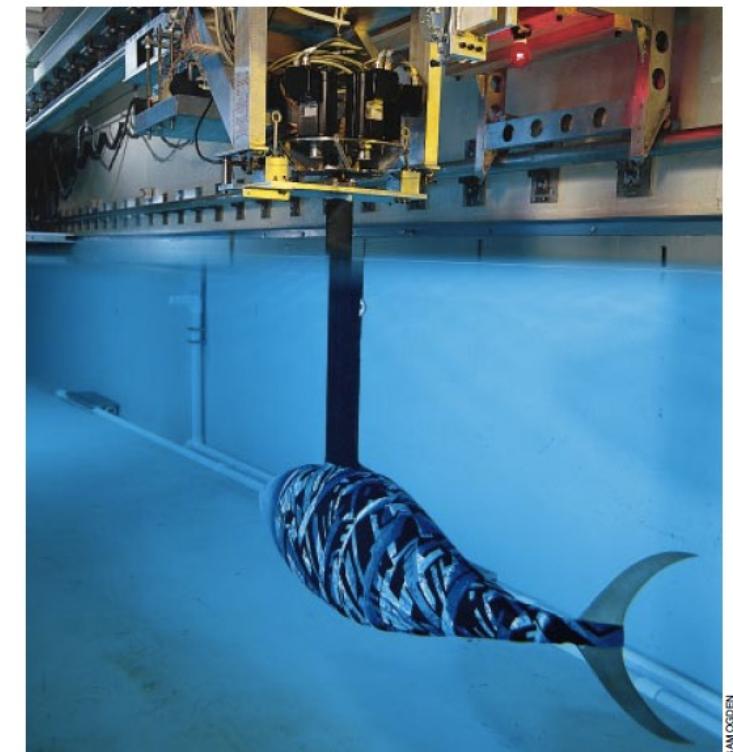
A strut supports the robot, encloses the tendons and conveys control and sensor information.

Links are connected by aluminum hinges, to which are affixed beams supporting ribs spaced one inch apart. The ribs and flexible beams hold the skin in place while allowing the body to flex continuously.

A skin of foam and Lycra is smooth enough to eliminate wrinkles or bulges and their stray turbulence.



Pulleys and tendons convey power from the motors to the links. Sensors on the tendons and in the fish measure input power, as well as external forces, pressure and velocity, and track vortices as they move along the robot's side.



SAM OGDEN

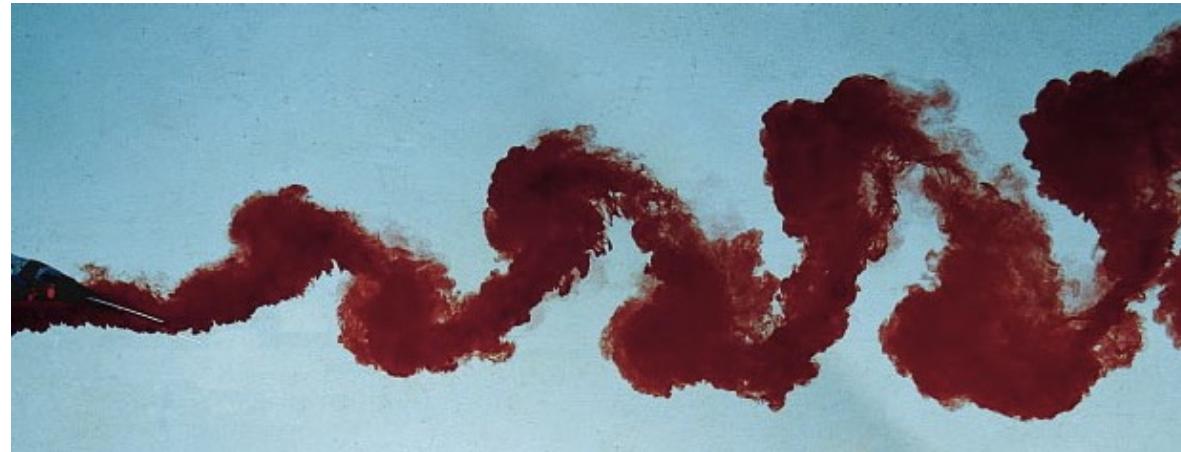
PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

---

# Nuoto del robot tonno



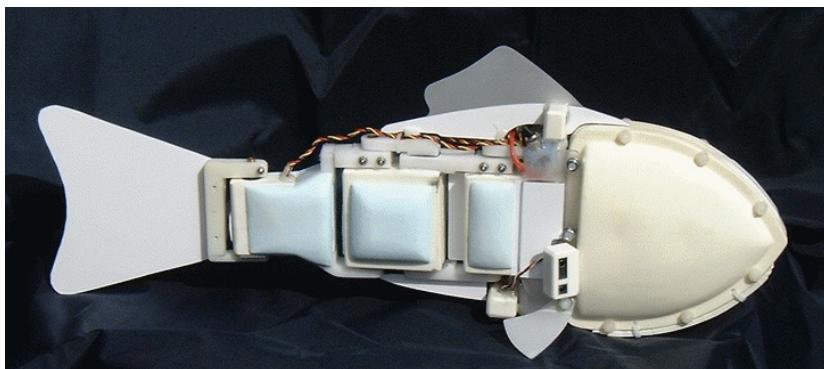
PROF. ROSARIO SORBELLO



ROBOTICS LAB

# Architettura di controllo di un robot pesce

University of Essex, UK



PROF. ROSARIO SORBELLO

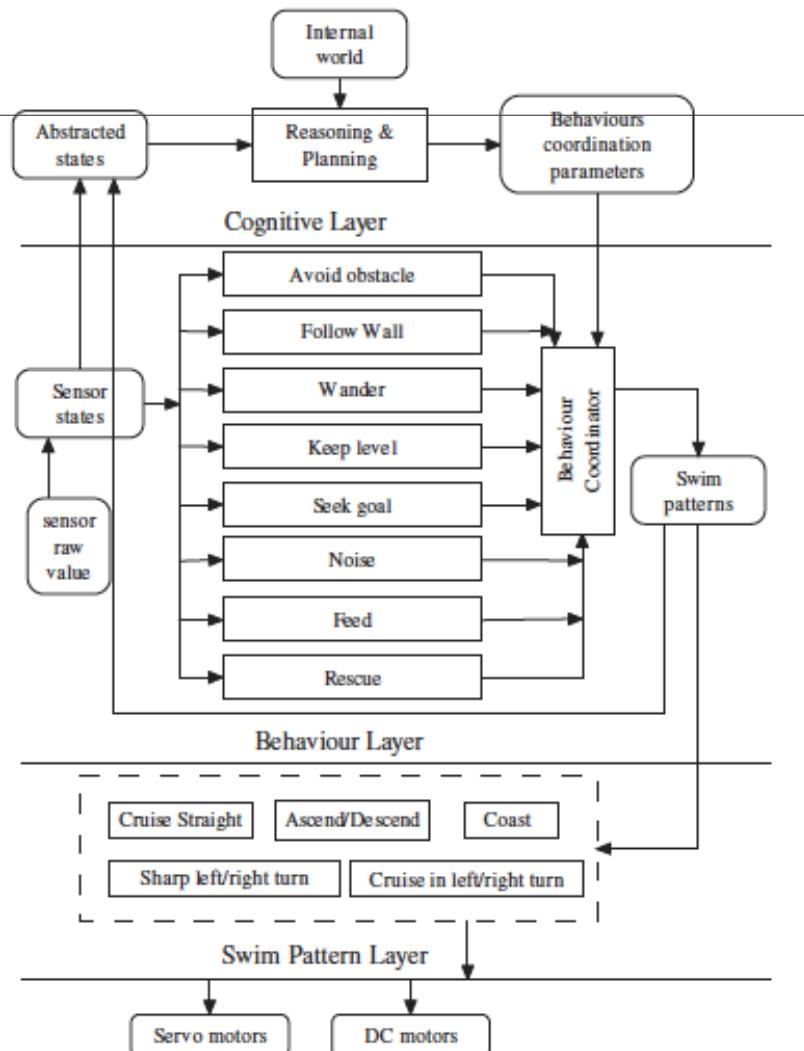


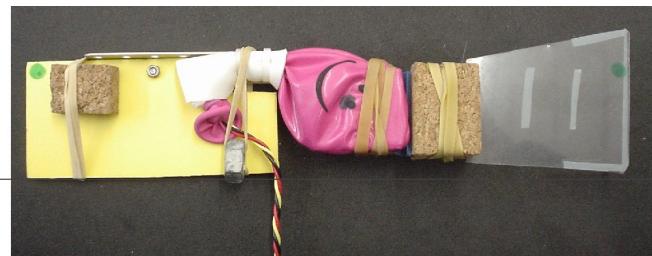
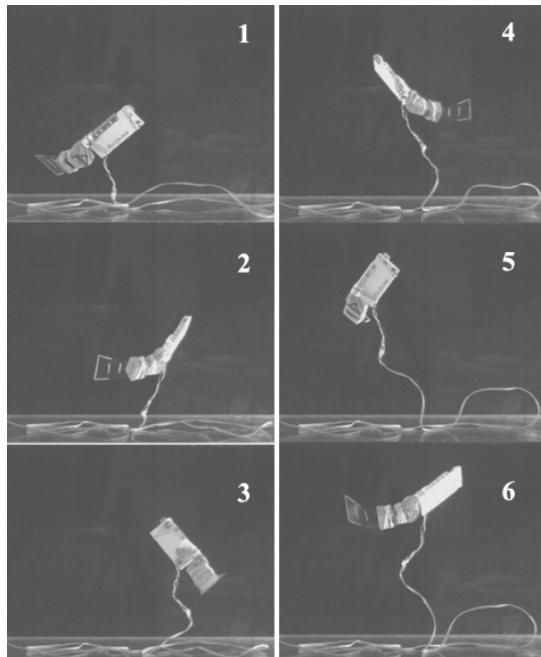
Fig. 1. The hybrid control architecture for our robotic fish



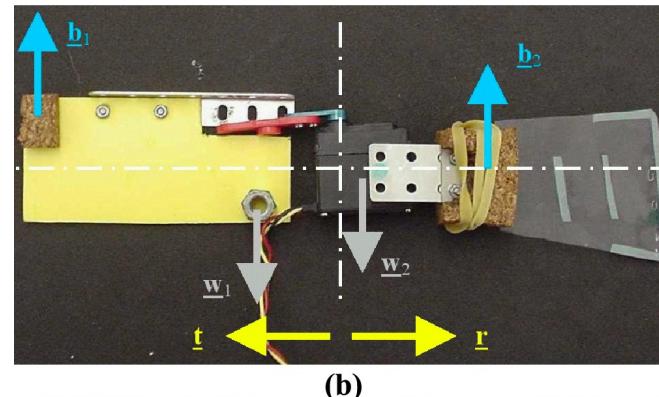
ROBOTICS LAB

# Robot pesce minimale

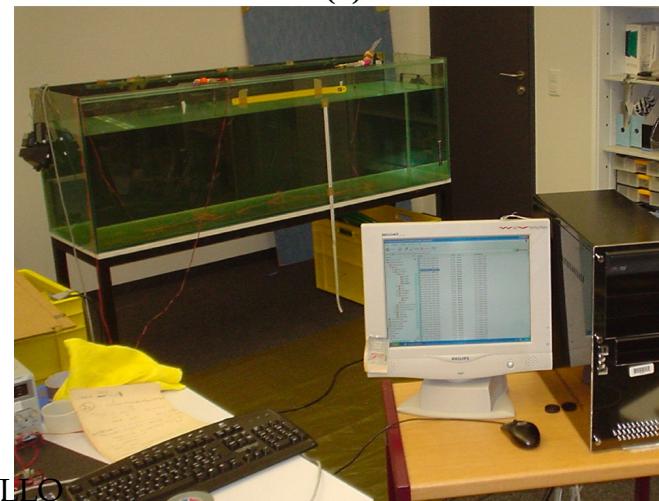
University of Zurich



(a)



(b)



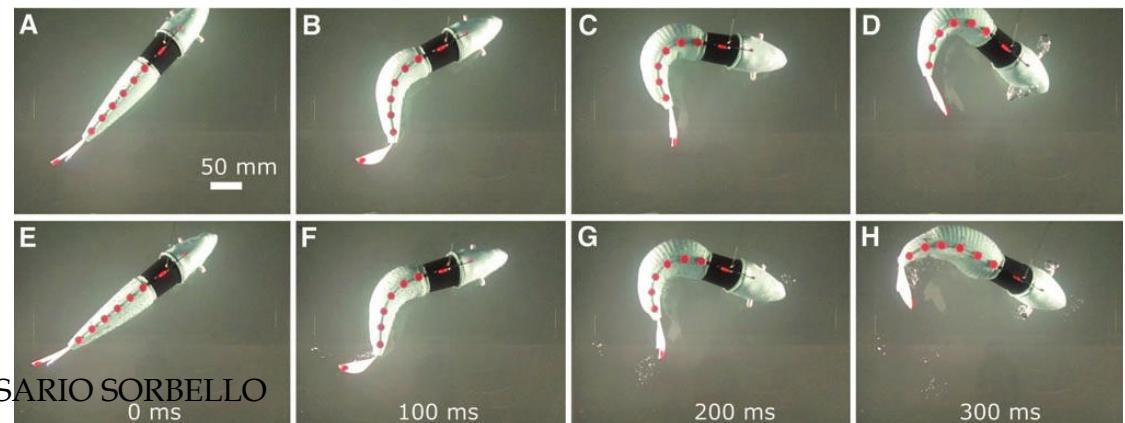
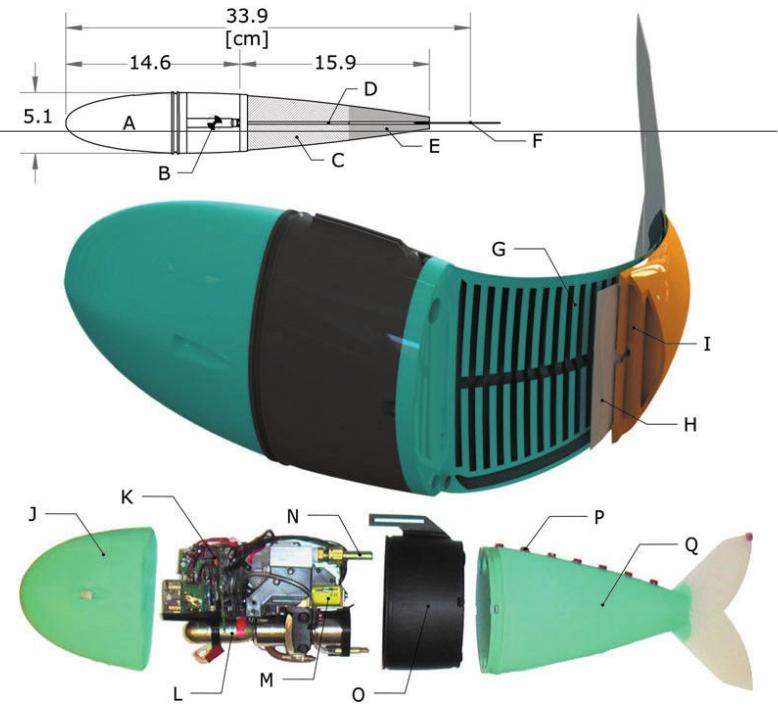
(c)

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Pesce “soft-robot” (MIT)



PROF. ROSARIO SORBELLO  
0 ms

100 ms

200 ms

300 ms



OSAKA UNIVERSITY



ATR

g-tec

## SONAR, LASER E MAPPA IN TEAMBOTS

PROF. ROSARIO SORBELLO

4° LEZIONE - PALERMO 19 OTTOBRE 2022



**ROBOTICS LAB**

## ROBOT MULTIFORAGEN150EXPLORE

---

- INTRODUZIONE DI SENSORISTICA + EFFICIENTE

VERSIONE BASE MULTIFORAGEN150:

PER OGNI OSTACOLO RILEVO SOLO IL PUNTO IN CUI QUESTO  
RISULTA MENO DISTANTE DAL ROBOT STESSO

ESEMPIO: 4 PARETI SOLO 4 PUNTI UNO PER PARETE

PROBLEMA: VICOLI CIECHI, CORRIDOI CHIUSI O OSTACOLI A V

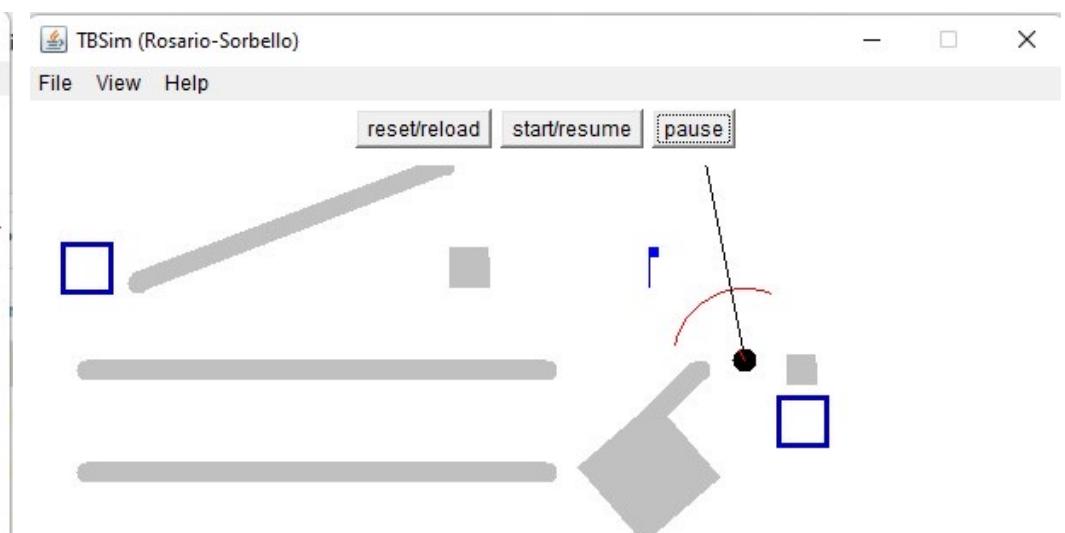
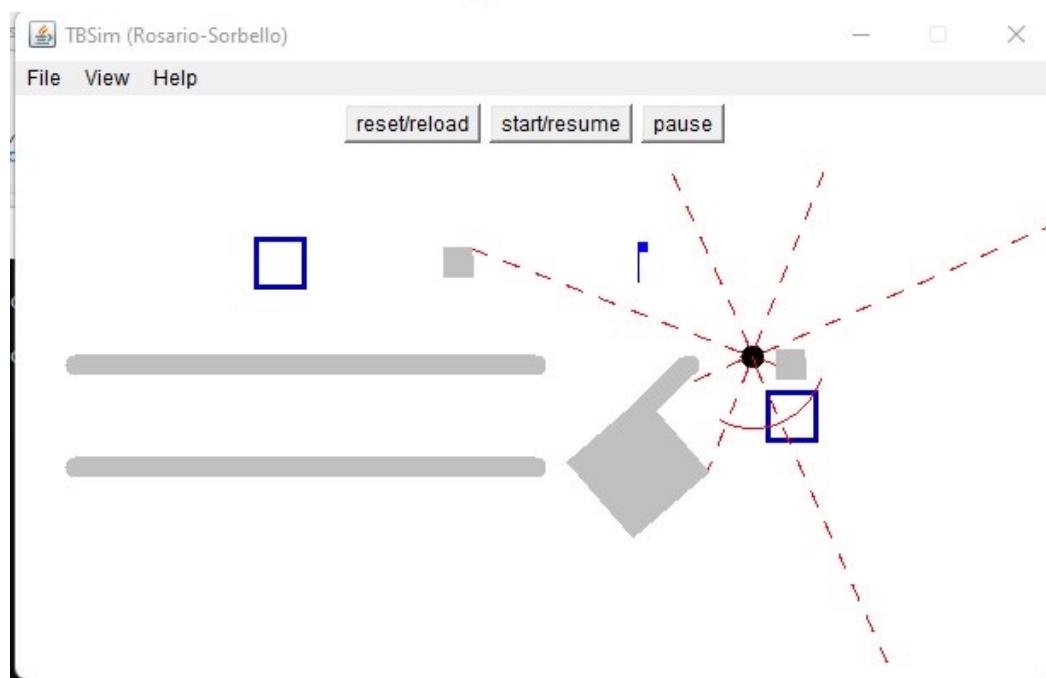
PROF. ROSARIO SORBELLO



## ROBOT MULTIFORAGEN150EXPLORE

Le caratteristiche che contraddistinguono il **MultiForageN150Explore** sono:

- una cintura di 16 sonar, disposti circolarmente a  $360^\circ$  lungo tutto il corpo del robot,
- un sensore laser ottico, che esegue una scansione mantenendosi all'interno di un angolo di  $45^\circ$  rispetto alla direzione del robot.



PROF. ROSARIO SORBELLO

SONAR

---

I sonar disposti sul corpo del **MultiForageN150Explore** permettono di rilevare qualunque ostacolo che si trovi nel loro raggio di azione. È possibile utilizzare tutti i sonar contemporaneamente, una parte o nessuno. Soltanto i sonar accesi vengono utilizzati per individuare gli ostacoli.

Le rilevazioni fatte dal sonar non sono mai precise al 100% ma presentano un certo errore che simula i malfunzionamenti dei sistemi reali che utilizzano i sonar.

Sono comunque presenti tre livelli di precisione (*Low, Medium, High*) che permettono di far variare il valore dell'errore massimo, il quale è influenzato anche da una componente che è direttamente proporzionale alla distanza dal robot dell'oggetto rilevato.

PROF. ROSARIO SORBELLO



## LASER

---

Oltre alla cintura di sonar, il **MultiForageN150Explore** è dotato di un laser ottico che effettua una scansione dell'ambiente mantenendosi all'interno di un angolo di 45° rispetto alla direzione del robot. Il laser può essere acceso o spento.

La precisione delle rilevazioni eseguite con il laser è di gran lunga maggiore rispetto a quelle ottenute con il sonar e non dipende dalla distanza dell'oggetto individuato.

C'è da sottolineare il fatto che sia il laser che il sonar possono esclusivamente rilevare la presenza di un ostacolo, ma non possono dire nulla riguardo la natura di tale ostacolo.



## IMPIEGO DELLA SENSORISTICA

---

Le rilevazioni dei sonar e del laser ottico possono essere utilizzate in vari modi:

- per ricostruire l'ambiente in cui il robot si trova a partire dalla posizione degli ostacoli,
- per effettuare un calcolo più preciso delle traiettorie da effettuare per raggiungere un goal evitando efficacemente gli ostacoli (utilizzando il metodo dei campi magnetici per trovare tali traiettorie),
- per cercare eventuali porte o punti di uscita da stanze o labirinti.



## IMPIEGO DELLA SENSORISTICA

---

Innanzitutto va detto che sia sonar che laser hanno un indice di errore che dipende dalle proprietà fisiche del sistema stesso. Il laser ha un indice di errore minore del sonar.

L'altra componente dell'errore, per ogni singola rilevazione, è direttamente proporzionale alla distanza dell'oggetto rilevato dal robot.

In questo modo si potrebbe pensare di attribuire un peso ad ogni rilevazione che dipenda appunto dal tipo di sensore utilizzato e dalla distanza alla quale si trova l'oggetto identificato.

Inoltre, si supponga di voler rappresentare la conoscenza tramite una rappresentazione a griglia del mondo, e che ogni cella relativa ad una zona della mappa contenga una variabile intera `occupied` con valore compreso tra 0 e 1 che quantifica la certezza con cui un ostacolo si trovi in tale cella (il valore 1 corrisponde al fatto certo).

A questo punto, ogni volta che un ostacolo viene rilevato in una posizione relativa ad una cella, il valore della sua variabile `occupied` viene incrementato del peso assegnato alla rilevazione. Non appena il valore di `occupied` di una cella **X** supera il valore 1, si può affermare che un ostacolo si trova in posizione **X**, e di conseguenza si aggiorna la conoscenza che il robot ha del mondo.

PROF. ROSARIO SORBELLO

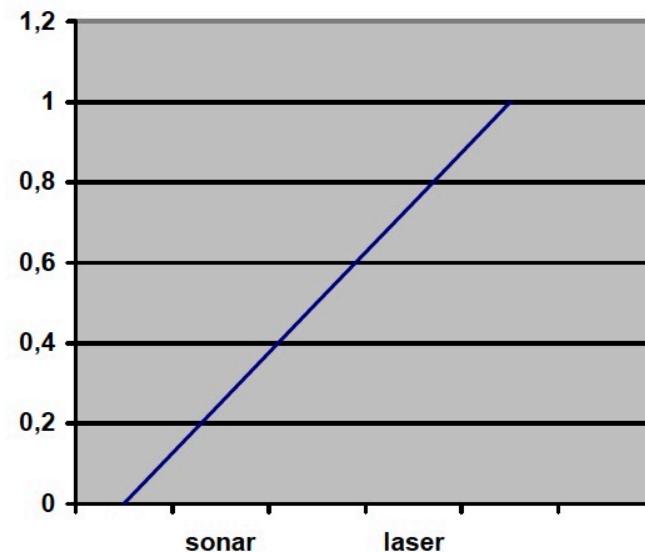


## PRECISIONE DISPOSITIVO E RILEVAZIONE

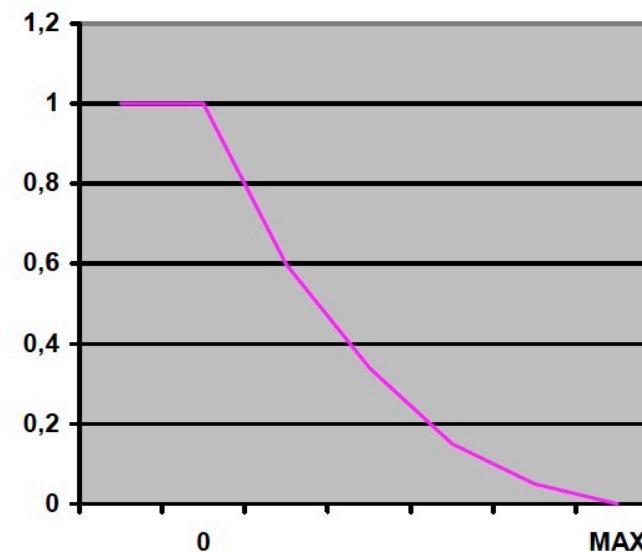
Il primo ha come ascissa il valore associato ai diversi dispositivi sonar e laser, mentre il secondo la distanza relativa dell'oggetto rilevato.

A questo punto possiamo ad esempio supporre di moltiplicare i risultati relativi ai due insiemi *fuzzy*, per ottenere il peso associato ad ogni singola rilevazione.

**Precisione dispositivo**



**Precisione rilevazione**

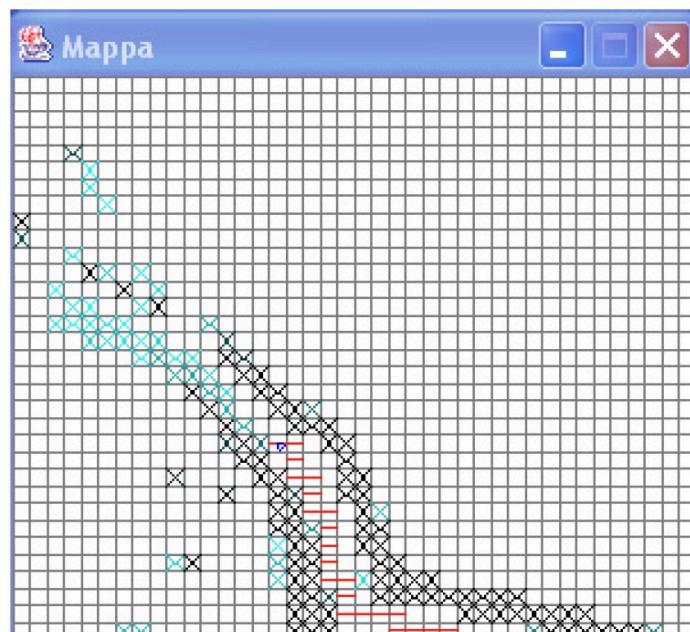


PROF. ROSARIO SORBELLO



## MAPPA

All'interno del package **CSAI.unipa.knowledgment** sono fornite le classi per implementare e gestire una mappa. Il modo in cui usare tale mappa può essere visto dall'esempio fornito all'interno della cartella \Esempi\gara in cui un robot **MultiForageN150Explore** viene immesso in un campo di gara di cui conosce soltanto la posizione di alcune basi che deve conquistare una dietro l'altra, ed aumenta la sua conoscenza sul mondo man mano che procede nella sua esplorazione, generando



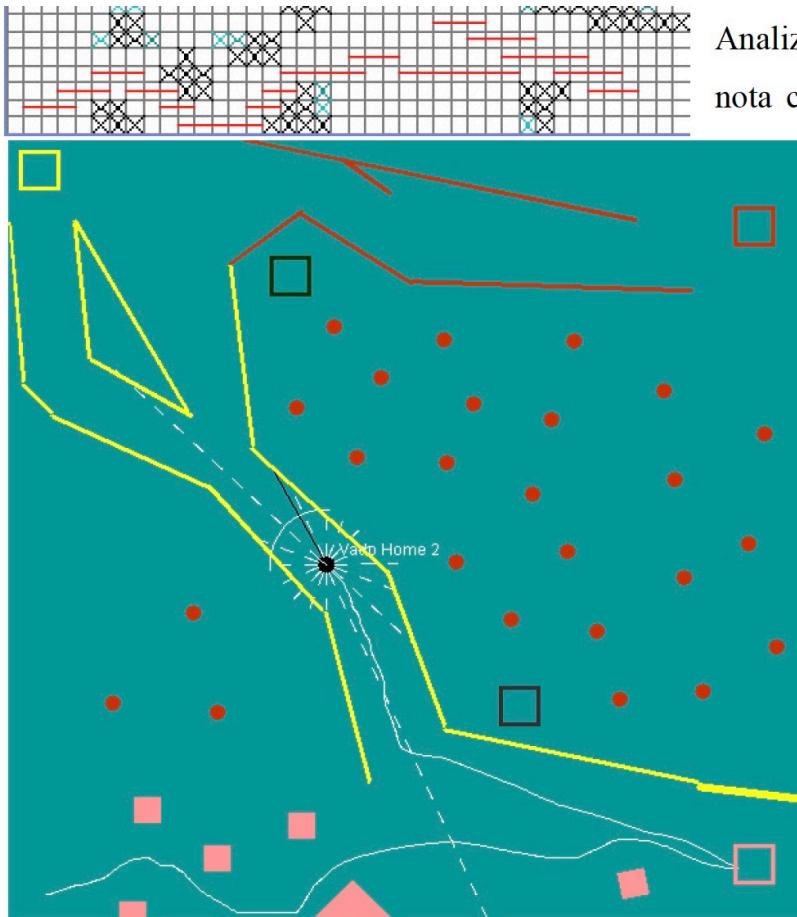
una mappa che utilizza la logica *fuzzy* descritta in precedenza.

Un ostacolo “certo” viene disegnato nella mappa come una x di colore nero che occupa la cella in cui l'ostacolo si trova, mentre ostacoli non ancora del tutto “certi” vengono disegnati come una x di colore blu con intensità che aumenta man mano che aumenta la certezza che quell'ostacolo effettivamente sia presente (e cioè man mano che viene segnalato dai sensori).

PROF. ROSARIO SORBELLO



## MAPPA



Analizzando la mappa ricostruita in figura si nota come la certezza di un ostacolo dipenda dalla sua vicinanza al robot (il percorso del robot è individuato da dei trattini orizzontali rossi, mentre la sua posizione da un cerchietto blu); gli ostacoli più lontani sono quelli meno certi, ancora da verificare.

Una volta effettuata la costruzione della mappa si potrebbe pensare di addestrare una rete neurale per effettuare il riconoscimento di stanze, di porte, di ostacoli pericolosi, vicoli ciechi e quant'altro possa servire ai fini della simulazione.

PROF. ROSARIO SORBELLO



L'uso dei sonar può portare a notevoli vantaggi anche quando i rilevamenti vengono utilizzati per il calcolo di traiettorie efficienti nel raggiungimento di un prefissato punto, evitando tutti gli ostacoli che si trovano davanti. È possibile vedere come nell'esempio precedentemente citato \\\'Esempi\\\'Gara il robot si muova con una certa facilità all'interno di ambienti molto complessi, semplicemente utilizzando tutti i suoi sensori per rilevare il maggior numero di ostacoli e quindi pianificare traiettorie che tengano conto di una situazione più generale possibile.

Si rende molto efficiente l'uso dei sonar in particolare quando il robot si trova a passare lungo stretti corridoi, in quanto il numero elevato di ostacoli individuati lo obbliga a muoversi mantenendosi sempre a equa distanza dalle pareti, anche se il punto di attrazione si trova al di là di una delle due pareti.



## **5.1. Importare le classi**

Innanzi tutto deve essere inserita la cartella CSAI all'interno della directory \\tb\\src\\.

Si importano i seguenti package:

```
import CSAI.unipa.clay.*  
import CSAI.unipa.knowledgment.NodeMap  
import CSAI.unipa.abstractrobot.*
```

e deve essere dichiarata la classe del robot come estensione della classe:

```
ControlSystemMFN150Explore
```



## 5.2. Dichiarare le variabili

Tra le variabili globali si dichiarano le seguenti:

```
...
    private NodeBoolean          sonar_configuration
    private NodeBoolean          laser_configuration
    private NodeVec2Array        PS SONAR
    private NodeVec2Array        PS LASER
    private NodeMap              MAP
...

```

Le variabili `sonar_configuration` e `laser_configuration` vengono utilizzate rispettivamente per definire il comportamento dei sonar e del laser nei vari stati dell'FSA.

Le variabili `PS SONAR` e `PS LASER` contengono il risultato delle rilevazioni del sonar e del laser.

La variabile `MAP` è la classe che gestisce la mappa.



### **5.3. Metodo *configure()***

Il **MultiForageN150Explore** di *default* ha tutti i sonar accesi, utilizza il livello di precisione **MEDIUM** per la rilevazione dei dati, ed ha il laser ottico attivato.

È possibile modificare questi valori iniziali utilizzando nel *configure()* i metodi:

```
...
    public void turnOffSonar(int item)
    public void turnOnSonar(int item)
    public void turnOffAllSonar()
    public void turnOnLaser()
    public void turnOffLaser()
    public void setSonarPrecision(double prec)
...
...
```

I valori di precisione del sonar sono contenuti all'interno della classe *SonarObjectSensor*:

```
...
    SonarObjectSensor.LOW
    SonarObjectSensor.MEDIUM
    SonarObjectSensor.HIGH
...
...
```

PROF. ROSARIO SORBELLO



---

Sempre all'interno del metodo *configure()* va instanziata la mappa invocandone il costruttore a cui vanno passati come parametro: i limiti (in ordine left, right, top, bottom) che definiscono la mappa, la dimensione delle celle (che sono quadrate), il raggio di appartenenza ad una cella (indica quale è la distanza massima a partire dal centro di una cella, per cui un punto debba essere considerato appartenente a questa), il nome della mappa, e il robot che la sta creando, ad esempio:

```
...
    MAP = new NodeMap(-10, 10, 10, -10, 0.5, 0.4, "Mappa ",
                      true, abstract_robot);
...

```

A questo punto si associa il rilevamento di sonar e laser alle variabili *NodeVec2Array* appropriate, e si fondono insieme in un'unica variabile *PS\_ALL\_OBS* anch'essa di tipo *NodeVec2Array*:



```
...
    PS_OBS = new va_Obstacles_r(abstract_robot);

    PS SONAR = new va_SonarSensed_r(abstract_robot);

    PS LASER = new va_LaserSensed_r(abstract_robot);

    NodeVec2Array
    PS TEMP = new va_Merge_vava(PS LASER, PS SONAR);

    NodeVec2Array
    PS ALL OBS = new va_Merge_vava(PS TEMP, PS OBS);
...

```

La variabile `PS_ALL_OBS` contiene al suo interno tutti gli ostacoli che sono stati rilevati per mezzo del laser, del sonar, e del vecchio sistema di rilevamento, e va usata per calcolare le traiettorie che evitano gli ostacoli per mezzo della classe `v_Avoid_Sonar` contenuta nel package **CSAI.unipa.clay** invocandone il costruttore in questo modo:

```
...
    NodeVec2
    MS_AVOID_OBSTACLES = new v_AvoidSonar_va(3.0, abstract_robot.RADIUS+0.1,
                                                PS_ALL_OBS, PS_GLOBAL_POS);
...

```

PROF. ROSARIO SORBELLO



**ROB**

A questo punto vanno definiti i comportamenti utilizzando la `MS_AVOID_OBSTACLES` sopra definita. L'ultimo passo da fare è quello di definire il comportamento del sonar e del laser relativo ad ogni stato dell'FSA. Si supponga per esempio di avere un automa formato da 3 stati.

Si associa il comportamento del sonar all'FSA in questo modo:

```
...
d_SonarControl_ir
SONAR_CONFIGURATION=new d_SonarControl_ir(STATE_MACHINE,abstract_robot);
SONAR_CONFIGURATION.sonarActived[0] = new
                                int[]{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
SONAR_CONFIGURATION.sonarPrecision[0] = SonarObjectSensor.MEDIUM;
SONAR_CONFIGURATION.sonarActived[1] = new
                                int[]{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
```

```
SONAR_CONFIGURATION.sonarPrecision[1] = SonarObjectSensor.MEDIUM;
SONAR_CONFIGURATION.sonarActived[2] = new int[]{4,12};
SONAR_CONFIGURATION.sonarPrecision[2] = SonarObjectSensor.HIGH;
SONAR_CONFIGURATION.sonarActived[3] = new int[]{0};
SONAR_CONFIGURATION.sonarPrecision[3] = SonarObjectSensor.HIGH;
```

```
...
```



---

Per quanto riguarda il laser, se ne associa il comportamento all'FSA in questo modo:

---

```
...
    b_Select_ir
    LASER_CONFIGURATION = new b_Select_ir(STATE_MACHINE, abstract_robot);

    LASER_CONFIGURATION.embedded[0] = true;
    LASER_CONFIGURATION.embedded[1] = false;
    LASER_CONFIGURATION.embedded[2] = true;
    LASER_CONFIGURATION.embedded[3] = false;
...
...
```

Infine si associano questi comportamenti alle variabili globali che selezioneranno il comportamento adeguato dipendentemente dallo stato in cui il robot si trova:

```
...
    laser_configuration = LASER_CONFIGURATION
    sonar_configuration = SONAR_CONFIGURATION
...
...
```

PROF. ROSARIO SORBELLO



All'interno del *takeStep()* bisogna innanzi tutto gestire la mappa, invocando i metodi che effettuano i controlli sulla posizione del robot, e sui rilevamenti degli ostacoli.

I metodi da invocare sono i seguenti:

```
...
    MAP.setCellVisited(curr_time);
    MAP.setObstacle(PS SONAR.Value(curr_time),
                    abstract robot.getPrecisionRivelation(), curr_time);

    MAP.setObstacle(PS LASER.Value(curr_time),
                    LaserFinderObjectSensor.FUZZY, curr_time);
...
}
```

Il metodo *setCellVisited()* serve per settare come visitata la cella in cui il robot si trova in quel momento (escludendo quindi la possibilità che un ostacolo vi si possa trovare sopra).

Il metodo *setObstacle()* va invocato passandogli come parametri:

- il risultato delle rilevazioni del sonar (o laser),
- l'array che contiene i pesi delle singole rilevazioni. Nel caso del sonar viene restituito invocando il metodo *abstract\_robot.getPrecisionRivelation()* mentre per il laser (l'indice di errore non dipende dalla distanza ma soltanto dalle caratteristiche del dispositivo stesso) è contenuto all'interno della variabile statica *LaserFinderObjectSensor.FUZZY*,
- l'istante di tempo in cui si sono effettuate le rilevazioni.



---

Infine deve essere aggiornato il comportamento del sonar e del laser a seconda dello stato in cui ci si trova scrivendo le seguenti righe di codice:

```
...
    //SONAR
    bresult = sonar_configuration.Value(curr_time);

    //LASER
    bresult = laser_configuration.Value(curr_time);
...
```

Essendo stata `bresult` definita precedentemente come una variabile *booleana*.

I metodi `Value()` relativi a `sonar_configuration` e `laser_configuration` ritornano valore *true* ogni volta che rispettivamente le impostazioni del sonar o del laser vengono modificate (cioè ad ogni passaggio di stato, se sono stati definiti comportamenti per ogni stato).



OSAKA UNIVERSITY



## PIANIFICAZIONE

PROF. ROSARIO SORBELLO

5° LEZIONE - PALERMO 24 OTTOBRE 2022

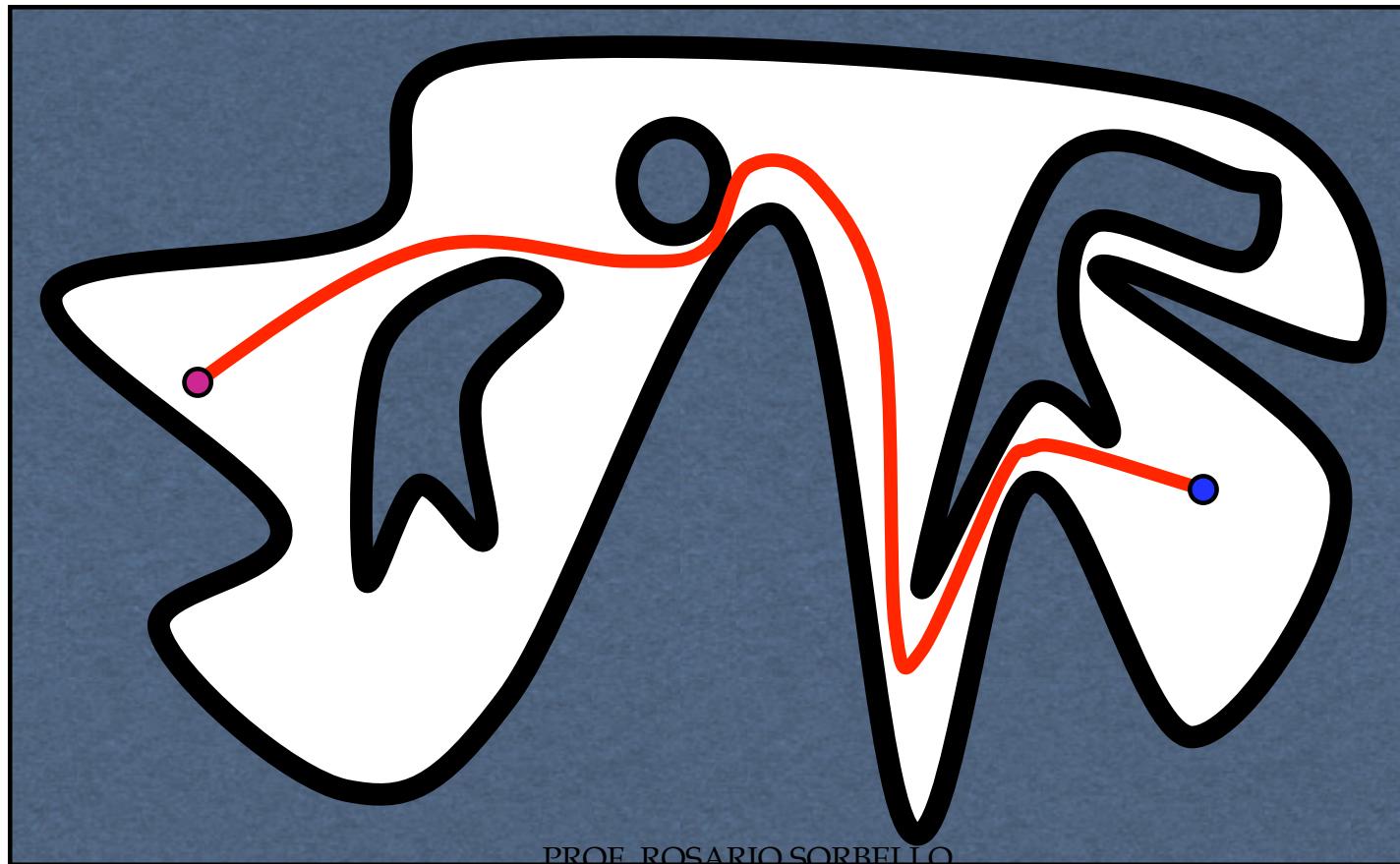


- Calcolare strategie di moto:
  - Cammini geometrici
  - Traiettorie parametrizzate
  - Sequenze di comandi motori
- Conseguire obiettivi:
  - Andare da A a B evitando ostacoli
  - Assemblare il prodotto P



ROBOTICS LAB

## CAMMINI GENERALIZZATI

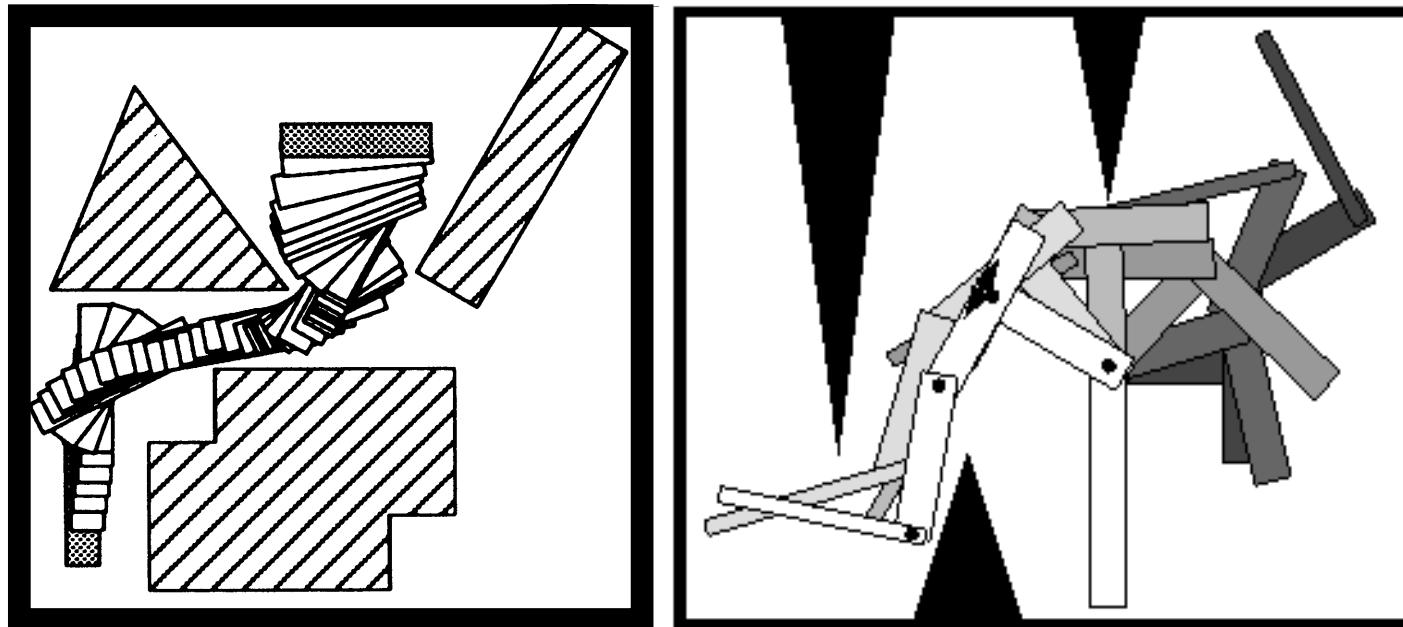


PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## CAMMINI GENERALIZZATI

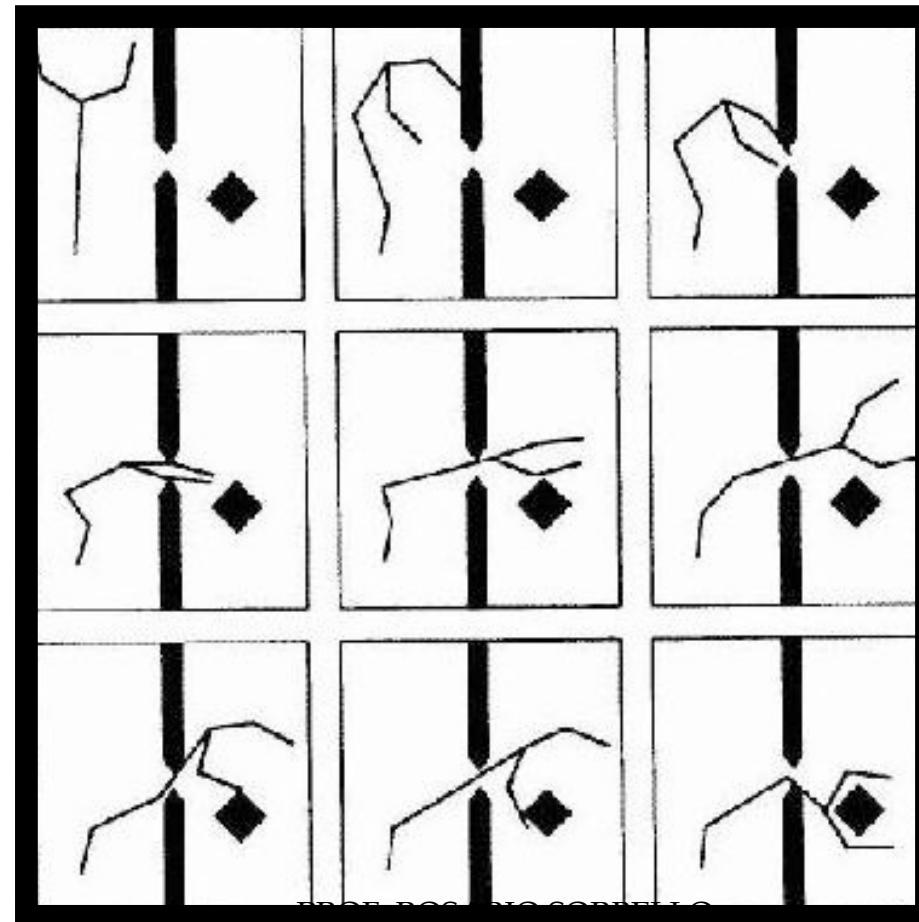


PROF. ROSARIO SORBELLO



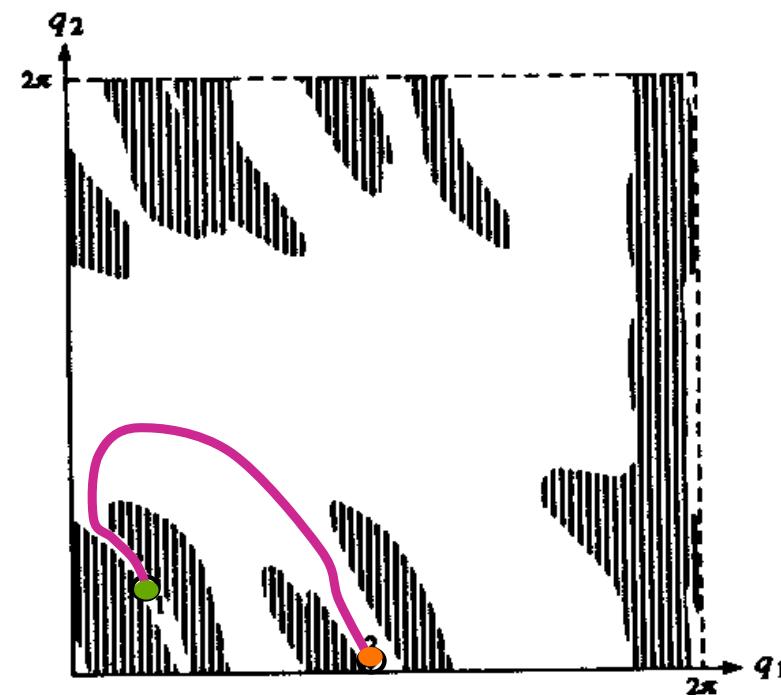
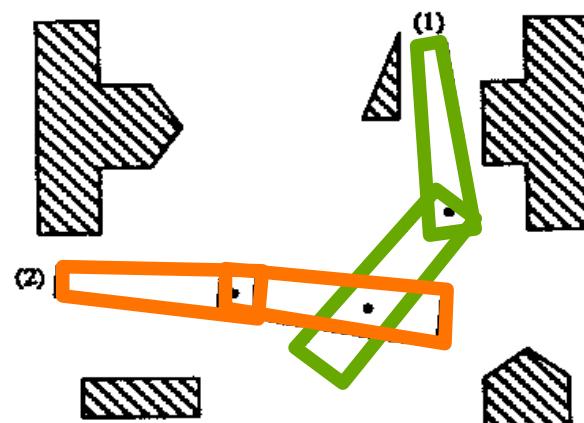
**ROBOTICS LAB**

## CAMMINI GENERALIZZATI





# Cammini generalizzati



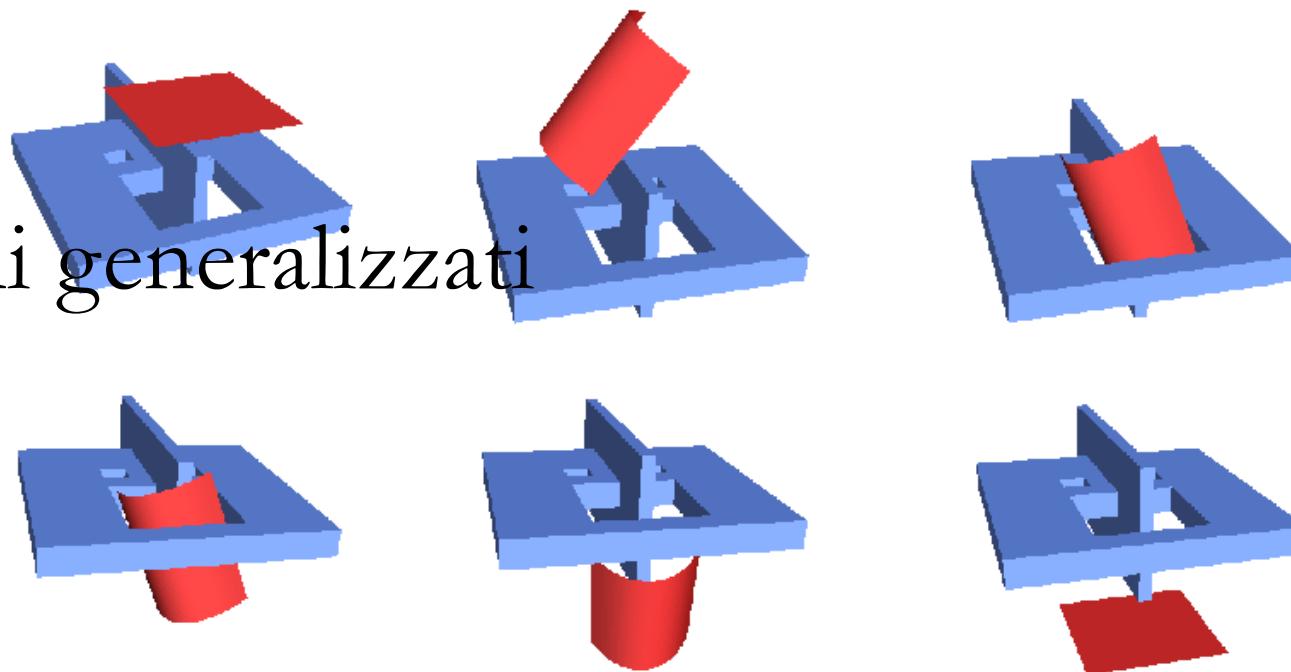
PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## CAMMINI GENERALIZZATI

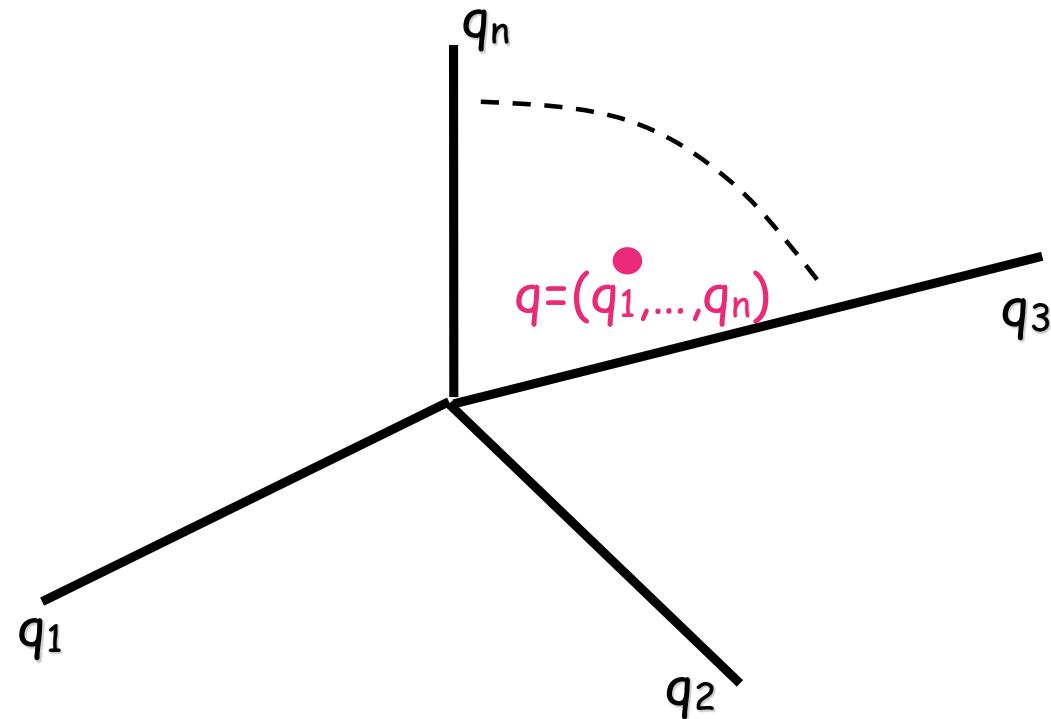
Cammini generalizzati



PROF. ROSARIO SORBELLO



## GRADI DI LIBERTA' DEL ROBOT



PROF. ROSARIO SORBELLO

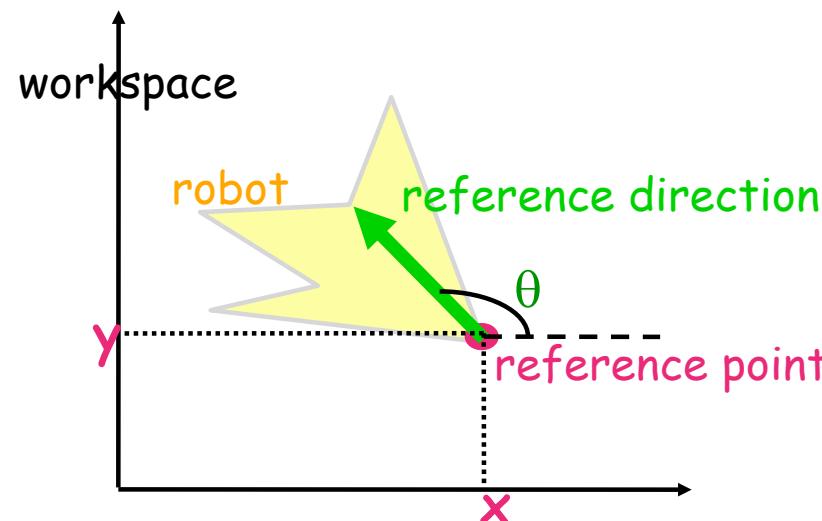


## DEFINIZIONE

---

- Una configurazione è una specifica delle posizioni dei punti del robot relativi ad un sistema di coordinate
- Una configurazione è espressa tramite un vettore che riassume i parametri di posizione e orientazione

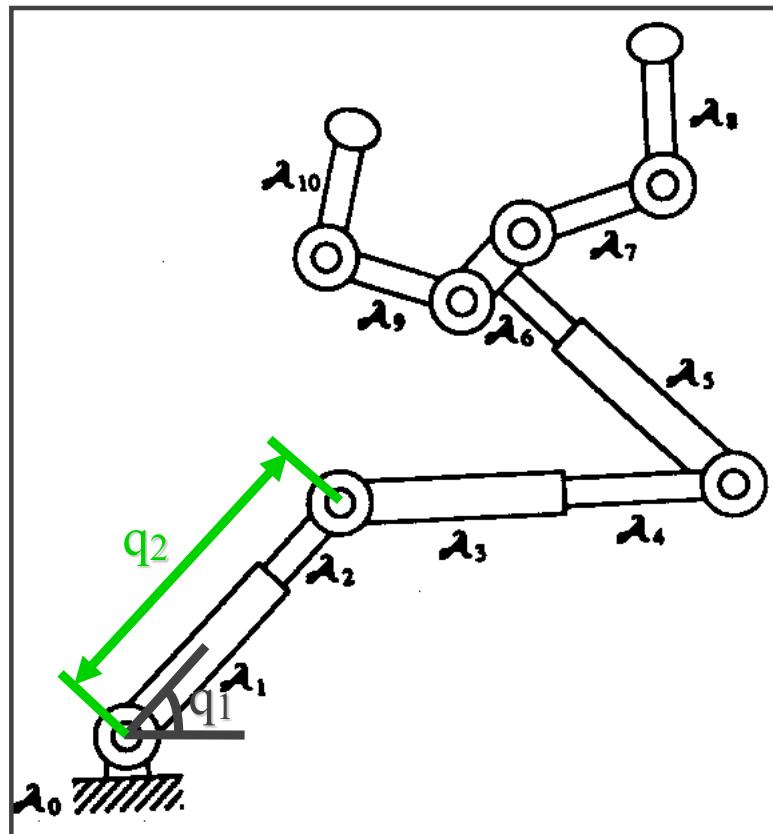
## ESEMPIO



- Robot bidimensionale:  $q = (x, y, \theta)$
- In 3D  $q$  sarebbe  $(x, y, z, \alpha, \beta, \gamma)$



## ROBOT ARTICOLATO

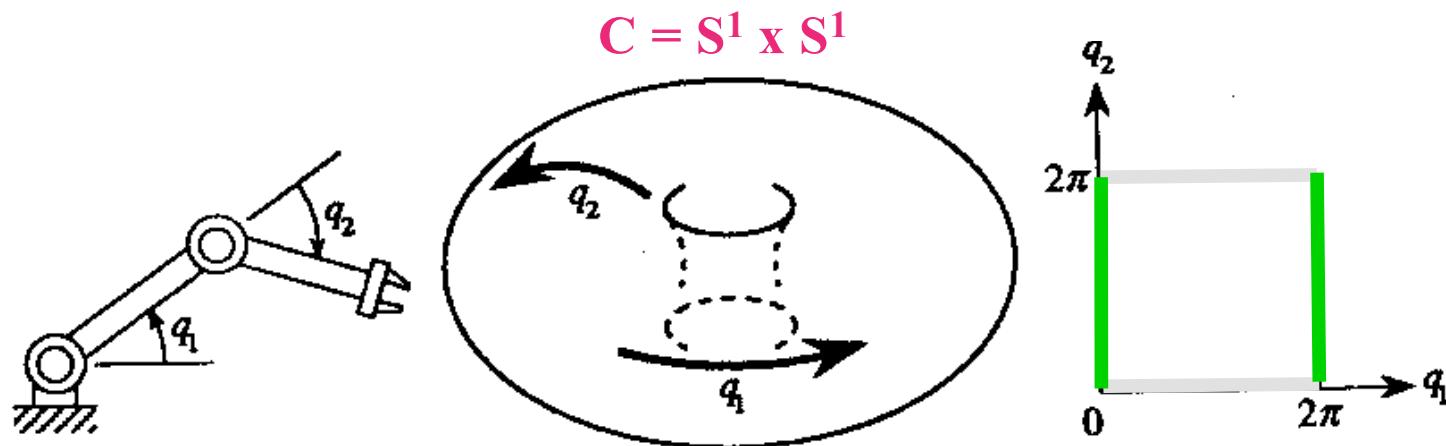


$$\mathbf{q} = (q_1, q_2, \dots, q_{10})$$

PROF. ROSARIO SORBELLO

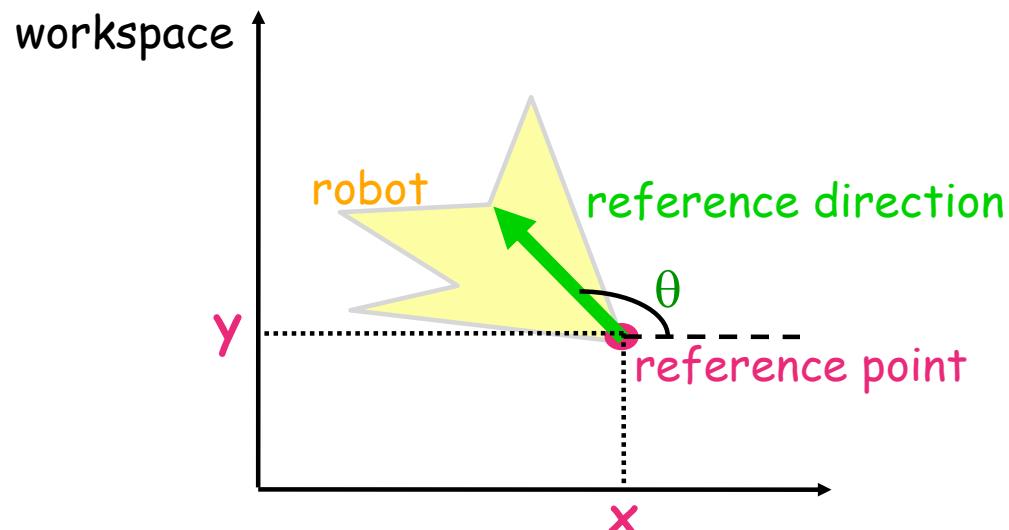
## SPAZIO DELLE CONFIGURAZIONI

- Spazio di tutte le possibili configurazioni
- La topologia non è cartesiana
- Manifold



PROF. ROSARIO SORBELLO

## ROBOT PLANARE



- Rappresentazione minima con 3 parametri:  $q = (x, y, \theta)$  con  $\theta \in [0, 2\pi)$ .
- Altra rappresentazione:  $q = (x, y, \cos\theta, \sin\theta)$

ROBOT IN 3 DIMENSIONI

---

- $q = (x, y, z, a, b, g)$
- Altra rappresentazione:  $q = (x, y, z, r_{11}, r_{12}, \dots, r_{33})$  dove

$r_{11}, r_{12}, \dots, r_{33}$  sono gli elementi della matrice di rotazione R:  
con:

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

$$- r_{i1}^2 + r_{i2}^2 + r_{i3}^2 = 1$$

$$- r_{i1}r_{j1} + r_{i2}r_{j2} + r_{i3}r_{j3} = 0$$

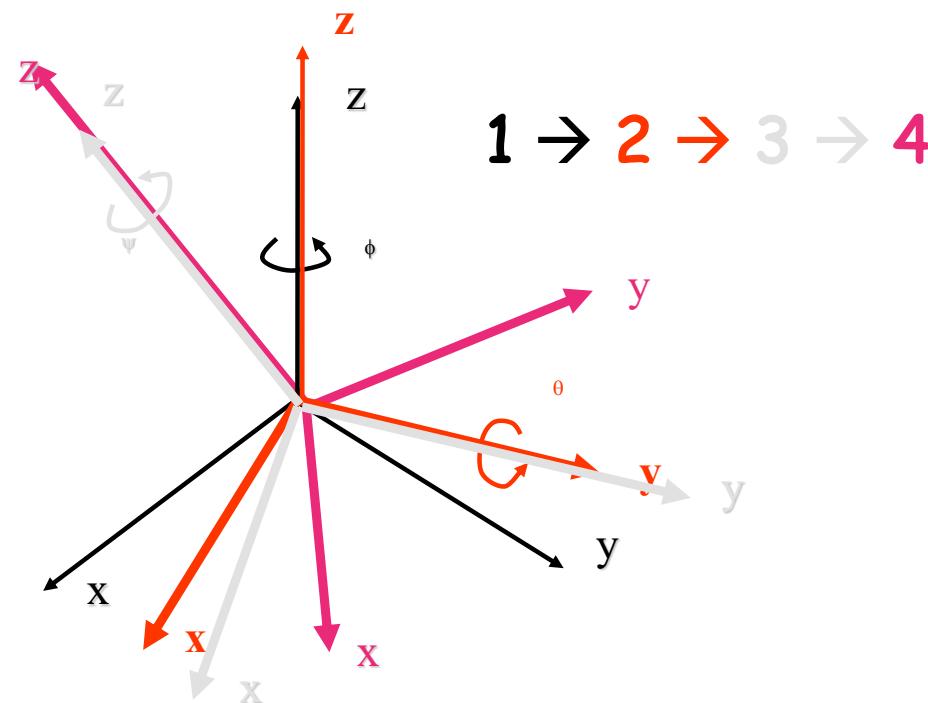
$$- \det(R) = +1$$

PROF. ROSARIO SORBELLO



## ROTAZIONI

Angoli di Eulero:  $(\phi, \theta, \psi)$





Una metrica  $d$  è una funzione di distanza in  $C$ :

$$d: (q_1, q_2) \in C^2 \rightarrow d(q_1, q_2) \geq 0$$

tale che:

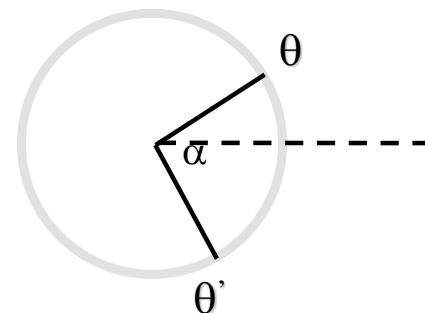
- $d(q_1, q_2) = 0$  se e solo se  $q_1 = q_2$
- $d(q_1, q_2) = d(q_2, q_1)$
- $d(q_1, q_2) \leq d(q_1, q_3) + d(q_3, q_2)$



- Robot **A** e punto  $x$  di **A**
- $x(q)$ : posizione di  $x$  nello spazio di lavoro tale che  $A$  sia nella configurazione  $q$
- La distanza  $d$  in  $C$  è definita come:  
$$d(q,q') = \max_{x \in A} ||x(q)-x(q')||$$
 dove  $||a - b||$  è la distanza euclidea tra i punti  $a$  e  $b$  nello spazio di lavoro

## ESEMPI

- $q = (x, y, \theta)$ ,  $q' = (x', y', \theta')$  con  $\theta, \theta' \in [0, 2\pi)$
- $\alpha = \min\{|\theta - \theta'|, 2\pi - |\theta - \theta'|\}$

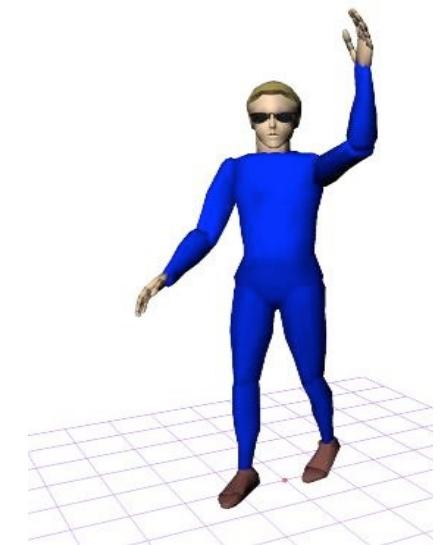
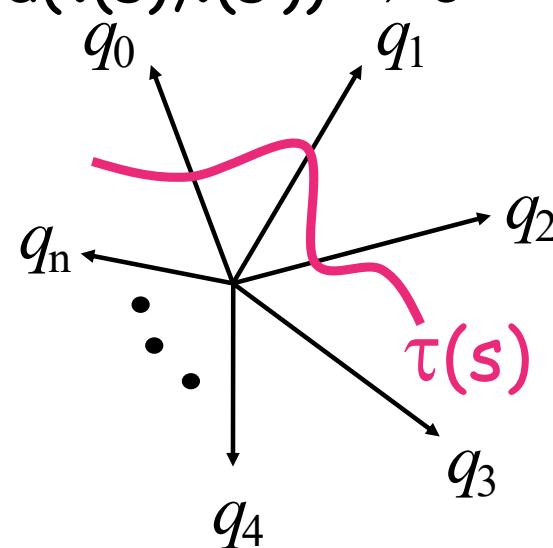


- $d(q, q') = \sqrt{(x - x')^2 + (y - y')^2 + \alpha^2}$
- $d(q, q') = \sqrt{(x - x')^2 + (y - y')^2 + (\alpha\rho)^2}$
- dove  $\rho$  è la max distanza tra il punto di riferimento e il punto del robot



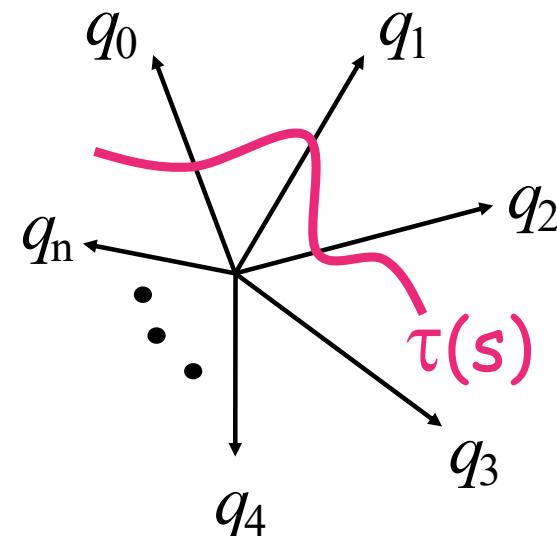
## CAMMINO

- Un cammino in  $C$  è una curva connessa che connette le configurazioni  $q$  e  $q'$ :  
$$\tau : s \in [0,1] \rightarrow \tau(s) \in C$$
- $s' \rightarrow s \Rightarrow d(\tau(s), \tau(s')) \rightarrow 0$



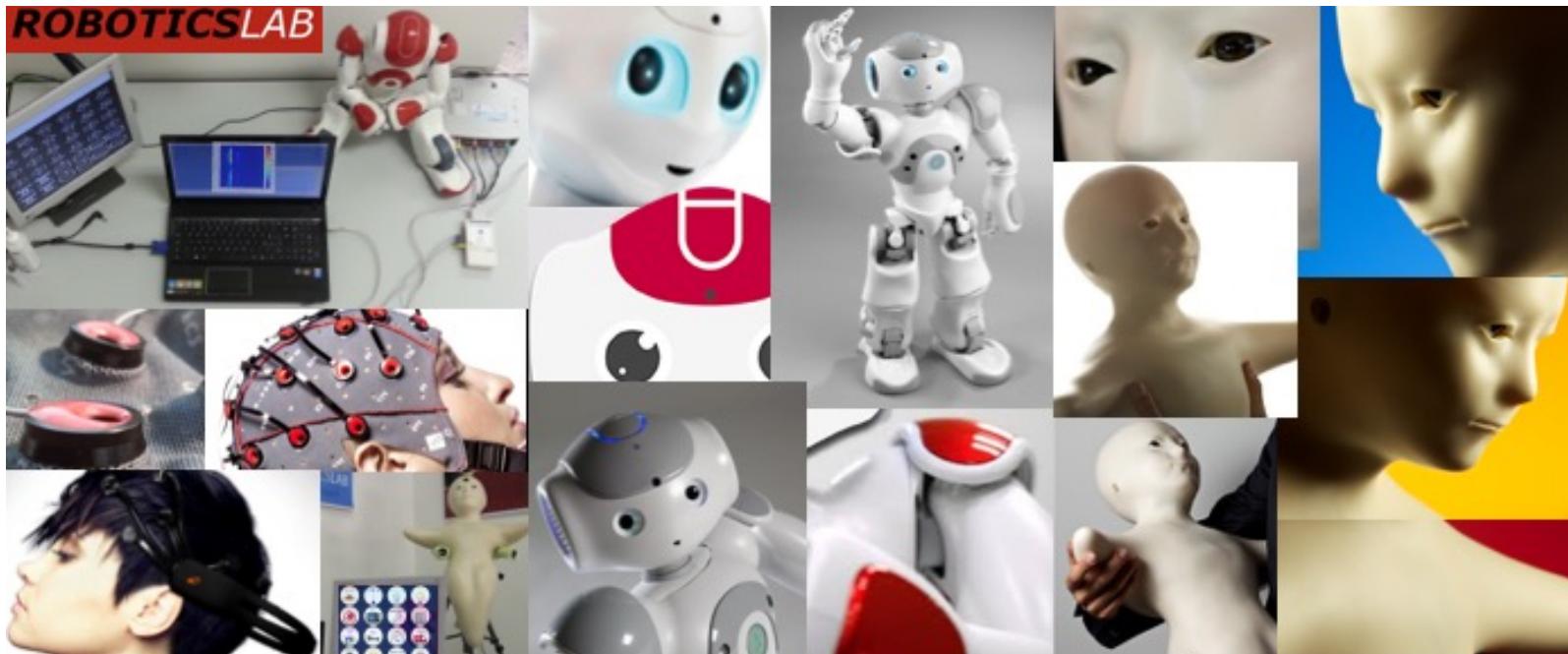
## ALTRI VINCOLI POSSIBILI

- Lunghezza finita, curvatura, continuità, ecc...
- Una traiettoria è un cammino parametrizzato nel tempo:  
 $\tau : t \in [0, T] \rightarrow \tau(t) \in C$





OSAKA UNIVERSITY



ATR

g-tec

## PIANIFICAZIONE

PROF. ROSARIO SORBELLO

7° LEZIONE - PALERMO 28 OTTOBRE 2022

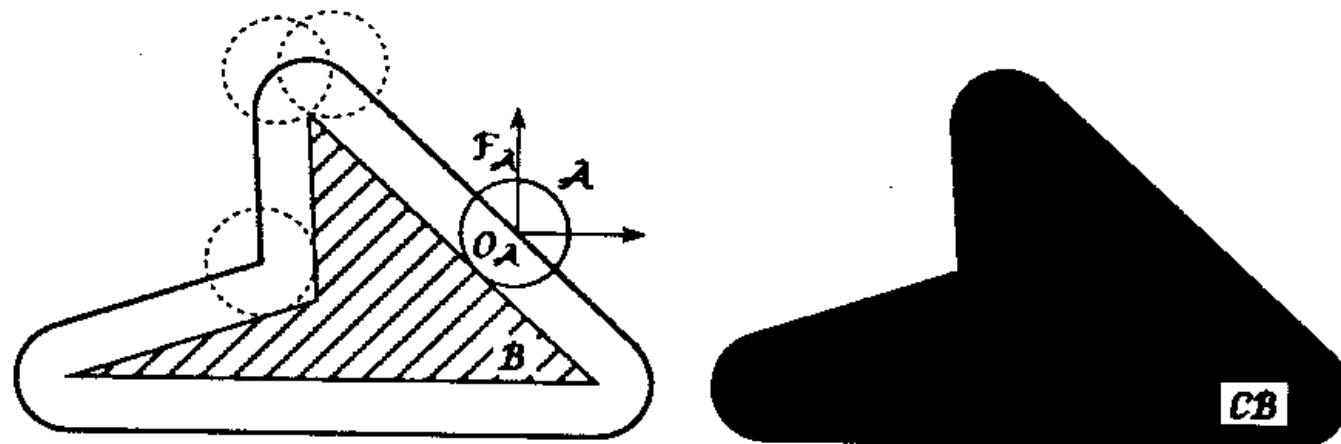


- Una configurazione  $q$  è detta libera da collisioni se il robot piazzato in  $q$  non ha intersezioni con gli ostacoli nello spazio di lavoro
- Lo spazio libero  $F$  è l'insieme delle configurazioni libere da collisioni
- Un C-ostacolo è l'insieme di configurazioni tali che il robot collide con un ostacolo dello spazio di lavoro
- Una configurazione è detta semi-libera se il robot tocca un ostacolo



ROBOTICS LAB

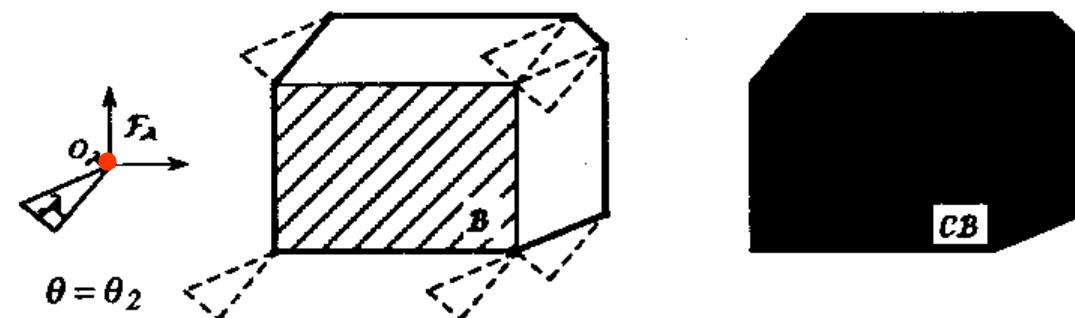
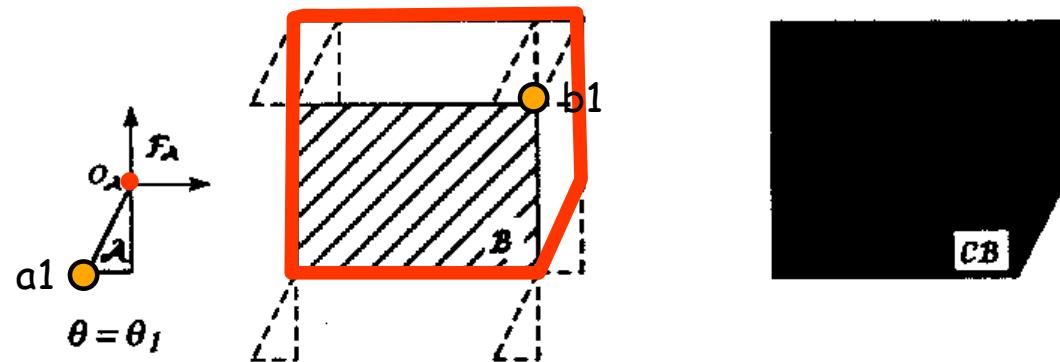
## ROBOT DISCO IN 2 DIMENSIONI



PROF. ROSARIO SORBELLO



## ROBOT RIGIDO IN 2 DIMENSIONI

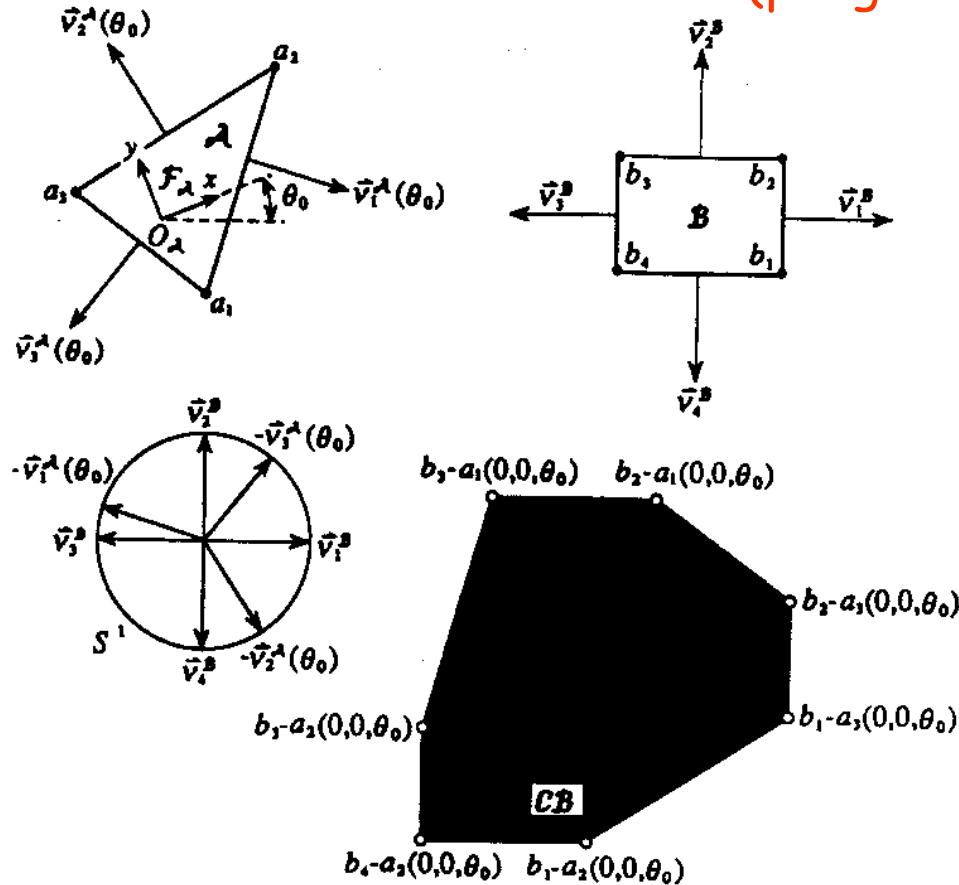


PROF. ROSARIO SORBELLO



## ESTENSIONE DELL'OSTACOLO

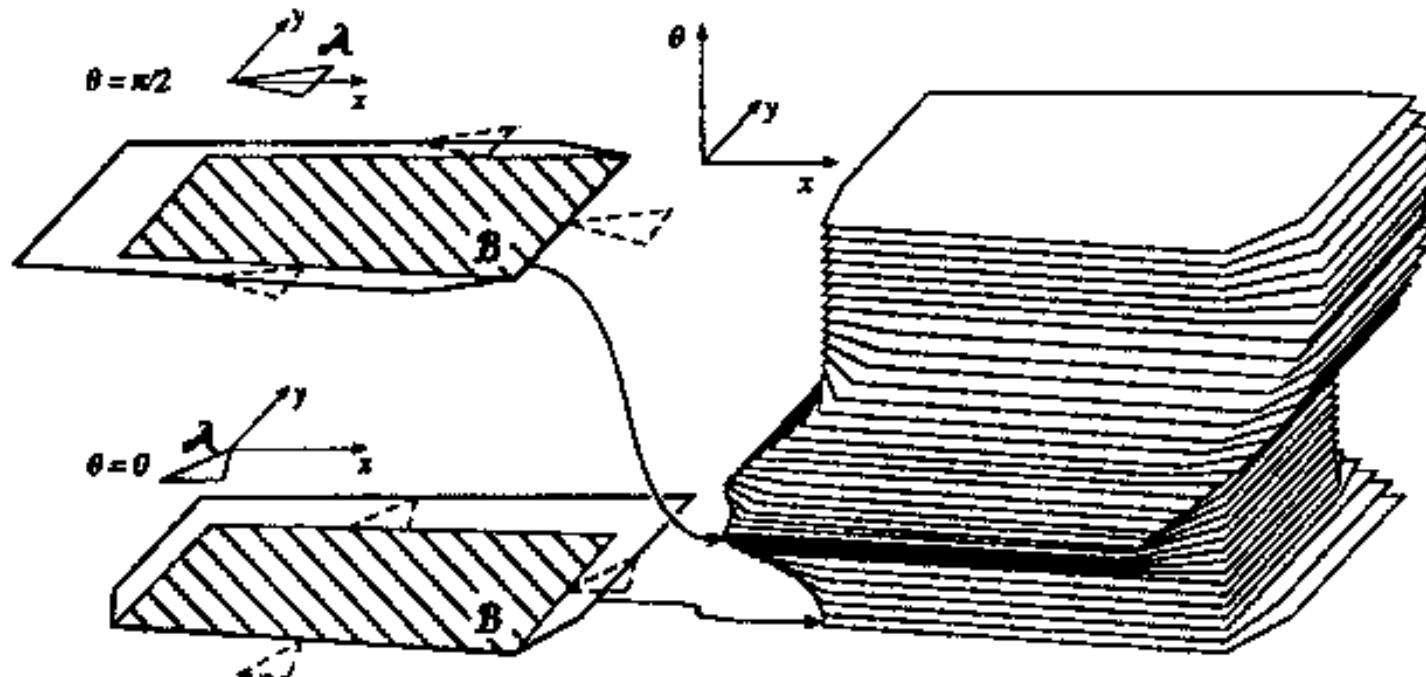
(poligoni convessi)



PROF. ROSARIO SORBELLO



## ROTO-TRASLAZIONE IN 2D

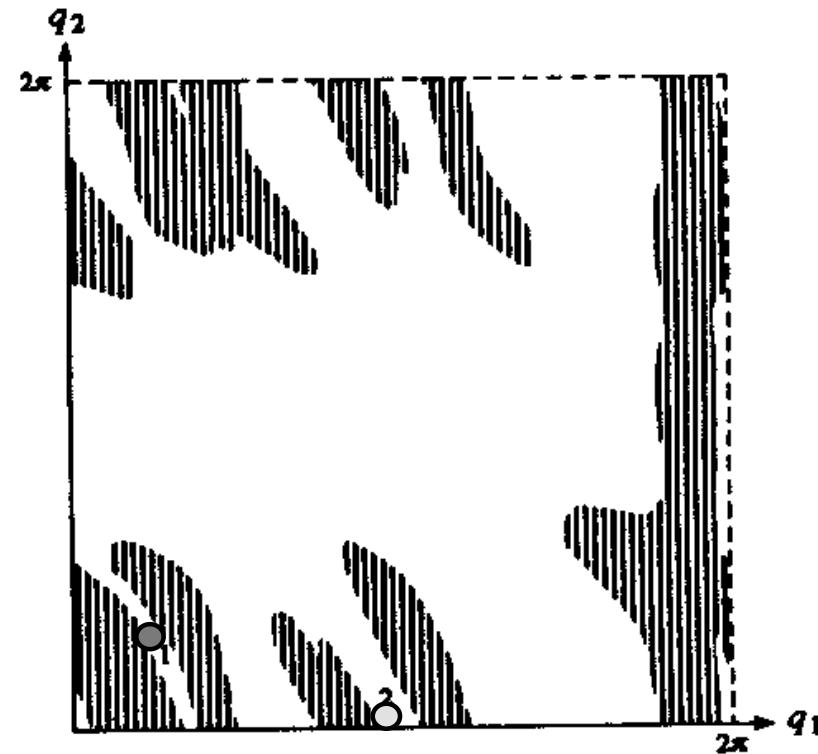
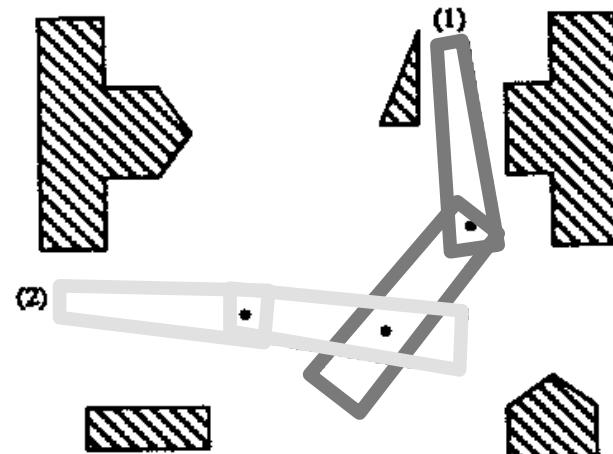


PROF. ROSARIO SORBELLO



ROBOTICS LAB

# SPAZIO DELLE CONFIGURZIONE PER UN ROBOT ARTICOLATO



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## CAMMINO LIBERO E SEMI-LIBERO

---

- Un cammino libero giace interamente nello spazio libero F
- Un cammino semi-libero giace interamente nello spazio semi-libero

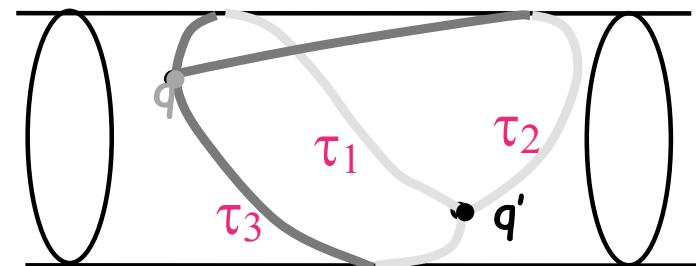
PROF. ROSARIO SORBELLO



- Il robot e gli ostacoli sono modellati come insiemi chiusi, che contengono i loro contorni
- Anche i C-ostacoli sono quindi insiemi chiusi
- Di conseguenza, lo spazio libero  $F$  è un insieme aperto di  $C$ . Ogni configurazione libera è il centro di un cerchio di raggio non zero interamente contenuto in  $F$
- Lo spazio semi-libero è invece un insieme chiuso di  $C$



- Due cammini con gli stessi vertici sono omotopici se uno può essere deformato con continuità nell'altro
- Esempio:
  - $t_1$  e  $t_2$  sono omotopici
  - $t_1$  e  $t_3$  non sono omotopici



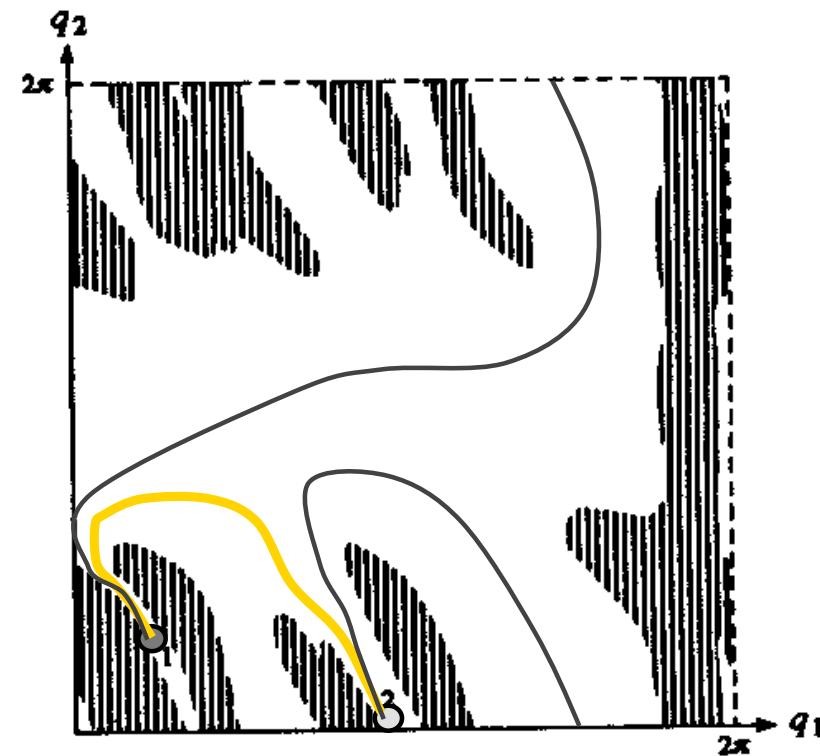
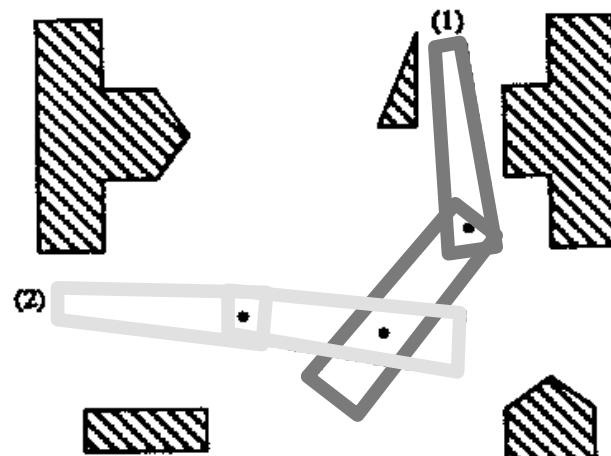


- C è connesso se due configurazioni possono essere connessi da un cammino
- C è semplicemente connesso se due cammini generiche che connettono gli stessi vertici sono omotopici
- Altrimenti C è con connessioni multiple



ROBOTICS LAB

## C-OSTACOLI PER ROBOT ARTICOLATO

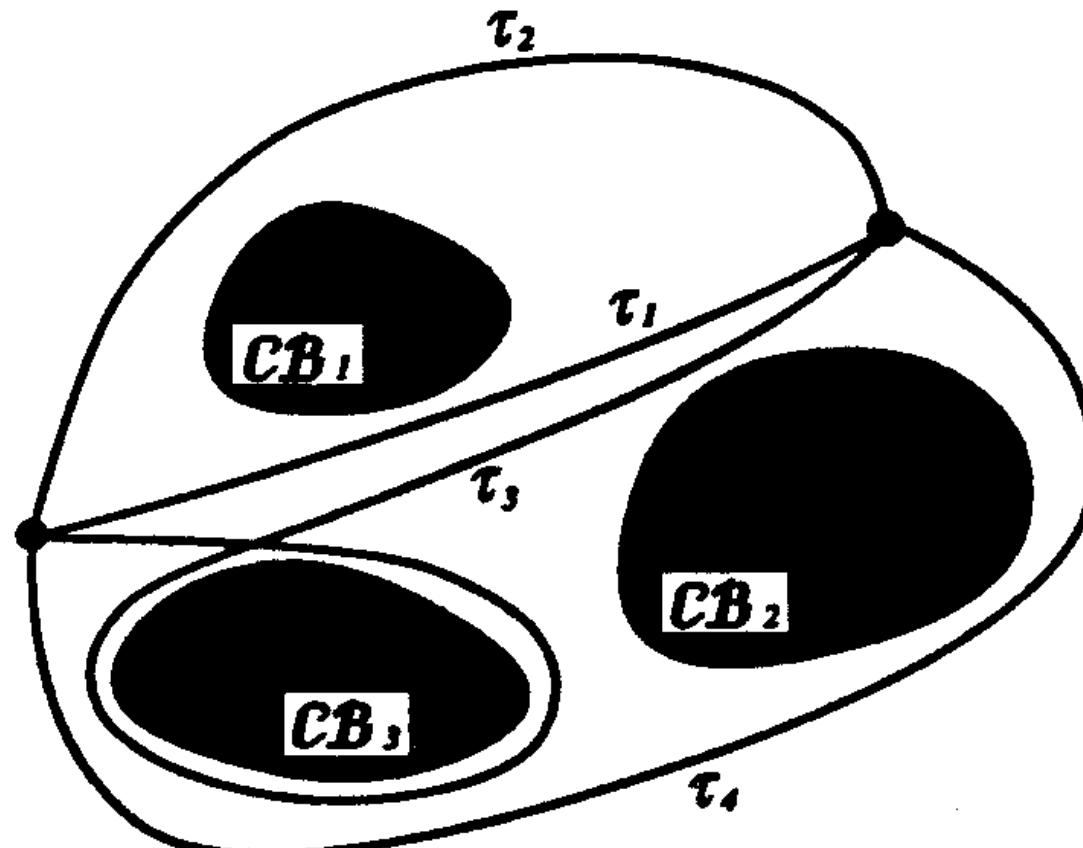


PROF. ROSARIO SORBELLO



ROBOTICS LAB

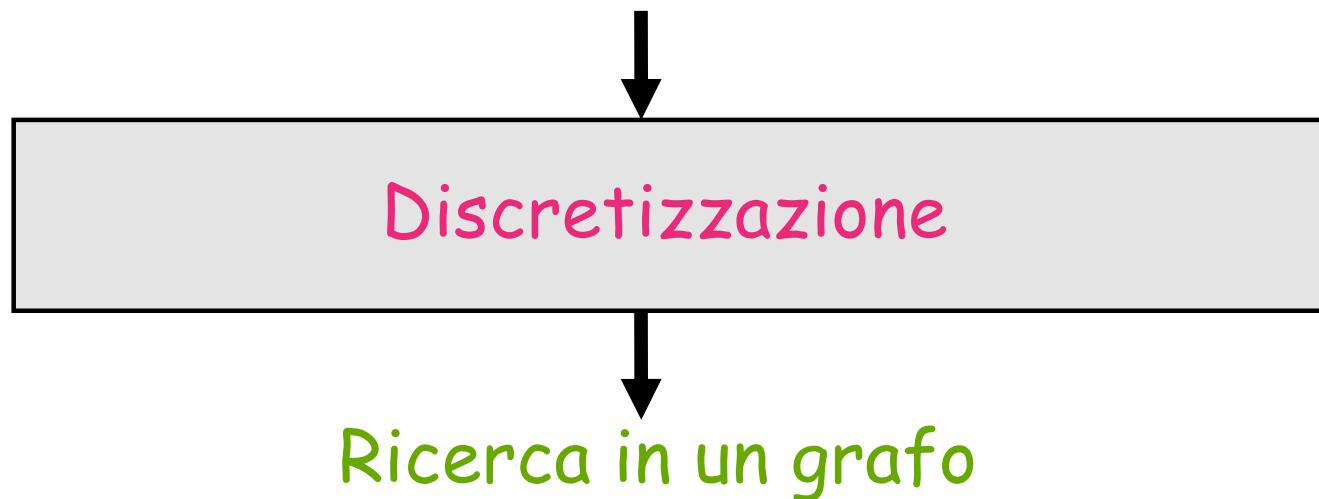
## CLASSI DI CAMMINI OMOTOPICI



PROF. ROSARIO SORBELLO



Rappresentazione continua  
(formulazione dello spazio delle configurazioni)

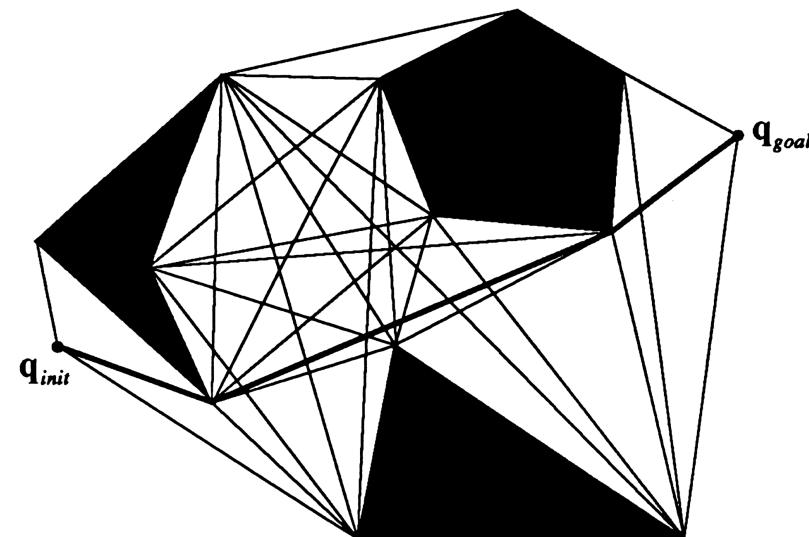




## □ Grafo di visibilità

Introdotto nel robot Shakey negli anni  
60. Produce il cammino minimo in spazi

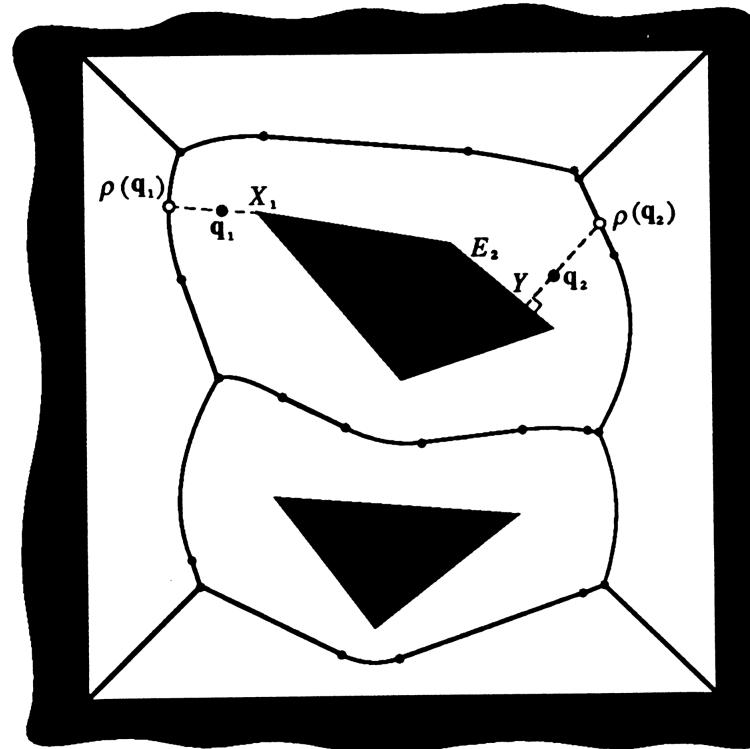
2-D



PROF. ROSARIO SORBELLO



□ **Diagramma di Voronoi**  
Nato da studi sulla Geometria  
Computazionale. Genera  
cammini sicuri. Essenzialmente  
si applica a spazi 2-D





**ROBOTICS LAB**

## SCOMPOSIZIONE IN CELLE

---

### Scomposizione esatta in celle

Lo spazio libero  $F$  è rappresentato da una collezione di celle che non si sovrappongono la cui unione è  $F$

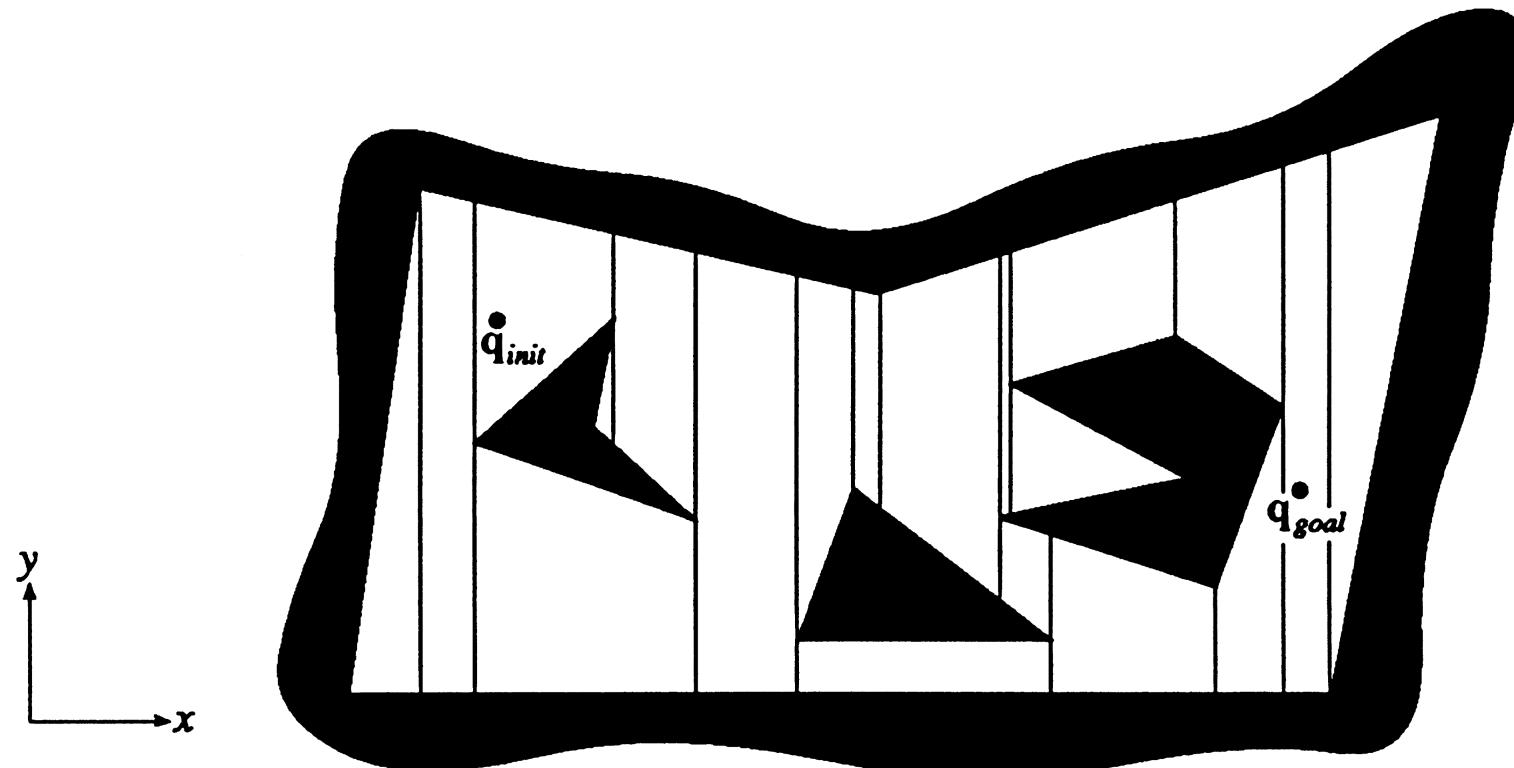
Esempi: scomposizioni trapezoidali

PROF. ROSARIO SORBELLO



ROBOTICS LAB

## SCOMPOSIZIONE TRAPEZIODEALE



PROF. ROSARIO SORBELLO



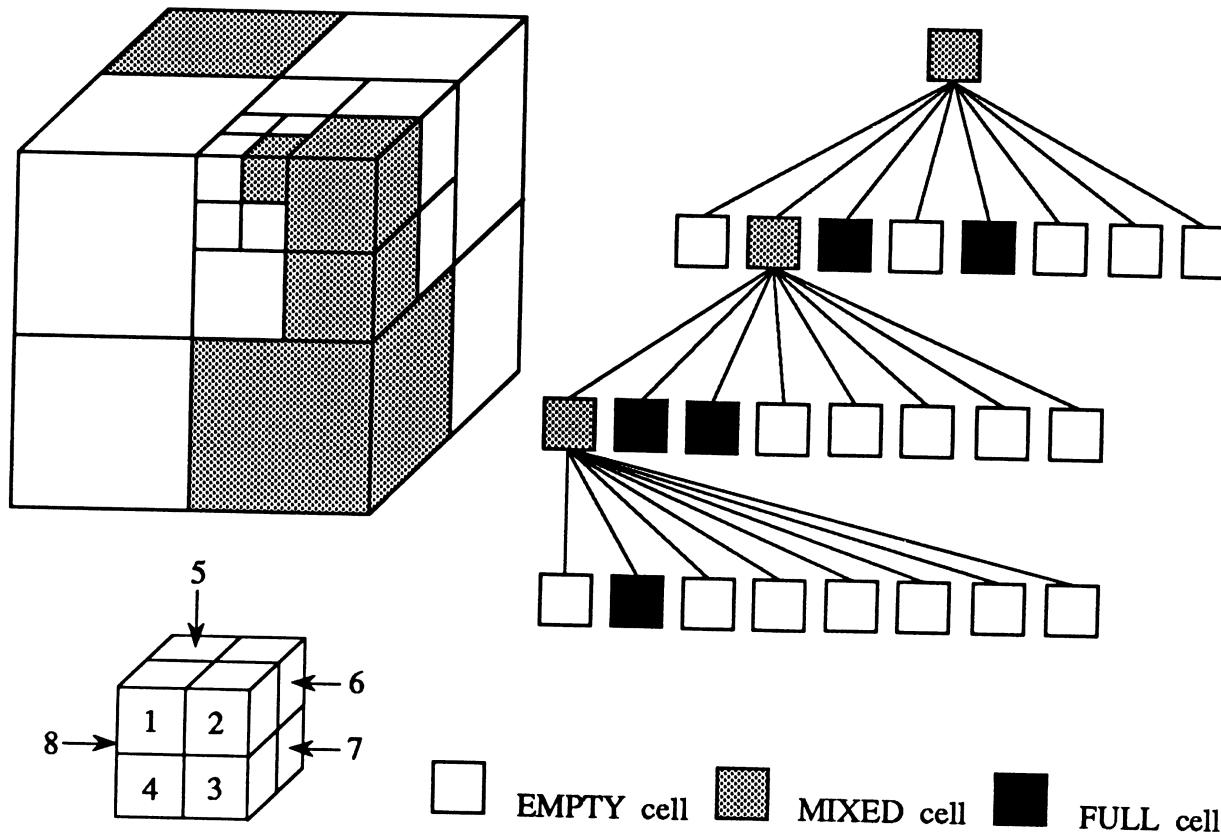
### Scomposizione approssimata in celle

F è rappresentata da una collezione di celle che non si sovrappongono, la cui unione è contenuta in F

Esempi: quadtree, octree,  $2^n$ -tree



## SCOMPOSIZIONE IN OCTREE



PROF. ROSARIO SORBELLO



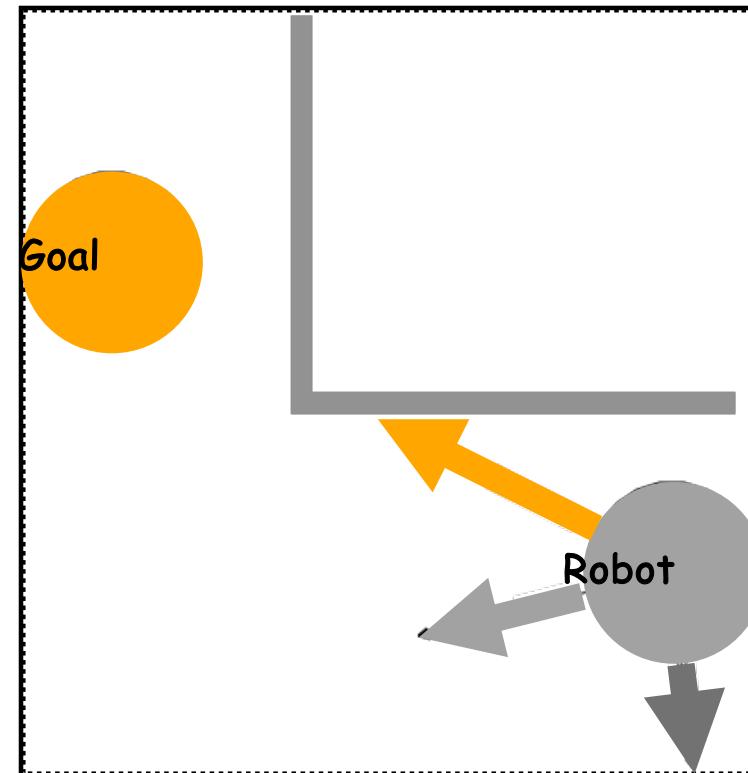
- Proposte inizialmente per evitare gli ostacoli in tempo reale (Schemi di Arkin)
- **Campo di potenziale:** Funzione scalare sullo spazio libero
- **Campo ideale (funzione di navigazione):** minimo globale continuo al goal, nessun minimo locale, infinito in prossimità di ostacoli
- Forza applicata al robot: gradiente negativo del potenziale. Il robot si muove sempre seguendo questa forza.



Khatib, 1986

$$F_{Goal} = -k_p(x - x_{Goal})$$

$$F_{Obstacle} = \begin{cases} \eta \left( \frac{1}{\rho} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2} \frac{\partial \rho}{\partial x} & \text{if } \rho \leq \rho_0, \\ 0 & \text{if } \rho > \rho_0 \end{cases}$$



PROF. ROSARIO SORBELLO



## Mappe

Rappresenta la connettività dello spazio libero mediante una rete di curve unidimensionali

## Scomposizione in celle

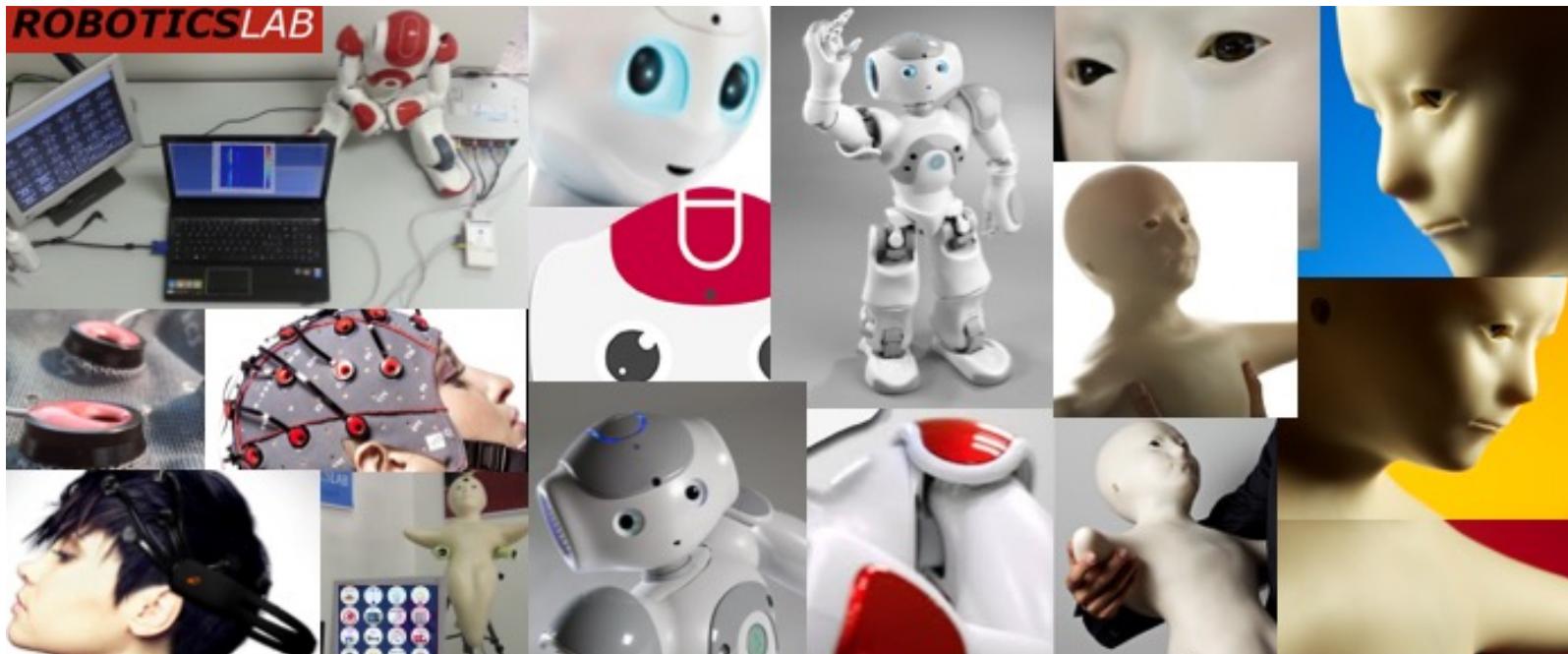
Scompone lo spazio libero in celle semplici e rappresenta lo spazio mediante il grafo di adiacenza delle celle

## Campi di potenziale

Definisce una funzione sullo spazio libero che ha un minimo globale nel gola e segue il gradiente



OSAKA UNIVERSITY



ATR

g-tec

## PIANIFICAZIONE 2° PARTE

PROF. ROSARIO SORBELLO

7° LEZIONE - PALERMO 28 OTTOBRE 2022



**ROBOTICS LAB**

# Pianificazione

---

Cercare un linguaggio:  
abbastanza espressivo per i problemi che si vogliono risolvere  
che ammetta un algoritmo ragionevolmente efficiente

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Pianificazione

---

PIANIFICAZIONE non è  
RISOLUZIONE DI PROBLEMI

- Rappresentazione di obiettivi
- Rappresentazione di stati
- Rappresentazione di azioni

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Ricerca di soluzioni di problemi

---

- *Rappr. azioni* attraverso descrizioni di stati successori
- *Rappr. stati* (iniziale,.., finale) completa e non ambigua
- *Rappr. obiettivi* attraverso il test obiettivo e le funzioni euristiche per il raggiungimento
- *Rappr. piani* come sequenza ininterrotta di azioni per raggiungere lo stato finale dallo stato iniziale; non sono previste interferenze da processi/agenti esterni (tempo fisso)



**ROBOTICS LAB**

# Agente pianificatore

---

- *Guarda al futuro per cercare azioni che contribuiscono al raggiungimento dell'obiettivo.*
- Un agente pianificatore genera un obiettivo da raggiungere, quindi formula dei piani di azione per raggiungerlo dallo stato corrente e poi li esegue. Una volta che ha un piano continua ad eseguirlo finché il piano è finito, poi ricomincia con un nuovo obiettivo.
- L'uso di esplicite rappresentazioni logiche permette al pianificatore di indirizzare le sue deliberazioni razionalmente.



**ROBOTICS** LAB

# Agente pianificatore

---

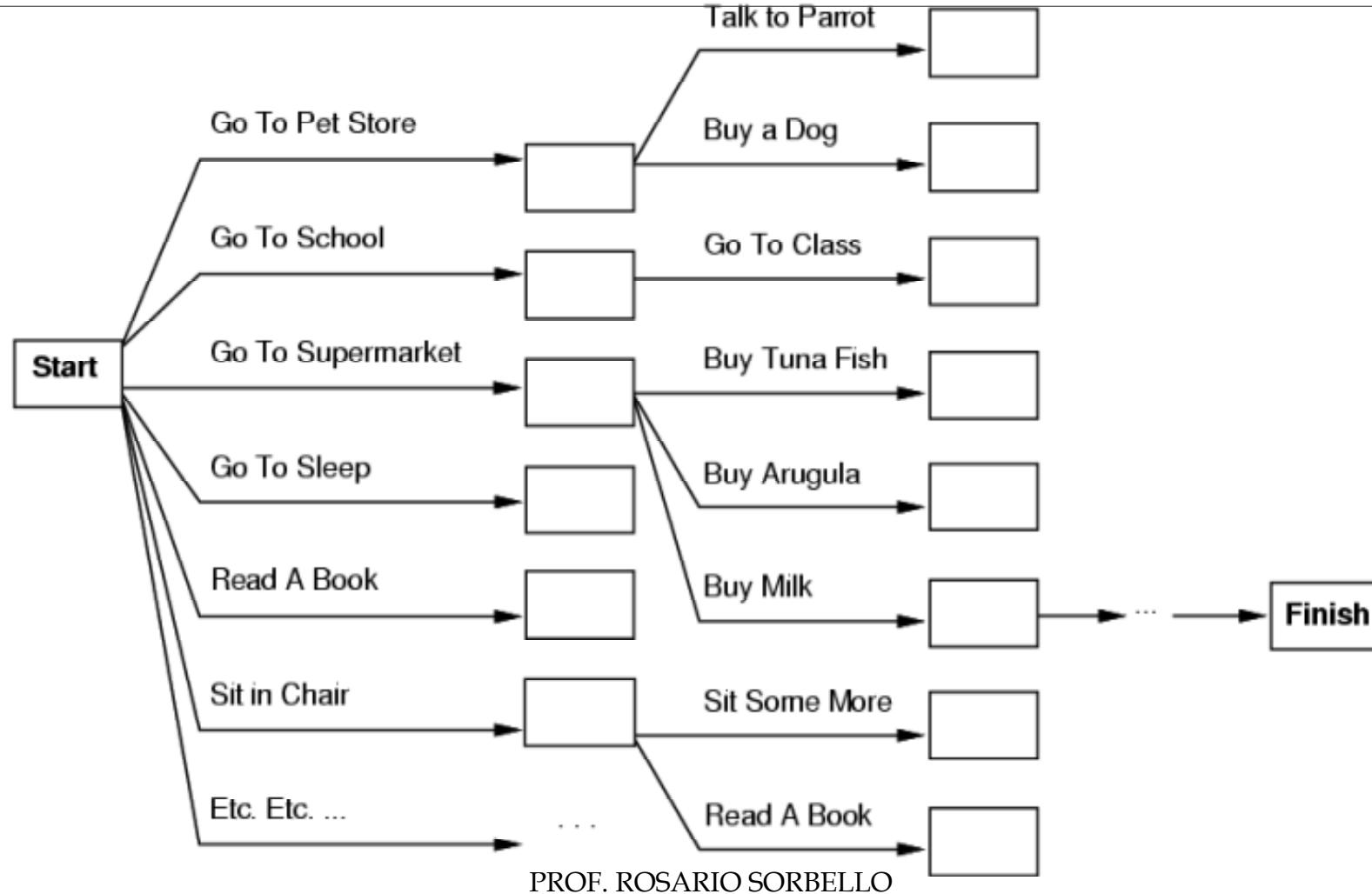
Quando lo stato del mondo è accessibile, un agente può usare le percezioni fornite dall'ambiente esterno per costruire un modello completo e corretto dello stato corrente del mondo. L'agente deve anche occuparsi del caso in cui l'obiettivo sia irrealizzabile, o sia già vero nello stato iniziale (piano completo vuoto).

L'agente interagisce con l'ambiente in modo minimo, utilizza le percezioni per definire lo stato iniziale (e l'obiettivo iniziale), quindi segue i passi del piano che ha costruito.

PROF. ROSARIO SORBELLO



# Problema della spesa





# Problema della spesa

---

- L'agente fa dei tentativi considerando le azioni possibili (cosa comprare), che sono classificate dalla funzione di valutazione (giusto/sbagliato). Finita una azione non sa cosa cercare successivamente e ricomincia il processo per tentativi.  
Avrebbe ***bisogno di strutturare le sue deliberazioni*** per lavorare allo stesso modo sia nello stato iniziale che in quello corrente, date le informazioni correnti.
- Un cammino attraverso questo spazio dallo stato iniziale a quello obiettivo costituisce un *piano* per il problema della spesa



# Pianificazione

---

1. *Il pianificatore deve essere in grado di fare connessioni dirette tra stati ed azioni.*

Gli algoritmi di pianificazione utilizzano descrizioni in linguaggi formali (logica del primo ordine), per cui gli stati e gli obiettivi sono rappresentati da insiemi di formule mentre le azioni sono rappresentate da *descrizioni logiche di precondizioni ed effetti* (le azioni modificano gli stati).



**ROBOTICS LAB**

# Pianificazione

---

2. *Il pianificatore può aggiungere azioni al piano ove sia necessario e non attraverso una sequenza incrementale che comincia dallo stato iniziale.*

Non è necessaria alcuna connessione tra l'ordine di pianificazione e l'ordine di esecuzione; si può *ridurre* così *il fattore di ramificazione* e la necessità di tornare indietro a causa di decisioni arbitrarie.



**ROBOTICS LAB**

# Pianificazione

---

*3. Molte parti del mondo sono indipendenti da molte altre parti del mondo.*

Adozione di obiettivi congiunti ed uso di *algoritmi di divide-et-impera* (efficienti perché portano a risolvere sottoproblemi di minore difficoltà e nel caso in cui l'effort per la ricomposizione dei risultati parziali non sia elevato).

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Pianificazione nel calcolo delle situaz.

---

- Il *calcolo delle situazioni* è una formalizzazione con il calcolo dei predicati degli *stati*, delle *azioni* e degli *effetti delle azioni sugli stati*.
- Si rappresenta la conoscenza sugli stati e sulle azioni con formule del calcolo dei predicati del primo ordine e poi si usa un sistema di deduzione per porre domande. La risposta alla domanda costituisce un piano per raggiungere lo stato desiderato.



**ROBOTICS LAB**

# Pianificazione nel calcolo delle situaz.

---

Un problema di pianificazione è rappresentato da formule logiche che descrivono le tre parti del problema:

- ***Stato iniziale*** (una formula logica arbitraria che descrive una situazione  $S^o$ )
- ***Stato obiettivo*** (un'interrogazione logica che cerca situazioni opportune  $S$ )
- ***Operatori*** (un insieme di descrizioni di azioni che trasformano gli stati  $S^o \rightarrow S$ )



# Pianificazione nel calcolo delle situaz.

---

Il calcolo delle situazioni si basa sulla considerazione che *le azioni trasformano gli stati*. E' importante gestire sequenze di azioni oltre ad azioni singole.

## **Sequenza di azioni** (definizione):

*una sequenza di azioni vuota non ha alcun effetto su una situazione, il risultato di una sequenza di azioni non vuota è comparabile all'applicazione della prima azione e poi all'applicazione delle azioni rimanenti nella situazione che si ottiene.*



**ROBOTICS LAB**

# Pianificazione nel calcolo delle situaz

---

*Una soluzione è un piano  $p$  che, quando applicato allo stato iniziale  $S^o$ , produce una situazione che soddisfa l'interrogazione dell'obiettivo.*

Con il linguaggio formale (logica del primo ordine) si esprimono obiettivi e piani e tramite il procedimento di inferenza si trovano i piani più opportuni.

Però il procedimento di inferenza non fornisce alcuna garanzia sul piano risultante  $p$  a parte il fatto che raggiunga l'obiettivo.

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Pianificazione nel calcolo delle situaz

---

Per rendere fattibile la pianificazione occorre:

- Restringere il linguaggio con cui si pongono i problemi per ridurre il numero di soluzioni possibili da cercare
- Usare un sistema di ragionamento specializzato (*pianificatore*)



## Rappresentazione di stati

Gli stati sono rappresentati da *congiunzioni di letterali ground privi di funzioni*, cioè predicati applicati a simboli costanti eventualmente negati.

Le formule del calcolo dei predicati che descrivono un insieme di stati del mondo vengono considerate esse stesse uno stato (=struttura dati che può essere modificata in modo corrispondente alle azioni dell'agente)



**ROBOTICS LAB**

# Linguaggio per pianificatori

---

## Rappresentazione di stati

Una descrizione di stato non deve essere necessariamente completa (es. agente in un ambiente inaccessibile) e corrisponde ad un insieme di stati completi possibili per i quali l'agente vorrebbe ottenere un piano di successo (=soddisfacente l'obiettivo)

PROF. ROSARIO SORBELLO



## Rappresentazione di obiettivi

Gli obiettivi sono rappresentati da *congiunzioni di letterali*. Gli obiettivi possono contenere anche variabili. Nel pianificatore si verifica l'esistenza di una sequenza di azioni che renda vero l'obiettivo.



**ROBOTICS LAB**

# Linguaggio per pianificatori

---

Data una *formula ben formata -obiettivo-*,

il metodo di ricerca trova una sequenza di azioni che produca uno stato del mondo descritto da una certa descrizione di stato  $S$  che soddisfi l'obiettivo

PROF. ROSARIO SORBELLO



ROBOTICS LAB

# Linguaggio per pianificatori

---

Le *azioni* per passare da uno stato ad un altro hanno *operatori* corrispondenti che:

modificano una *descrizione di stato prima dell'azione*  
in una *descrizione di stato dopo l'azione*

PROF. ROSARIO SORBELLO



## Rappresentazione per azioni

Gli **operatori STRIPS** consistono di:

**Descrizione dell'azione** (ciò che l'agente restituisce all'ambiente quando agisce)

**Precondizione** (*congiunzione di atomi –letterali positivi– che dice cosa deve essere vero prima che l'operatore possa essere applicato*)

**Effetto** (*congiunzione di letterali –positivi o negativi– che descrivono come cambia la situazione quando è applicato l'operatore*)



# Linguaggio per pianificatori (STRIPS)

*Notazione grafica per rappresentare azioni*

*At(here), Path(here, there)*

Go(there)

*At(there),  $\neg$  At(here)*

Non esiste alcuna variazione di situazioni esplicita

Ogni cosa nella precondizione si riferisce alla situazione immediatamente prima dell'azione, ogni cosa nell'effetto si riferisce alla situazione che è il risultato dell'azione



**ROBOTICS LAB**

# Linguaggio per pianificatori (STRIPS)

---

Un operatore con variabili si dice

***schema di operatore o regola STRIP***

(non corrisponde ad una singola azione esegibile, ma piuttosto ad una famiglia di azioni, una per ciascuna differente istanziazione delle variabili= particolarizzazioni di base delle regole che corrispondono agli operatori reali).

Possono essere eseguiti solo operatori istanziati



ROBOTICS LAB

# Linguaggio per pianificatori (STRIPS)

---

Un operatore  $op$  è *applicabile* in uno stato  $S$  se c'è qualche modo per istanziare le variabili di  $op$  in modo che ognuna delle precondizioni di  $op$  sia vera in  $S$

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Pianificazione di progressione

---

La *pianificazione di progressione* ricerca in uno spazio di descrizioni di stato nella direzione in avanti, dallo stato iniziale a quello obiettivo, applicando ripetutamente particolarizzazioni di regole di STRIPS

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Pianificazione di regressione

---

La *pianificazione di regressione* ricerca all'indietro riducendo il fattore di ramificazione (sottoobiettivi)

E' più efficiente della pianificazione di progressione

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Pianificatore

---

Potremmo risolvere un problema dato (es. problema della spesa) cominciando dallo stato iniziale e applicando gli operatori uno alla volta finché si raggiunge uno stato che includa tutti i letterali dell'obiettivo.

Un tale algoritmo risolutore è *pianificatore nello spazio delle situazioni* in quanto effettua una ricerca nello spazio delle situazioni possibili, e *pianificatore di progressione* in quanto cerca in avanti dalla situazione iniziale a quella obiettivo.

Problema legato all'*alto fattore di ramificazione*



**ROBOTICS LAB**

# Pianificatore

---

I nodi dell'albero di ricerca di un pianificatore in uno spazio di situazioni corrispondono a delle situazioni e il cammino attraverso l'albero di ricerca è il piano che viene restituito alla fine dal pianificatore.

Ciascun punto di ramificazione aggiunge un altro passo all'inizio (regressione) o alla fine (progressione) del piano.

PROF. ROSARIO SORBELLO



# Piani condizionali

---

- Si può inserire un condizionale per l'esecuzione nel piano in corrispondenza del punto di applicazione dell'operatore
- Nel costruire il piano successivo a un condizionale, il processo di pianificazione si divide in tanti rami quanti sono i disgiunti che potrebbero soddisfare le precondizioni di un operatore (**contesti**)



ROBOTICS LAB

# Pianificatore: ricerca nello spazio dei piani

---

In domini complessi non è possibile effettuare una ricerca nel piano delle situazioni.

Occorre cercare attraverso lo ***spazio dei piani*** piuttosto che attraverso lo spazio delle situazioni.

Partendo da un piano iniziale parziale lo si espande fino a trovare un piano completo che risolva il problema.

PROF. ROSARIO SORBELLO



ROBOTICS LAB

# Pianificatore: ricerca nello spazio dei piani

Gli *operatori per la ricerca nello spazio dei piani* trasformano i piani in altri piani mediante:

- **Aggiunta** di nuovi passi al piano
- **Riordino** di passi già presenti nel piano
- **Trasformazione di un piano** parzialmente ordinato in uno completamente ordinato
- **Trasformazione dello schema di piano** con variabili in una particolarizzazione (istanziazione di una variabile precedentemente non legata)

La soluzione è il piano finale; il cammino seguito per raggiungerla è irrilevante.

PROF. ROSARIO SORBELLO



- **AZIONE:** BUY(X)
- **PRECONDIZIONI :** AT(P), SELLS(P,X)
- **EFFETTI:** HAVE(X)

PRECONDITIONS

*At(p) Sells(p,x)*

COMMAND

**Buy(x)**

ADD-LIST

*Have(x)*



ROBOTICS LAB

## PIANIFICAZIONE DELLO SPAZIO DEGLI STATI

---

GLI ALGORITMI DI PIANIFICAZIONE DELLO SPAZIO DEGLI STATI (SPAZIO DELLE SITUAZIONI) O **STATE-SPACE (SITUATION SPACE) PLANNING ALGORITHMS** CERCANO NELLO SPAZIO DEI POSSIBILI STATI DEL MONDO UN PERCORSO CHE RISOLVA IL PROBLEMA.

1. POSSONO ESSERE BASATI SULLA **PROGRESSIONE**: UNA RICERCA IN AVANTI DALLO STATO INIZIALE ALLA RICERCA DELLO STATO OBIETTIVO.
2. POSSONO ESSERE BASATI SULLA **REGRESSIONE**: UNA RICERCA A RITROSO DAGLI OBIETTIVI VERSO LO STATO INIZIALE.

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## PIANIFICAZIONE DELLO SPAZIO DEGLI STATI

---

-STRIPS È UN ALGORITMO BASATO SULLA REGRESSIONE INCOMPLETA.

-I PIANIFICATORI DELLO SPAZIO DEI PIANI CERCANO NELLO SPAZIO DEI PIANI PARZIALI, CHE SONO INSIEMI DI AZIONI CHE POSSONO NON ESSERE TOTALMENTE ORDINATI.

-I PIANIFICATORI DI ORDINE PARZIALE SONO BASATI SUL PIANO E INTRODUCONO VINCOLI DI ORDINAMENTO SOLO SE NECESSARI (MINIMO IMPEGNO), PER EVITARE DI CERCARE INUTILMENTE NELLO SPAZIO DEGLI ORDINAMENTI POSSIBILI.

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## PIANIFICAZIONE DOPO STRIPS

---

- RICERCA NELLO SPAZIO DEI PIANI
- PIANI PARZIALI
- GOAL FINALE E SUB GOALS
- PIANI PARZIALMENTE ORDINATI
- VINCOLI SEQUENZIALI TRA AZIONI

PROF. ROSARIO SORBELLO



## Partially Ordered Plan

- Plan
  - Steps
  - Ordering constraints
  - Variable binding constraints
  - Causal links

Ci sono i passi, o le azioni che intendiamo compiere. Ci sono vincoli di ordine che dicono quali passi devono essere prima di quali altri. Devono essere coerenti, ma non devono specificare un ordine totale dei passi. Ci sono vincoli di vincolo delle variabili che specificano quali variabili sono uguali a quali altre variabili o costanti. E ci sono collegamenti causali che ci ricordano quali effetti delle azioni stiamo usando per soddisfare quali precondizioni di altre azioni.

**STEPS:** AZIONE CHE PIANIFICHIAMO DI ATTUARE

**ORDERING CONSTRAINTS:** SPECIFICANO QUALI STEPS DEVONO AVVENIRE PRIMA DI ALTRI

**VARIABLE BINDING CONSTRAINTS:** SPECIFICA QUALI VARIABILI SONO UGUALI AD ALTRE VARIABILI O COSTANTI

**CAUSAL LINKS:** CI RICORDANO QUALI EFFETTI DELLE AZIONI STIAMO USANDO PER SODDISFARE QUALI PRECONDIZIONI DI ALTRE AZIONI

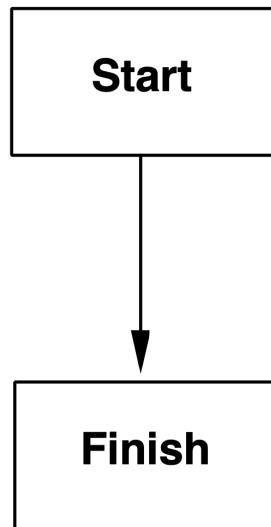
Let's just briefly review the parts of a partially ordered plan. There are the steps, or actions we plan to take. There are ordering constraints that say which steps have to be before which other ones. They need to be consistent, but they don't need to specify a total order on the steps. There are variable binding constraints that specify which variables are equal to which other variables or constants. And there are causal links to remind us which effects of actions we are using to satisfy which preconditions of other actions.

PROF. ROSARIO SORBELLO



## PIANO PARZIALMENTE ORDINATO

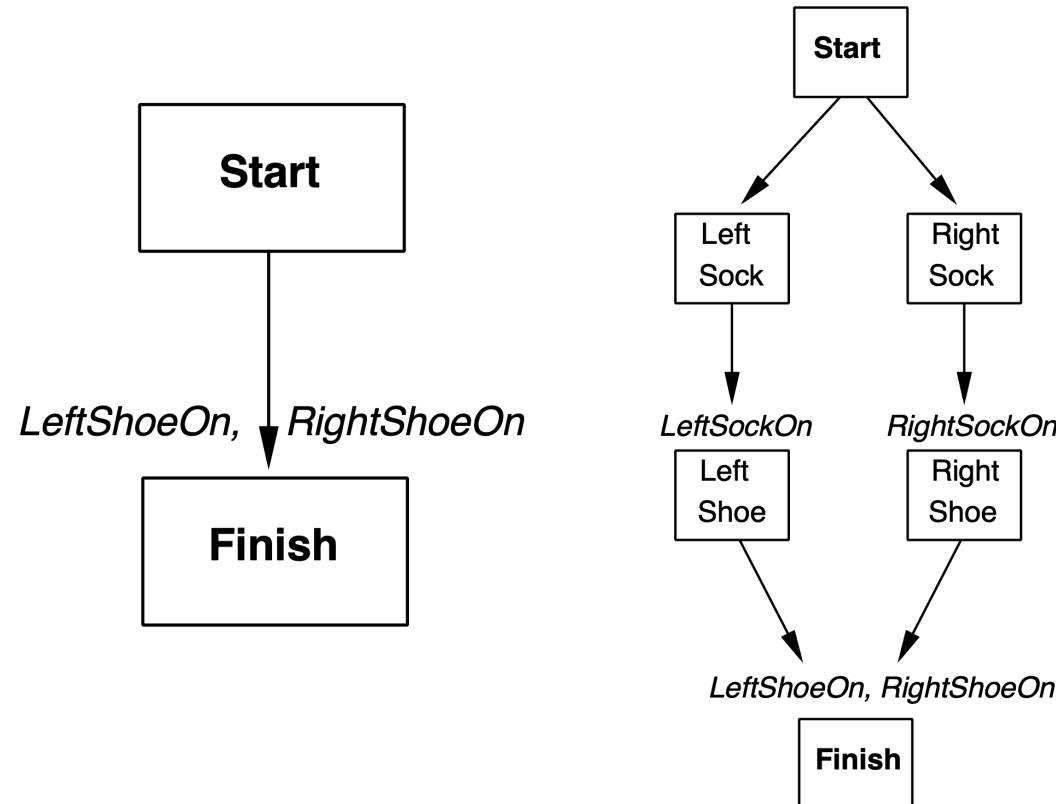
---



PER PRIMA COSA, CREIAMO IL PIANO INIZIALE. RICORDIAMO CHE CONTIENE DUE PASSI "FITTIZI", CHIAMATI INIZIO E FINE. IL PASSO DI INIZIO HA COME EFFETTI LE CONDIZIONI INIZIALI E IL PASSO DI FINE HA COME PRECONDIZIONI LE CONDIZIONI DELL'OBIETTIVO.



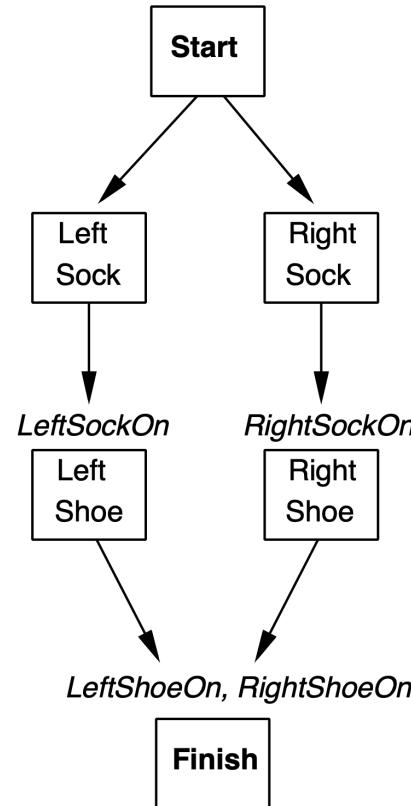
## ORDINAMENTO PARZIALE



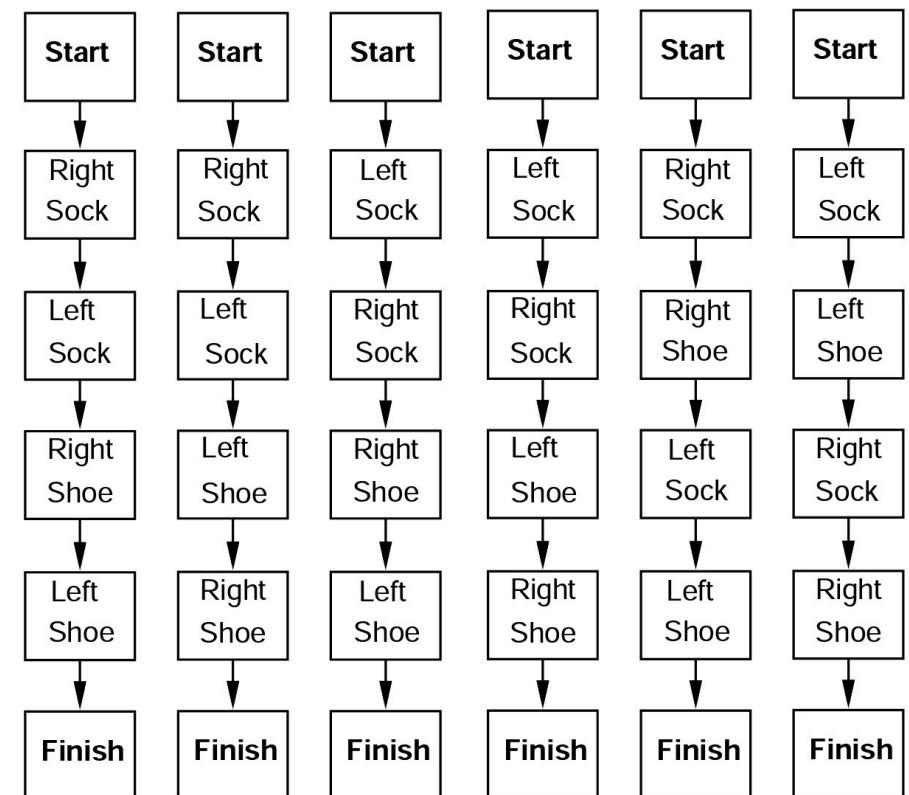
PROF. ROSARIO SORBELLO

# ORDINAMENTO PARZIALE e ORDINAMENTO TOTALE

Partial Order Plan



Total Order Plans:

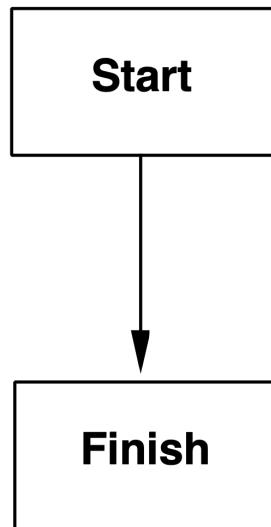


PROF. ROSARIO SORBELLO



## PIANO PARZIALMENTE ORDINATO

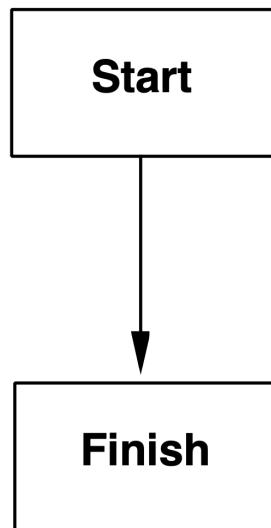
---



-QUINDI, SI ITERA FINO A QUANDO IL PIANO NON È COMPLETO. RICORDIAMO CHE UN PIANO È COMPLETO QUANDO OGNI PRECONDIZIONE DI OGNI PASSO È SODDISFATTA DALL'EFFETTO DI QUALCHE ALTRO PASSO; QUESTE SONO LE RELAZIONI DI CUI TENIAMO TRACCIA USANDO I COLLEGAMENTI CAUSALI. QUINDI, FINCHÉ IL PIANO NON È COMPLETO, CONTINUIAMO A ESEGUIRE I SEGUENTI TRE PASSAGGI



## PIANO PARZIALMENTE ORDINATO



PER PRIMA COSA, SCEGLIAMO UN SOTTO-OBIETTIVO (**SUB-GOALS**), CIOÈ UNA PRECONDIZIONE NON ANCORA SODDISFATTA. CHIAMEREMO QUESTA PRECONDIZIONE **C**.

SECONDA COSA, SCEGLIAMO UN OPERATORE CHE POSSA SODDISFARE **C** PER NOI.

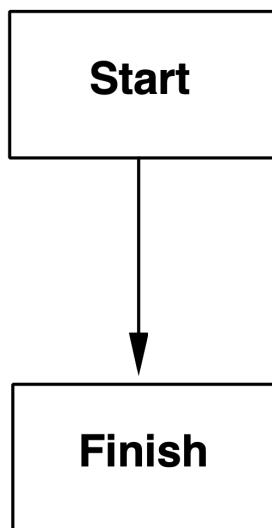
TERZA COSA, VERIFICHIAMO CHE IL LAVORO SVOLTO NELLA FASE PRECEDENTE NON ABbia CAUSATO LA "MINACCIA", O LA POSSIBILE VIOLAZIONE, DI UNO DEI NOSTRI PRECEDENTI LEGAMI CAUSALI. SE UN LEGAME È MINACCIATO, DOBBIAMO RISOLVERLO.

PROF. ROSARIO SORBELLO



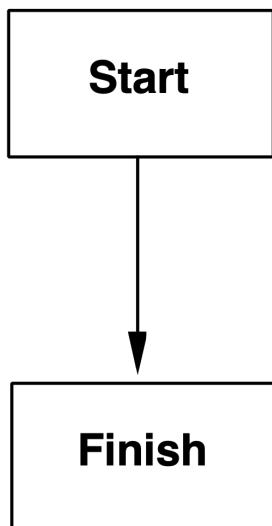
A plan in POP (whether it be a finished one or an unfinished one) comprises:

- A set of *plan steps*. Each of these is a STRIPS operator, but with the variables instantiated.
- A set of *ordering constraints*:  $S_i \prec S_j$  means step  $S_i$  must occur sometime before  $S_j$  (not necessarily immediately before).
- A set of *causal links*:  $S_i \xrightarrow{c} S_j$  means step  $S_i$  achieves precondition  $c$  of step  $S_j$ .



**START** È UN PASSO SENZA PRECONDIZIONI, SOLO EFFETTI: GLI EFFETTI SONO LO STATO INIZIALE DEL MONDO.

**FINISH** È UN PASSO SENZA EFFETTI, SOLO PRECONDIZIONI: LE PRECONDIZIONI SONO L'OBIETTIVO.



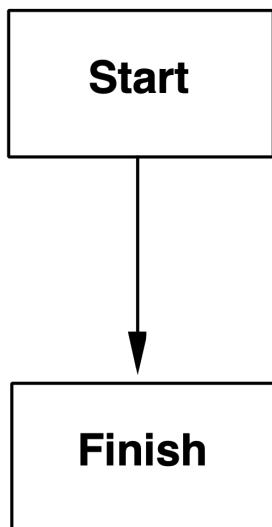
This initial plan is refined using POP's *plan refinement operators*. As we apply them, they will take us from an unfinished plan to a less and less unfinished plan, and ultimately to a solution plan. There are four operators, falling into two groups:

- *Goal achievement operators*
  - *Step addition*: Add a new step  $S_i$  which has an effect  $c$  that can achieve an as yet unachieved precondition of an existing step  $S_j$ . Also add the following constraints:  $S_i \prec S_j$  and  $S_i \xrightarrow{c} S_j$  and  $\text{Start} \prec S_i \prec \text{Finish}$ .
  - Use an effect  $c$  of an existing step  $S_i$  to achieve an as yet unachieved precondition of another existing step  $S_j$ . And add just two constraints:  $S_i \prec S_j$  and  $S_i \xrightarrow{c} S_j$ .
- Causal links must be *protected* from *threats*, i.e. steps that delete (or negate or *clobber*) the protected condition. If  $S$  threatens link  $S_i \xrightarrow{c} S_j$ :
  - **Promote**: add the constraint  $S \prec S_i$ ; **or**
  - **Demote**: add the constraint  $S_j \prec S$

**Aggiunta di un passo:** Aggiungere un nuovo passo  $S_i$  che ha un effetto  $c$  che può realizzare una precondizione non ancora raggiunta di un passo esistente  $S_j$ . Aggiunge anche i seguenti vincoli:  $S_i \prec S_j$  e  $S_i c \rightarrow S_j$  e  $\text{Start} \prec S_i \prec \text{Finish}$ .

Utilizzare un effetto  $c$  di un passo esistente  $S_i$  per raggiungere una precondizione non ancora raggiunta di un altro passo esistente  $S_j$ . E aggiungere solo due vincoli:  $S_i \prec S_j$  e  $S_i c \rightarrow S_j$ .

- I legami causali devono essere protetti da minacce, cioè da passi che cancellano (o negano o bloccano) **la condizione protetta**. Se  $S$  minaccia il legame  $S_i c \rightarrow S_j$
- **Promuovere**: aggiungere il vincolo  $S \prec S_i$ ;
- oppure- **Retrocedere**: aggiungere il vincolo  $S_j \prec S$



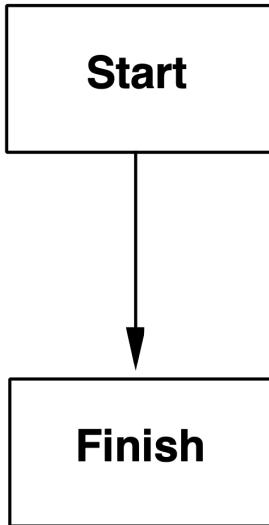
This initial plan is refined using POP's *plan refinement operators*. As we apply them, they will take us from an unfinished plan to a less and less unfinished plan, and ultimately to a solution plan. There are four operators, falling into two groups:

- *Goal achievement operators*
  - *Step addition*: Add a new step  $S_i$  which has an effect  $c$  that can achieve an as yet unachieved precondition of an existing step  $S_j$ . Also add the following constraints:  $S_i \prec S_j$  and  $S_i \xrightarrow{c} S_j$  and  $\text{Start} \prec S_i \prec \text{Finish}$ .
  - Use an effect  $c$  of an existing step  $S_i$  to achieve an as yet unachieved precondition of another existing step  $S_j$ . And add just two constraints:  $S_i \prec S_j$  and  $S_i \xrightarrow{c} S_j$ .
- Causal links must be *protected* from *threats*, i.e. steps that delete (or negate or *clobber*) the protected condition. If  $S$  threatens link  $S_i \xrightarrow{c} S_j$ :
  - **Promote**: add the constraint  $S \prec S_i$ ; **or**
  - **Demote**: add the constraint  $S_j \prec S$

**Aggiunta di un passo:** Aggiungere un nuovo passo  $S_i$  che ha un effetto  $c$  che può realizzare una precondizione non ancora raggiunta di un passo esistente  $S_j$ . Aggiunge anche i seguenti vincoli:  $S_i \prec S_j$  e  $S_i c \rightarrow S_j$  e  $\text{Start} \prec S_i \prec \text{Finish}$ .

Utilizzare un effetto  $c$  di un passo esistente  $S_i$  per raggiungere una precondizione non ancora raggiunta di un altro passo esistente  $S_j$ . E aggiungere solo due vincoli:  $S_i \prec S_j$  e  $S_i c \rightarrow S_j$ .

- I legami causali devono essere protetti da minacce, cioè da passi che cancellano (o negano o bloccano) **la condizione protetta**. Se  $S$  minaccia il legame  $S_i c \rightarrow S_j$
- **Promuovere**: aggiungere il vincolo  $S \prec S_i$ ;
- oppure- **Retrocedere**: aggiungere il vincolo  $S_j \prec S$



The goal achievement operators ought to be obvious enough. They find preconditions of steps in the unfinished plan that are not yet achieved. The two goal achievement operators remedy this either by adding a new step whose effect achieves the precondition, or by exploiting one of the effects of a step that is already in the plan.

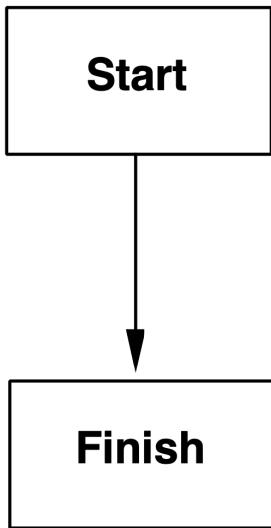
The promotion and demotion operators may be less clear. Why are these needed? POP uses problem-decomposition: faced with a conjunctive precondition, it uses goal achievement on each conjunct separately. But, as we know, this brings the risk that the steps we add when achieving one part of a precondition might interfere with the achievement of another precondition. And the idea of promotion and demotion is to add ordering constraints so that the step cannot interfere with the achievement of the precondition.

Finally, we have to be able to recognise when we have reached a *solution plan*: a finished plan.

A solution plan is one in which:

- every precondition of every step is achieved by the effect of some other step and all possible clobberers have been suitably demoted or promoted; and
- there are no contradictions in the ordering constraints, e.g. disallowed is  $S_i \prec S_j$  and  $S_j \prec S_i$ ; also disallowed is  $S_i \prec S_j$ ,  $S_j \prec S_k$  and  $S_k \prec S_i$ .

Gli operatori per il raggiungimento degli obiettivi dovrebbero essere abbastanza ovvi. Trovano le precondizioni dei passi del piano incompiuto che non sono ancora state raggiunte. I due operatori di raggiungimento dell'obiettivo vi pongono rimedio aggiungendo un nuovo passo il cui effetto realizza la precondizione, oppure sfruttando uno degli effetti di un passo già presente nel piano. Gli operatori di promozione e retrocessione possono essere meno chiari. Perché sono necessari? POP utilizza la scomposizione del problema: di fronte a una precondizione congiuntiva, utilizza il raggiungimento dell'obiettivo su ogni congiunzione separatamente. Ma, come sappiamo, questo comporta il rischio che i passi aggiunti quando si raggiunge una parte della precondizione possano interferire con il raggiungimento di un'altra precondizione. L'idea di promozione e retrocessione consiste nell'aggiungere vincoli di



GLI OPERATORI PER IL RAGGIUNGIMENTO DEGLI OBIETTIVI DOVREBBERO ESSERE ABBASTANZA OVVI. TROVANO LE PRECONDIZIONI DEI PASSI DEL PIANO INCOMPIUTO CHE NON SONO ANCORA STATE RAGGIUNTE. I DUE OPERATORI DI RAGGIUNGIMENTO DELL'OBIETTIVO VI PONGONO RIMEDIO AGGIUNGENDO UN NUOVO PASSO IL CUI EFFETTO REALIZZA LA PRECONDIZIONE, OPPURE SFRUTTANDO UNO DEGLI EFFETTI DI UN PASSO GIÀ PRESENTE NEL PIANO. GLI OPERATORI DI PROMOZIONE E RETROCESSIONE POSSONO ESSERE MENO CHIARI. PERCHÉ SONO NECESSARI? POP UTILIZZA LA SCOMPOSIZIONE DEL PROBLEMA: DI FRONTE A UNA PRECONDIZIONE CONGIUNTIVA, UTILIZZA IL RAGGIUNGIMENTO DELL'OBIETTIVO SU OGNI CONGIUNZIONE SEPARATAMENTE. MA, COME SAPPIAMO, QUESTO COMPORTA IL RISCHIO CHE I PASSI AGGIUNTI QUANDO SI RAGGIUNGE UNA PARTE DELLA PRECONDIZIONE POSSANO INTERFERIRE CON IL RAGGIUNGIMENTO DI UN'ALTRA PRECONDIZIONE. L'IDEA DI PROMOZIONE E RETROCESSIONE CONSISTE NELL'AGGIUNGERE VINCOLI DI ORDINE IN MODO CHE IL PASSO NON POSSA INTERFERIRE CON IL RAGGIUNGIMENTO DELLA PRECONDIZIONE. INFINE, DOBBIAMO ESSERE IN GRADO DI RICONOSCERE QUANDO ABBIAMO RAGGIUNTO UN PIANO DI SOLUZIONE: UN PIANO FINITO. UN PIANO DI SOLUZIONE È UN PIANO IN CUI: - OGNI PRECONDIZIONE DI OGNI PASSO È RAGGIUNTA PER EFFETTO DI QUALCHE ALTRO PASSO E TUTTI I POSSIBILI CLOBBER SONO STATI ADEGUATAMENTE RETROCESSI O PROMOSSI; E - NON CI SONO CONTRADDIZIONI NEI VINCOLI DI ORDINAMENTO, AD ESEMPIO NON SONO AMMESSI  $SI \prec SJ$  E  $SJ \prec SI$ ; NON SONO AMMESSI NEPPURE  $SI \prec SJ$ ,  $SJ \prec SK$  E  $SK \prec SI$ .



## Shopping Domain

- Actions
  - Buy(x, store)
    - Pre: At(store), Sells(store, x)
    - Eff: Have(x)
  - Go(x, y)
    - Pre: At(x)
    - Eff: At(y),  $\neg$ At(x)
- Goal
  - Have(Milk)  $\wedge$  Have(Banana)  $\wedge$  Have(Drill)
- Start
  - At(Home)  $\wedge$  Sells(SM, Milk)  $\wedge$  Sells(SM, Banana)  $\wedge$  Sells(HW, Drill)

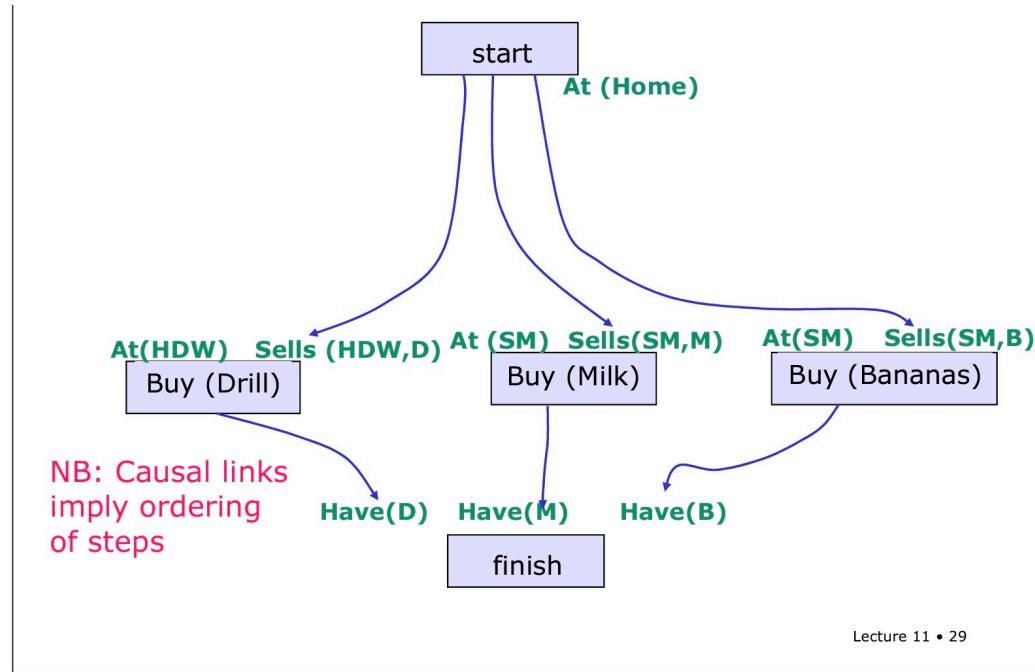


## PIANO PARZIALMENTE ORDINATO (ESEMPIO)

---

VEDIAMO ORA COSA SUCCIDE QUANDO ESEGUIAMO QUESTO ALGORITMO NEL DOMINIO DELLO SHOPPING. USEREMO IL BLU PER I LEGAMI CAUSALI E ASSUMEREMO CHE I LEGAMI CAUSALI IMPLICHINO VINCOLI D'ORDINE, IN MODO DA NON DOVER DISEGNARE TROPPE FRECCE EXTRA (SI NOTI, TUTTAVIA, CHE POTREMMO FINIRE PER AGGIUNGERE VINCOLI D'ORDINE CHE NON SONO ASSOCIATI AI LEGAMI CAUSALI. LI METTEREMO IN ROSSO). INOLTRE, PER EVITARE CHE LA FIGURA DIVENTI INCREDIBILMENTE INGOMBRA, A VOLTE SCRIVEREMO IL NOME DI UNA CONDIZIONE SOLO UNA VOLTA, QUANDO C'È UN LEGAME CAUSALE. TECNICAMENTE, DOVREMBO SCRIVERLA IN FONDO AL PASSO CHE LA RENDE VERA E SCRIVERLA IN CIMA AL PASSO CHE NE HA BISOGNO. MA NON SEMPRE C'È SPAZIO PER QUESTO.

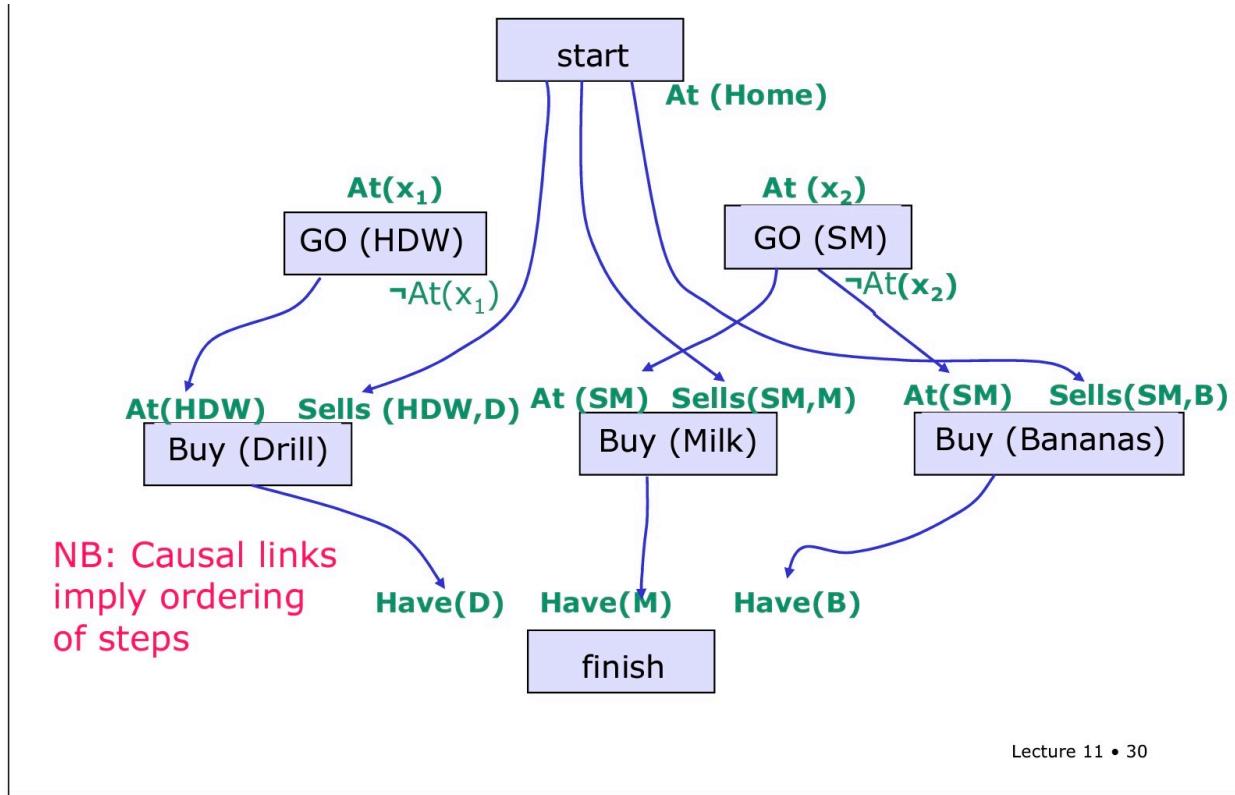
## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



PER NON PERDERE TEMPO, PARTIAMO DAL PRESUPPOSTO CHE IL PROCESSO DI PIANIFICAZIONE SIA GIÀ ABBASTANZA AVANZATO. ABBIAMO LE FASI DI INIZIO E FINE. E ABBIAMO DECISO DI FARE IL PASSO "COMPRARE IL TRAPANO", CHE CI PERMETTERÀ DI AVERE (TRAPANO). COMPRANDO IL LATTE OTTERREMO "AVERE" (LATTE). COMPRARE(BANANE) OTTERRÀ AVERE(BANANE). VENDERE (FERRAMENTA, TRAPANO), VENDERE (SUPERMERCATO, LATTE) E VENDERE (SUPERMERCATO, BANANE) ERANO TUTTI VERI NELLO STATO DI PARTENZA. E START PRODUCE ANCHE A CASA.

PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)

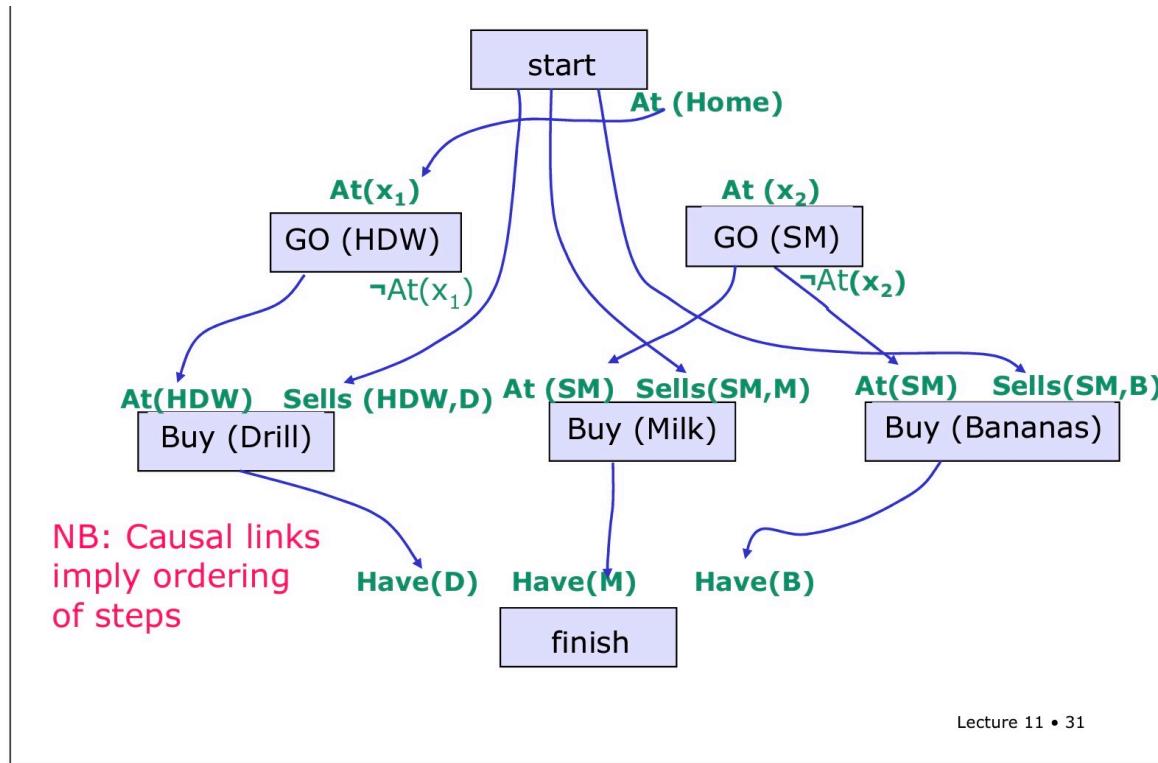


Lecture 11 • 30

USEREMO GO (FERRAMENTA) PER RAGGIUNGERE AT(FERRAMENTA). E USEREMO GO (SUPERMERCATO) PER RAGGIUNGERE AT (SUPERMERCATO). QUESTO PIANO È COMPLETO?

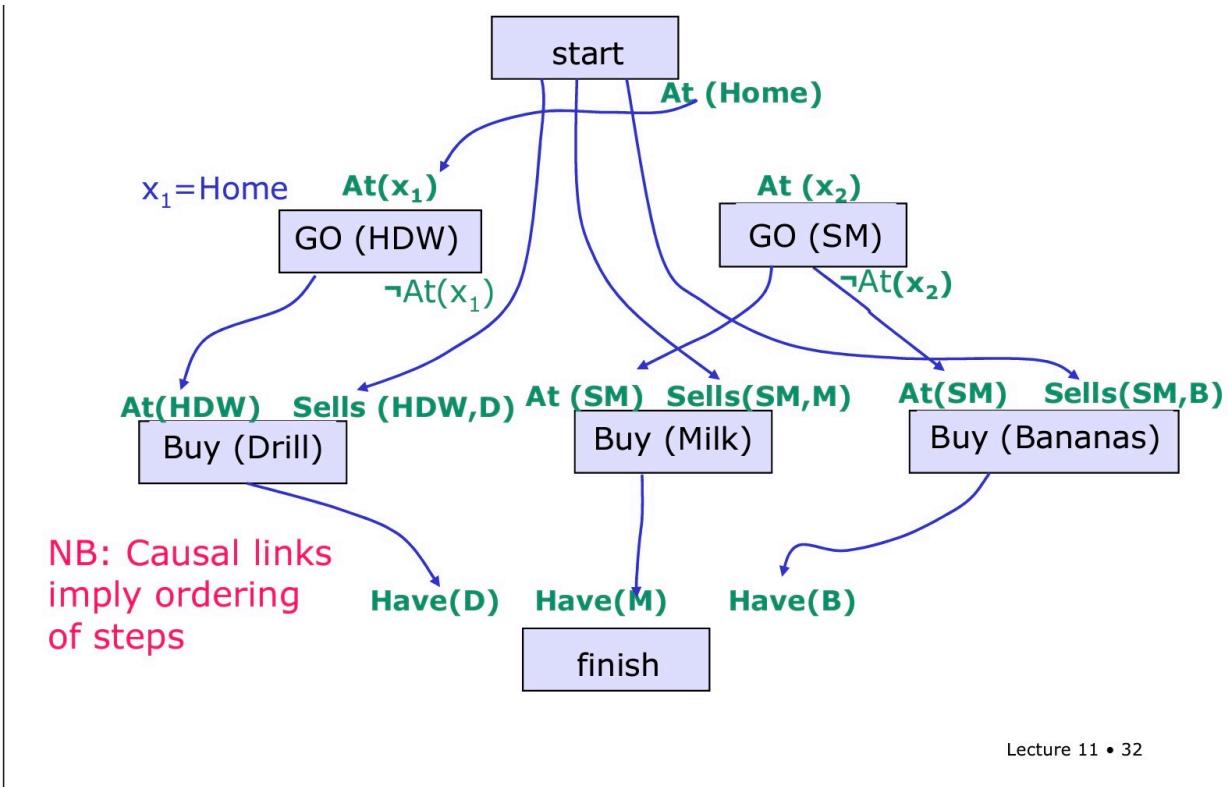
PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



NO, NON È ANCORA COMPLETO, PERCHÉ NON ABBIAMO SODDISFATTO LE PRECONDIZIONI AT DEI PASSI GO. COMINCIAMO QUINDI AD AFFRONTARE LA PRECONDIZIONE AT( $x_1$ ) DEL PASSO GO-TO HARDWARE-STORE. POSSIAMO SODDISFARLA UTILIZZANDO L'EFFETTO AT(HOME) DEL PASSO START.

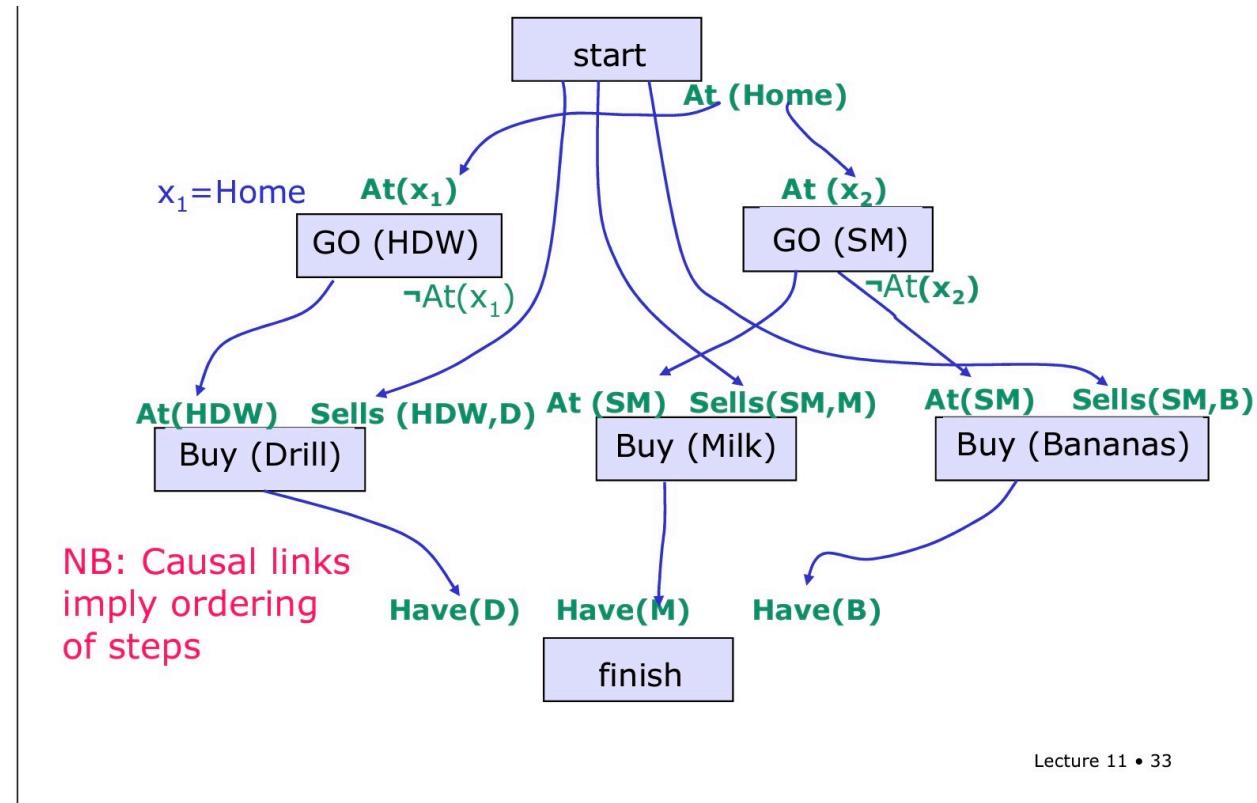
## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



DOBBIAMO ANCHE AGGIUNGERE IL VINCOLO DELLA VARIABILE X1 UGUALE A HOME.

PROF. ROSARIO SORBELLO

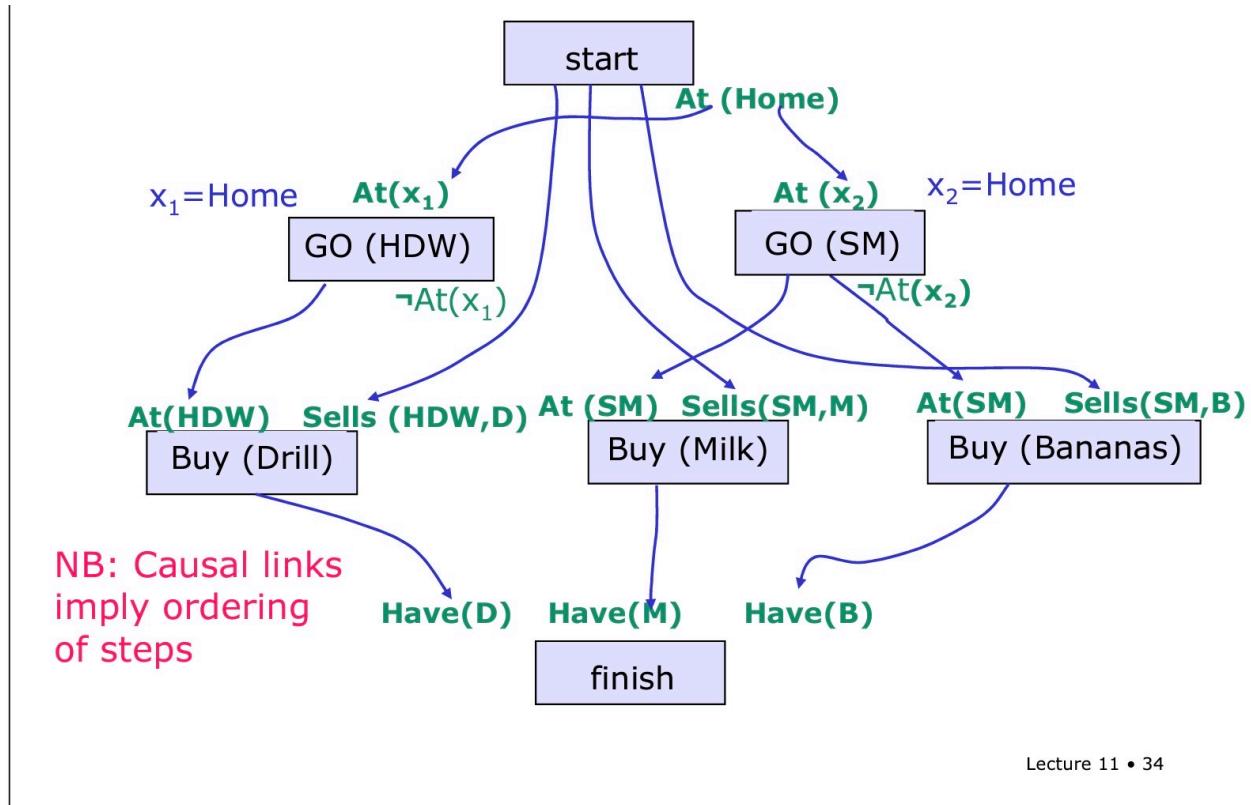
## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



ORA POSSIAMO LAVORARE SULLA PRECONDIZIONE  $\text{AT}(x_2)$  DEL PASSO GO-TO-SUPERMARKET. SEMBRA CHE SIA FACILE SODDISFARLA ANCHE CON L'EFFETTO  $\text{AT}(\text{HOME})$  DEL PASSO DI PARTENZA.

PROF. ROSARIO SORBELLO

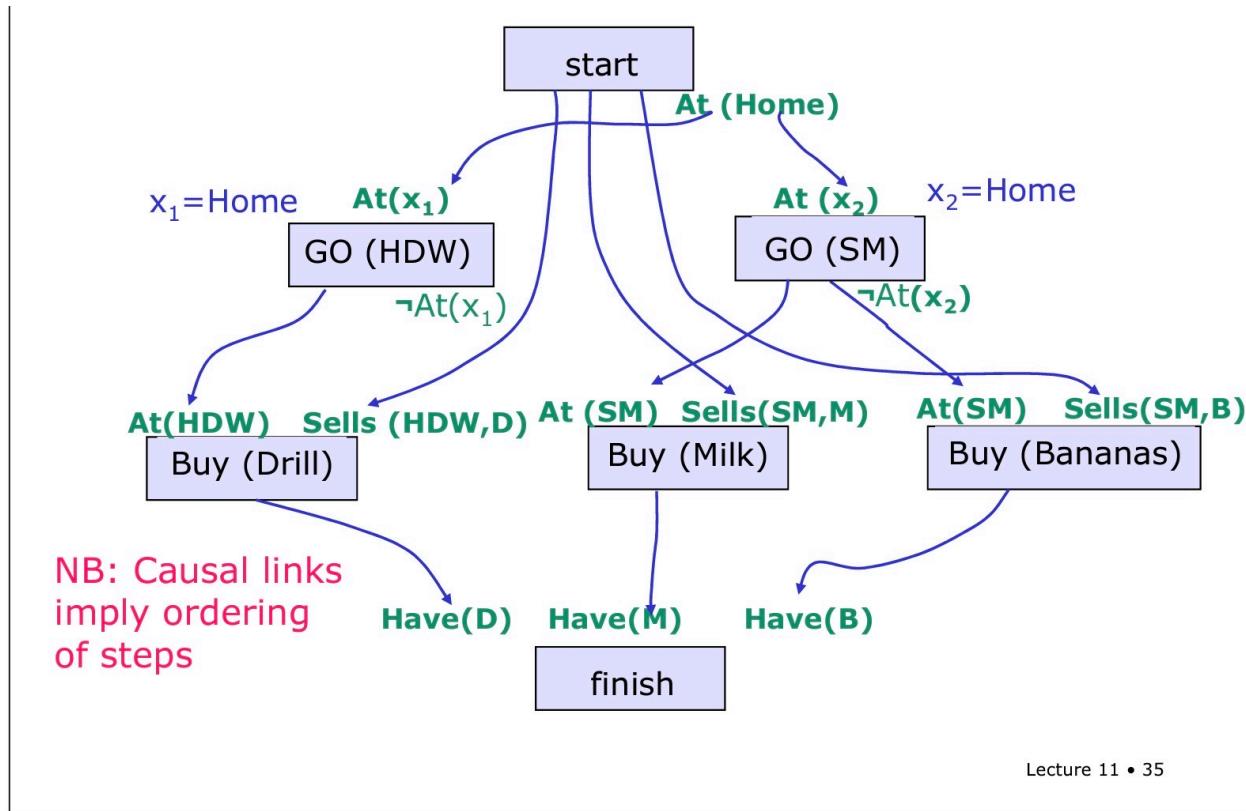
## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



PER FARE CIÒ, DOVREMBO AGGIUNGERE UN ALTRO VINCOLO VARIABILE  $X_2$  UGUALE A HOME. QUESTO È COERENTE CON GLI ALTRI VINCOLI, QUINDI SEMBRA TUTTO A POSTO.

PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)

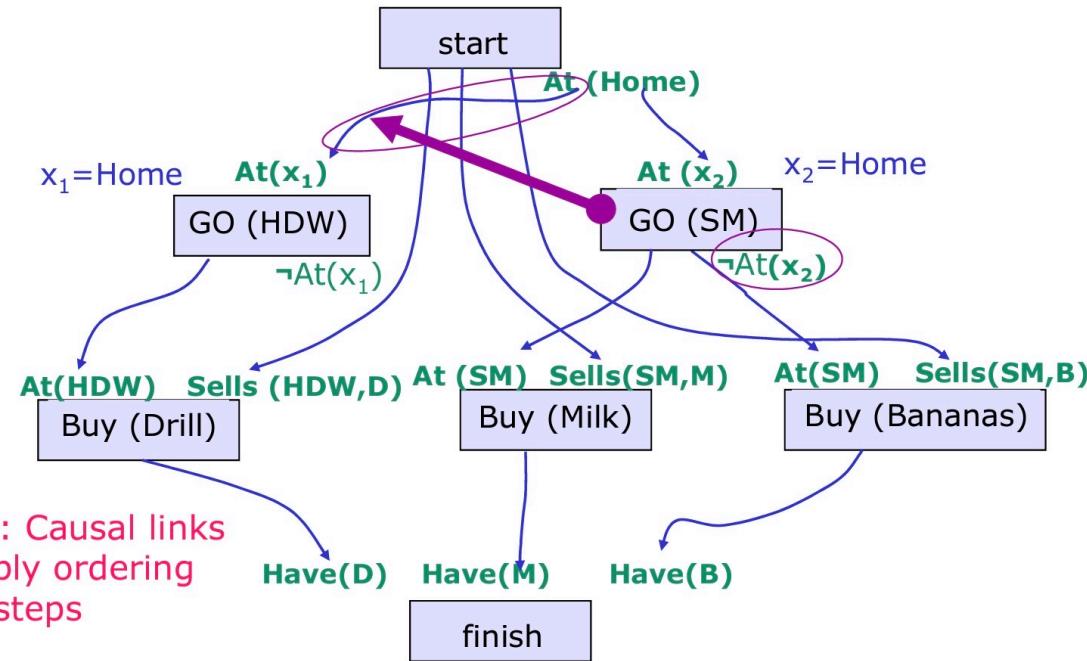


Lecture 11 • 35

VA BENE. ORA TUTTE LE NOSTRE PRECONDIZIONI SONO SODDISFATTE, MA POTREBBERO ESSERCI ANCORA DEI PROBLEMI. ABBIAMO UNA SERIE DI MINACCE DA RISOLVERE.

PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)

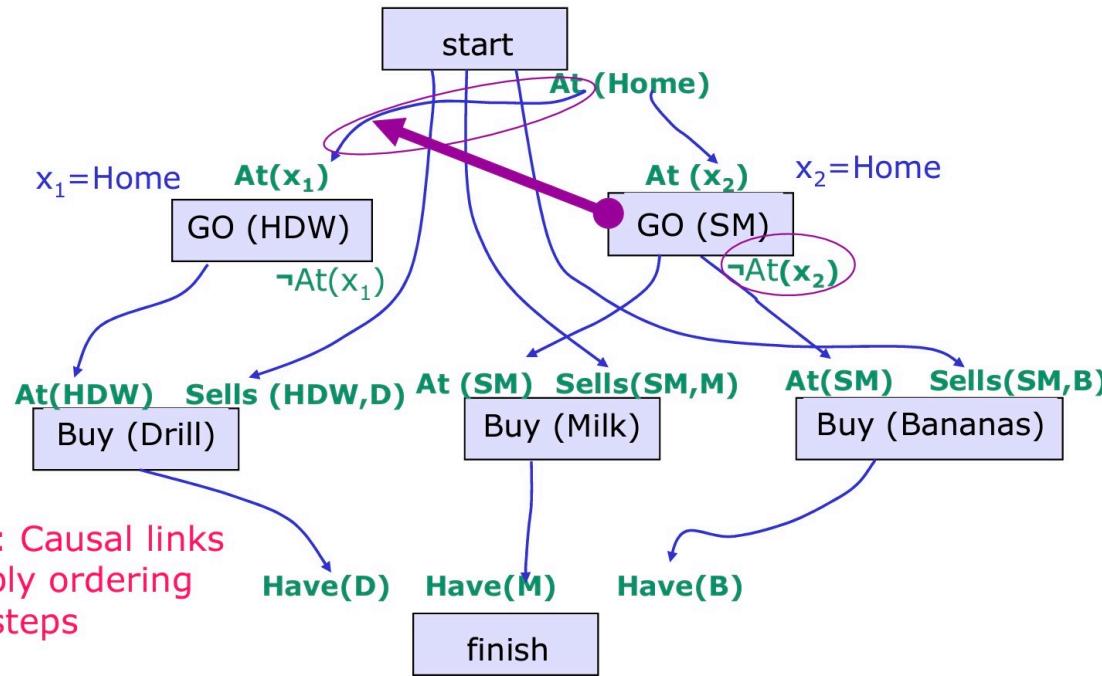


Lecture 11 • 36

OSSERVIAMO IL COLLEGAMENTO DA  $\text{AT}(\text{HOME})$  A  $\text{AT}(x_1)$ . QUESTO È ORA MINACCIATO DAL PASSO  $\text{GO}(\text{SM})$ . POSSIAMO VEDERE LA MINACCIA NOTANDO CHE  $\text{GO}(\text{SM})$  NON HA  $\text{AT}(\text{HOME})$  COME EFFETTO (PERCHÉ  $x_1$  È LEGATO A HOME) E CHE NON CI SONO VINCOLI DI ORDINE CHE IMPEDISCANO CHE AVVENGA DOPO L'INIZIO E PRIMA DI  $\text{GO}(\text{HDW})$ . MOSTREREMO LE MINACCE CON GRANDI FRECCE VIOLA.

PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



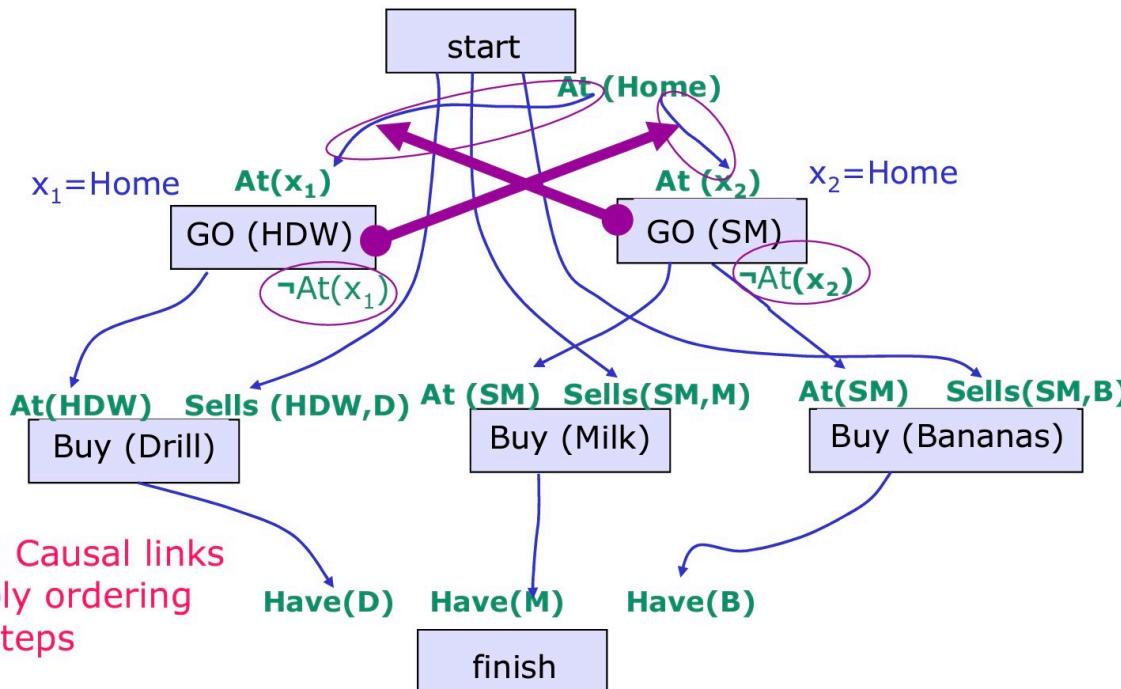
Lecture 11 • 37

QUINDI, COME POSSIAMO RISOLVERE QUESTA MINACCIA? NON POSSIAMO METTERE NULLA PRIMA DEL PASSO DI INIZIO (CI SONO VINCOLI TEMPORALI IMPLICITI CHE RICHIEDONO CHE TUTTO AVVENGA DOPO L'INIZIO E PRIMA DELLA FINE). QUINDI, FORSE POTREMMO RICHIEDERE CHE  $\text{GO}(\text{SM})$  AVVENGA DOPO  $\text{GO}(\text{HDW})$ .

PROF. ROSARIO SORBELLO



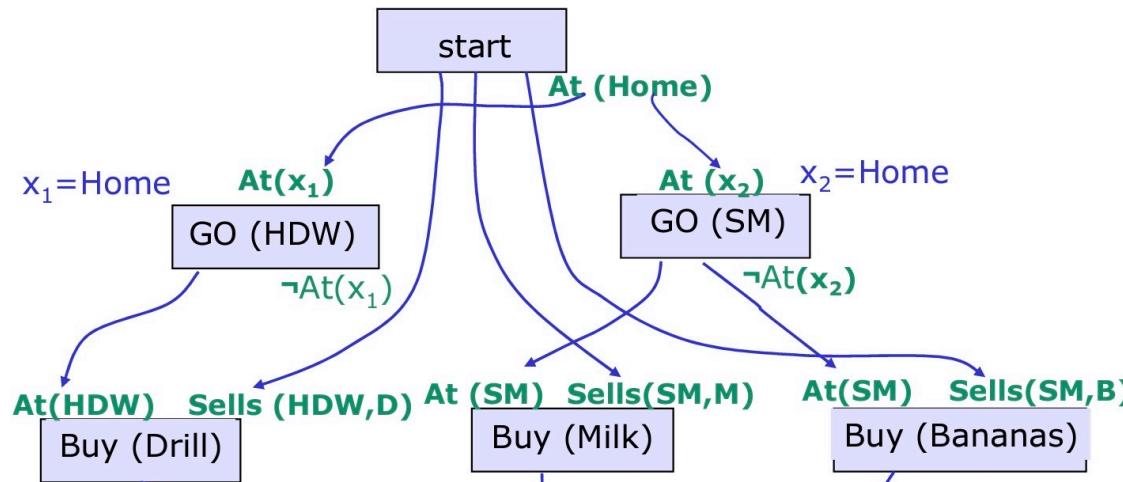
RO



Lecture 11 • 38

But we still have a problem. Ma abbiamo ancora un problema di simmetria. Guardate il collegamento da  $\text{At}(\text{Home})$  a  $\text{At}(x_2)$ . È minacciato da  $\text{Go}(\text{HDW})$ . L'unico modo per risolvere il problema è far sì che  $\text{Go}(\text{HDW})$  avvenga dopo  $\text{GO}(\text{SM})$ .

PROF. ROSARIO SORBELLO

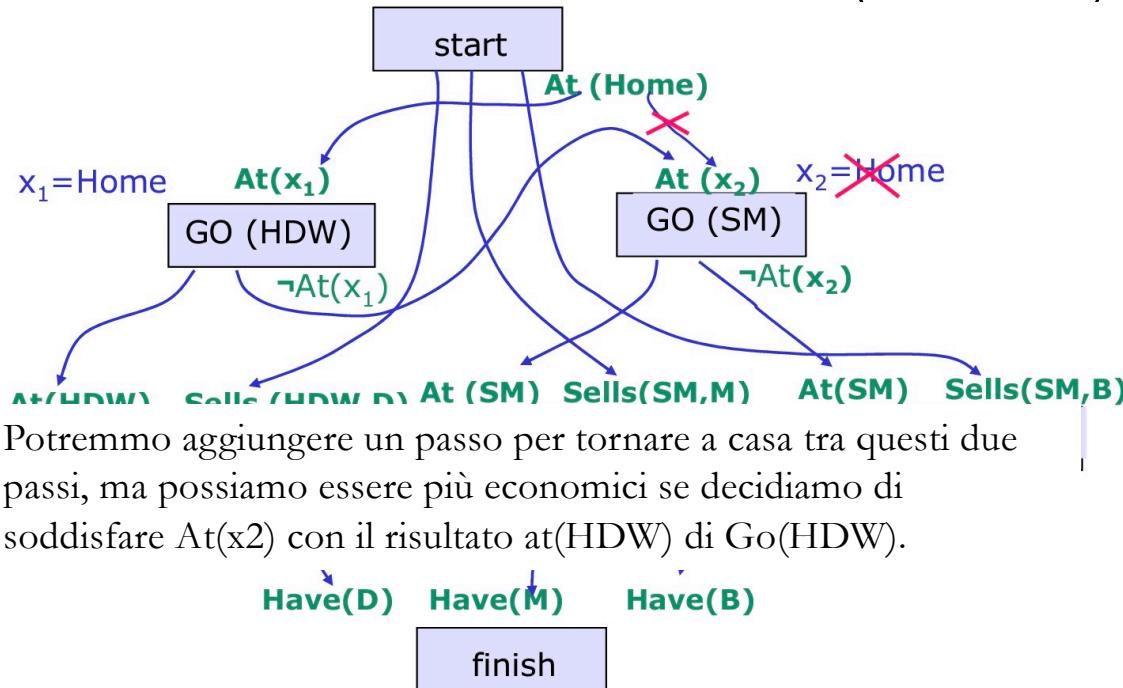


Essendo esseri umani intelligenti, possiamo fare un passo indietro e vedere che non possiamo andare da casa al negozio di ferramenta e al supermercato, senza tornare a casa nel mezzo. Quindi, dobbiamo fare qualcos'altro per far funzionare questo piano.

Lecture 11 • 39

Being clever humans, we can step back for a minute and see that we can't go from home to the hardware store and to the supermarket, without going back home in between. So, we need to do something else to make this plan work.

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



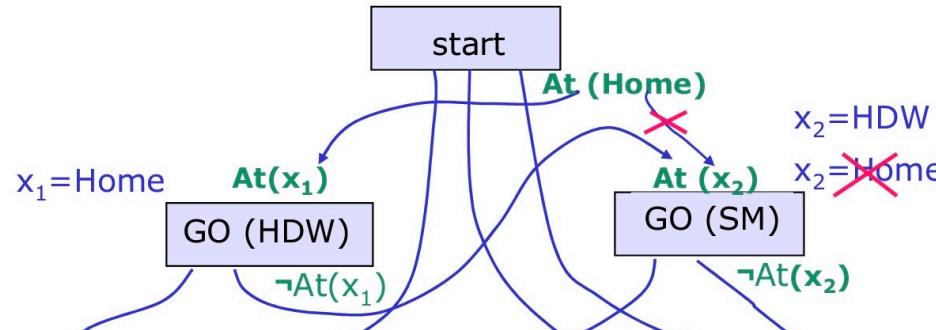
Potremmo aggiungere un passo per tornare a casa tra questi due passi, ma possiamo essere più economici se decidiamo di soddisfare  $\text{At}(x_2)$  con il risultato  $\text{at}(\text{HDW})$  di  $\text{Go}(\text{HDW})$ .

Lecture 11 • 40

We could add a step to go home in between those two steps, but we can be more economical if we instead decide to satisfy  $\text{At}(x_2)$  with the  $\text{at}(\text{HDW})$  result of  $\text{Go}(\text{HDW})$ .

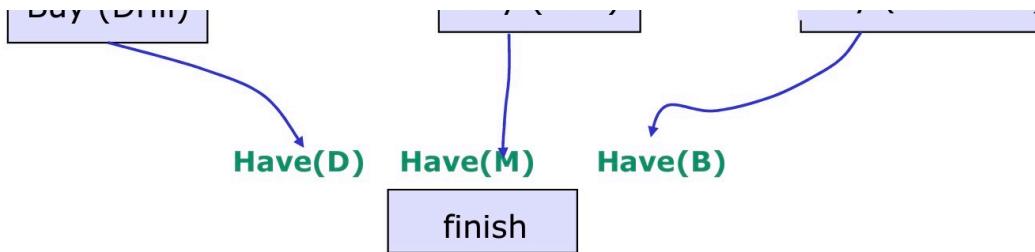
PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



Ora vincoliamo la variabile  $x_2$  a essere HDW piuttosto che home.

**A** E questa parte del piano sembra essere ragionevolmente a posto.

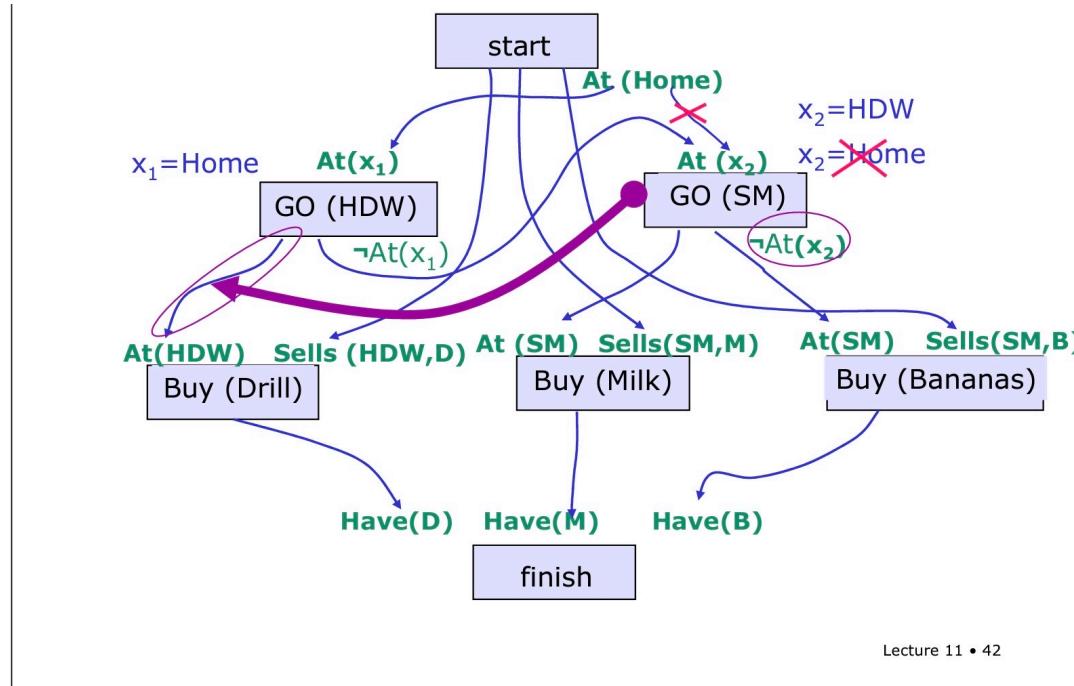


Lecture 11 • 41

Now we constrain the variable  $x_2$  to be HDW rather than home. And this part of the plan seems to be in reasonably good shape.

PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)

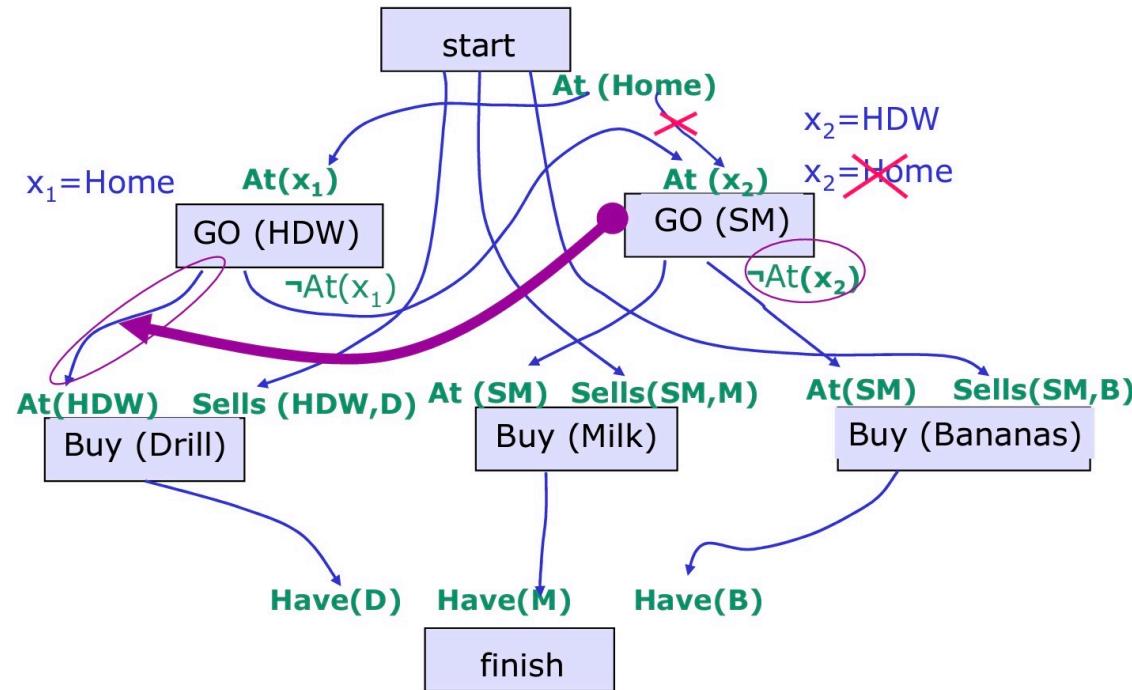


Lecture 11 • 42

MA NON ABBIAMO ANCORA FINITO! ABBIAMO ANCORA ALCUNE MINACCIE DA AFFRONTARE. ATTUALMENTE NON C'È NULLA CHE CI IMPEDISCA DI ANDARE DIRETTAMENTE AL SUPERMERCATO DOPO ESSERE STATI IN FERRAMENTA, SENZA COMPRARE IL NOSTRO TRAPANO. PIÙ FORMALMENTE, IL PASSO GO(SM) MINACCIA IL LEGAME CAUSALE CHE STABILISCE AT(HDW), TRA GO(HDW) E BUY(DRILL). QUESTO PERCHÉ GO(SM) NON HA AT(HDW) COME EFFETTO E PERCHÉ NON CI SONO VINCOLI DI ORDINE CHE GLI IMPEDISCANO DI ARRIVARE TRA GO(HDW) E BUY(DRILL).

PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)

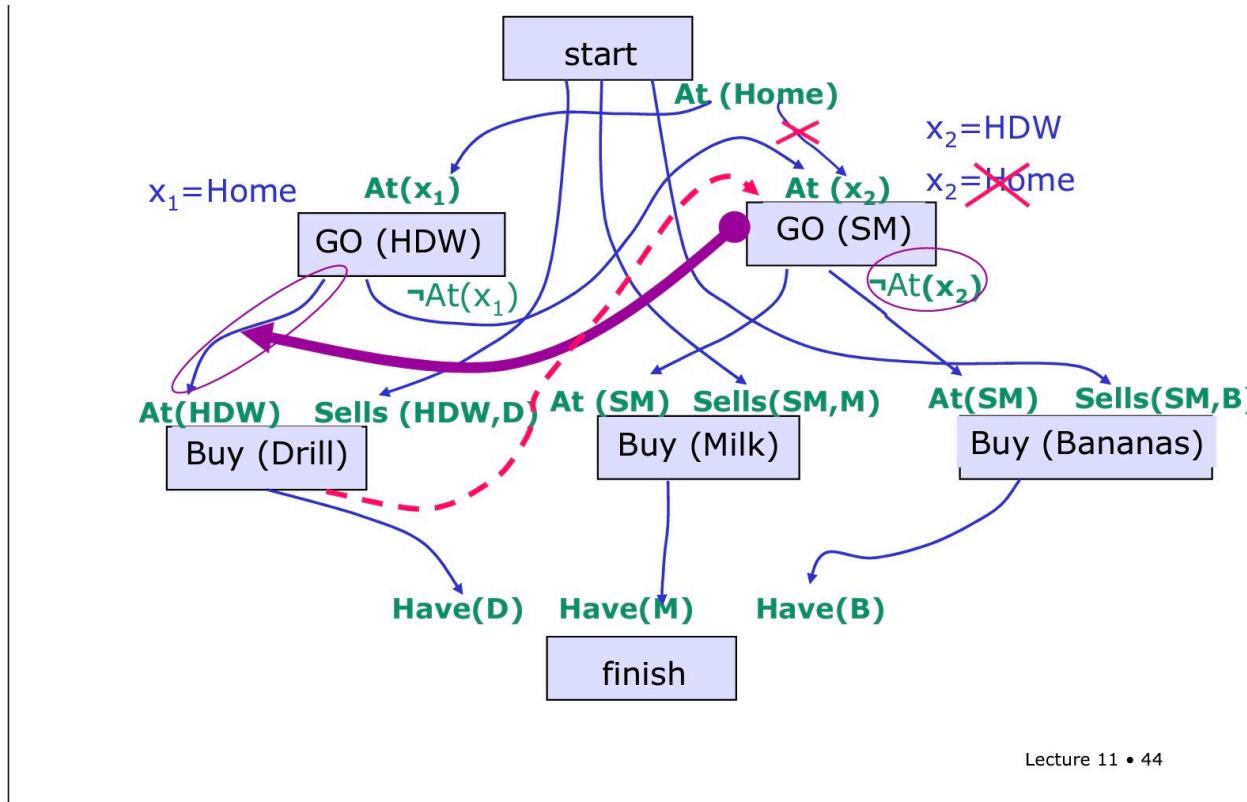


Lecture 11 • 43

POTREMMO CERCARE DI RISOLVERE LA MINACCIA METTENDO GO(SM) PRIMA DI GO(HDW), MA IN QUESTO CASO AVREMMO UN INSIEME DI VINCOLI TEMPORALI INCOERENTI (PERCHÉ C'È GIÀ UN LEGAME CAUSALE TRA GO(HDW) E GO(SM)).

PROF. ROSARIO SORBELLO

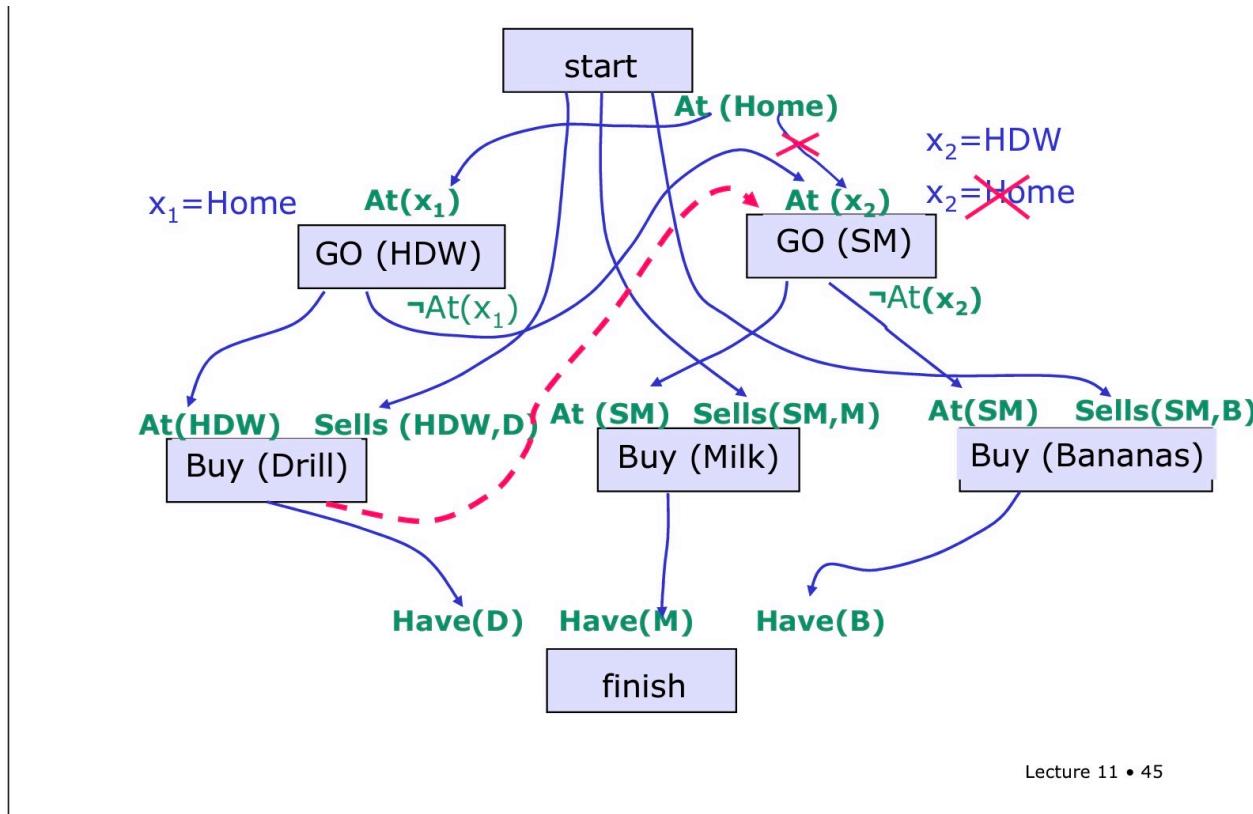
## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



PERTANTO, DOVREMO IMPORRE CHE GO(SM) AVVENGA DOPO BUY(DRILL). AGGIUNGEREMO UN VINCOLO TEMPORALE (LA LINEA ROSSA TRATTEGGIATA) PER ASSICURARCI CHE CIÒ AVVENGA.

PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)

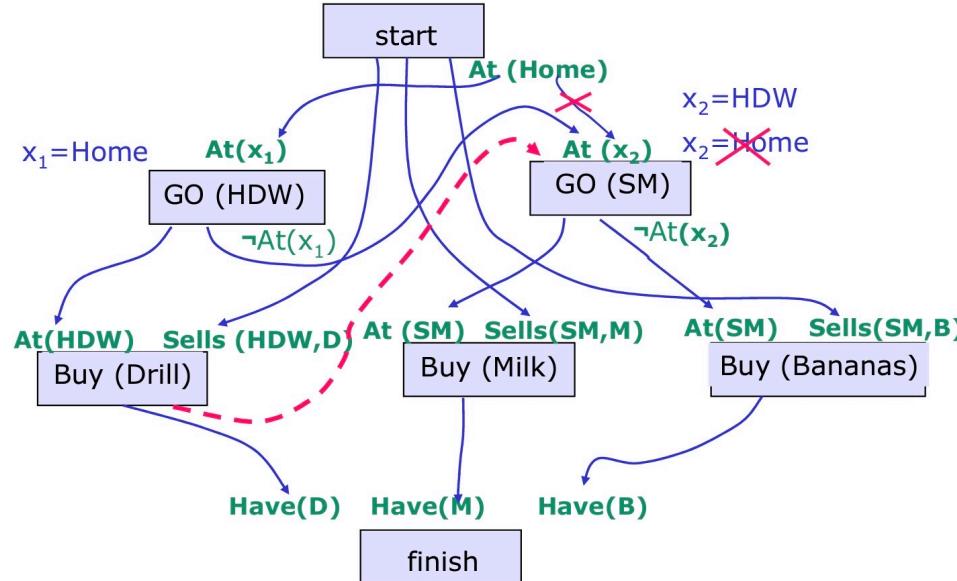


Lecture 11 • 45

QUINDI, ORA ABBIAMO UN PIANO. E CREDO CHE SIA CORRETTO. L'ESEMPIO DEL LIBRO ERA PIÙ COMPLICATO PERCHÉ BISOGNAVA ESSERE "A CASA" ALLA FINE, MA È PIÙ O MENO LA STESSA COSA, QUINDI NON LO FARÒ.

PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)

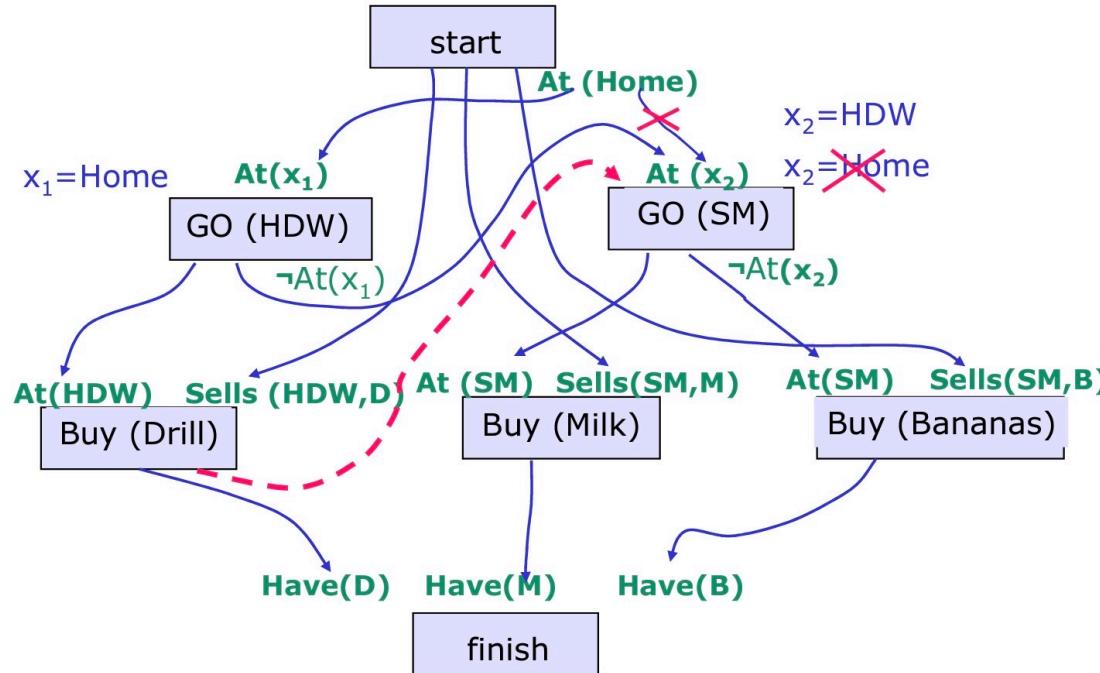


Lecture 11 • 46

ANALIZZIAMO UN PO' QUESTO PIANO. IL PIANO PREVEDE CHE SI VADA PRIMA DAL FERRAMENTA E POI SI COMPRI IL TRAPANO. DOPODICHÉ, DOBBIAMO ANDARE AL SUPERMERCATO. UNA VOLTA ARRIVATI AL SUPERMERCATO, PERÒ, POSSIAMO COMPRARE LATTE E BANANE IN ENTRAMBI GLI ORDINI. QUINDI, ABBIAMO FINITO CON UN PIANO PARZIALMENTE ORDINATO. MA IL TEOREMA DI CORRETTEZZA DELL'ALGORITMO IMPLICA CHE SE TERMINIAMO CON UN PIANO PARZIALMENTE ORDINATO, ALLORA QUALSIASI ORDINE DEI PASSI CHE SIA COERENTE CON I VINCOLI DI ORDINE DEL PIANO RAGGIUNGERÀ L'OBIETTIVO

PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



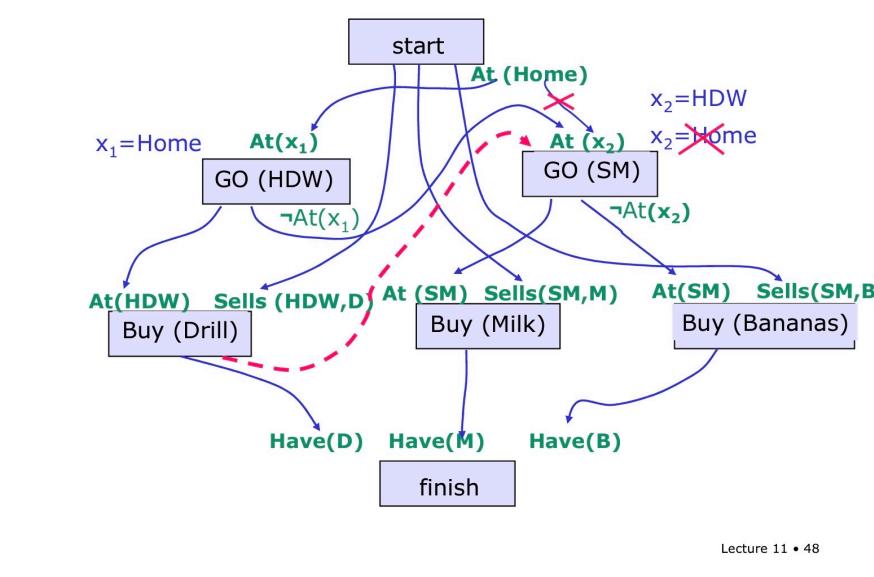
Lecture 11 • 47

SI NOTI CHE QUESTO NON È L'UNICO PIANO POSSIBILE CHE AVREMMO POTUTO ELABORARE PER QUESTO DOMINIO. SE AVESSIMO FATTO ALCUNE SCELTE IN MODO DIVERSO, AVREMMO POTUTO CONCLUDERE CON UN PIANO CHE PREVEDEVA DI ANDARE AL SUPERMERCATO PRIMA CHE DAL FERRAMENTA. ANCHE QUESTO SAREBBE ANDATO BENE.

PROF. ROSARIO SORBELLO



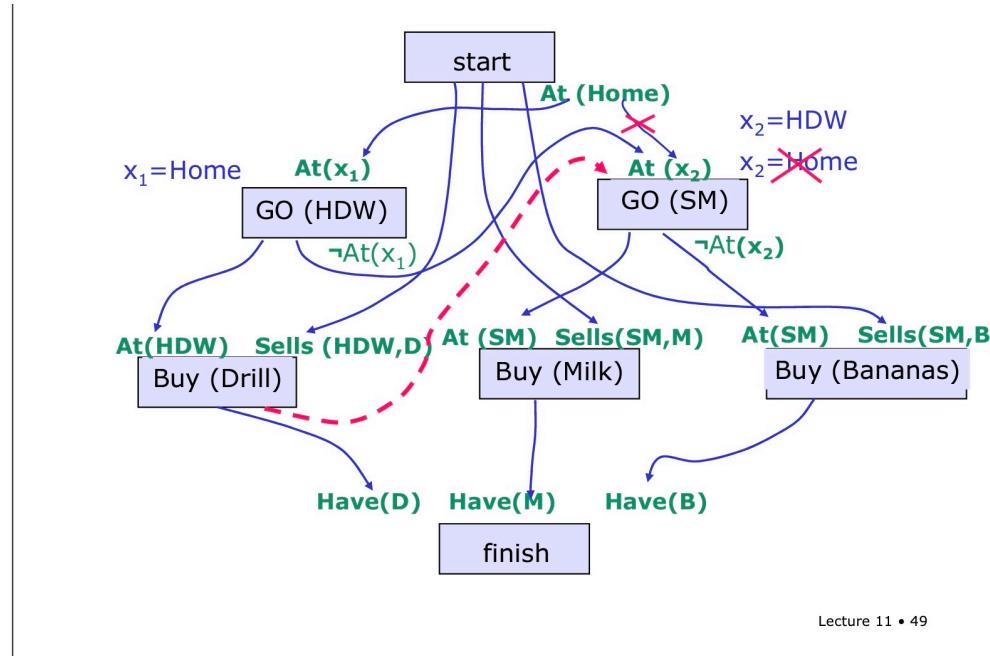
## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



L'ALGORITMO È CORRETTO E COMPLETO. QUINDI, SE C'È UN PIANO, LO TROVERÀ, E NON TROVERÀ MAI UN PIANO CHE SIA SEMPLICEMENTE ROTTO. MA NON È EFFICIENTE. NON SI PUÒ GARANTIRE CHE LO TROVI RAPIDAMENTE, NÉ CHE TROVI IL PIANO MIGLIORE. QUINDI, NON C'È NULLA CHE VI IMPEDISCA DI ANDARE A CASA, ANDARE DAL FERRAMENTA, TORNARE A CASA, ANDARE AL SUPERMERCATO. QUANDO ABBIAMO DOVUTO SCEGLIERE QUALCOSA PER SODDISFARE LE PRECONDIZIONI, SIAMO STATI MOLTO CONSERVATIVI NELLA SCELTA E ABBIAMO CERCATO DI RIUTILIZZARE I PASSI E LE VARIABILI CHE AVEVAMO GIÀ A DISPOSIZIONE, MA SE NON FOSSIMO STATI COSÌ CONSERVATIVI, AVREMMO POTUTO ISTANZIARE NUOVE AZIONI TUTTO IL GIORNO.

PROF. ROSARIO SORBELLO

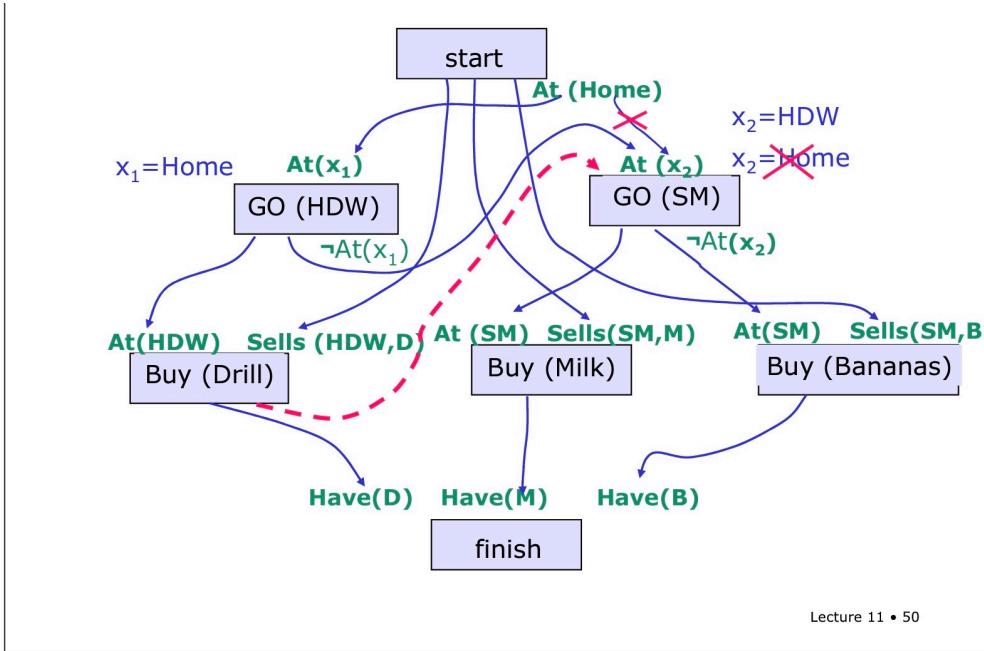
## PIANO PARZIALMENTE ORDINATO (ESEMPIO)



QUINDI, QUESTO È UN PROBLEMA POTENZIALE. TUTTI I DISCORSI SULLA SCELTA NON DETERMINISTICA NELL'ALGORITMO FACEVANO SEMBRARE CHE NON IMPORTASSE QUALI SCELTE SI FACCESERO PER PRIME. BEH, FORSE NON HA IMPORTANZA PER LA CORRETTEZZA DELL'ALGORITMO, MA CERTAMENTE HA IMPORTANZA PER L'EFFICIENZA! SIA NEL SENSO DI TROVARE UN PIANO VELOCEMENTE, SIA NEL SENSO DI TROVARE UN PIANO BREVE. QUINDI, AFFINCHÉ L'ALGORITMO SIA EFFICIENTE, È NECESSARIO ESSERE INTELLIGENTI NEL PROVARE LE COSE IN UN ORDINE RAGIONEVOLE.

PROF. ROSARIO SORBELLO

## PIANO PARZIALMENTE ORDINATO (ESEMPIO)

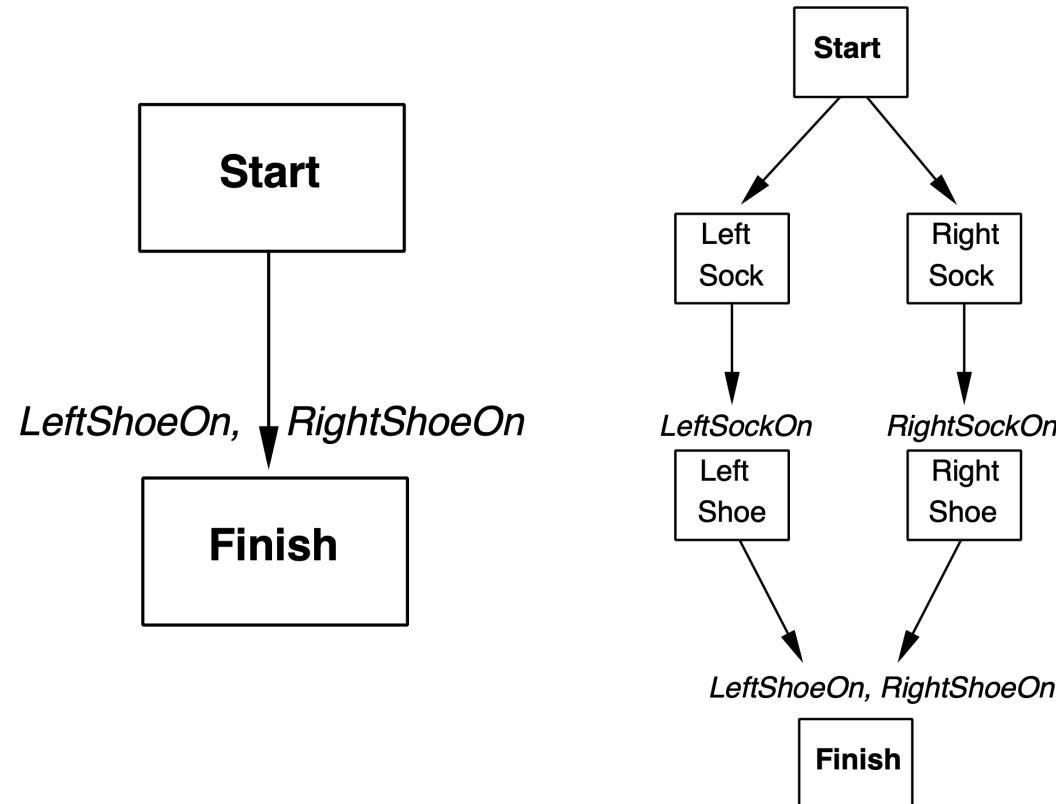


SE RIUSCITE A SODDISFARLO CON UN PASSO CHE AVETE GIÀ, È UN BUON INIZIO. OPPURE CERCATE DI SODDISFARLA CON UNA VARIABILE CHE AVETE GIÀ USATO UNA VOLTA. CI SONO ALCUNE IDEE GENERALI DI QUESTO TIPO. MA A VOLTE, IN UN DOMINIO REALE, DOVRETE USARE LA CONOSCENZA DEL DOMINIO PER DIRE CHE, DATI QUESTI DUE MODI DI PROVARE A SODDISFARE QUALCOSA, PROVATE SEMPRE QUESTO PER PRIMO. PER ESEMPIO, SE AVETE BISOGNO DI UN NUOVO OGGETTO, PROVATE SEMPRE A COMPRARLO PRIMA DI PROVARE A COSTRUIRLO.

PROF. ROSARIO SORBELLO



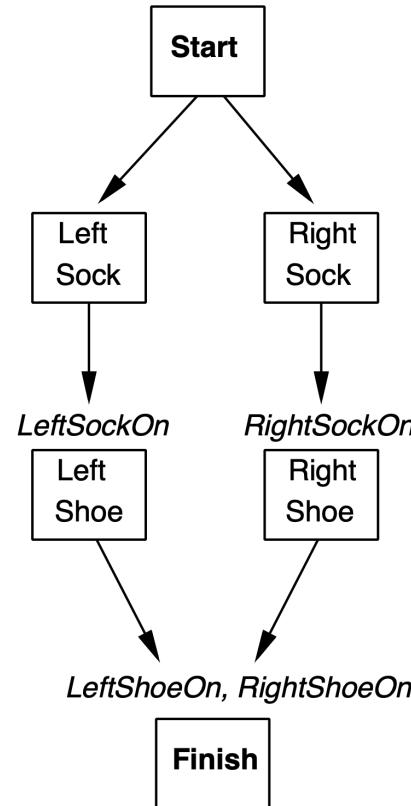
## ORDINAMENTO PARZIALE



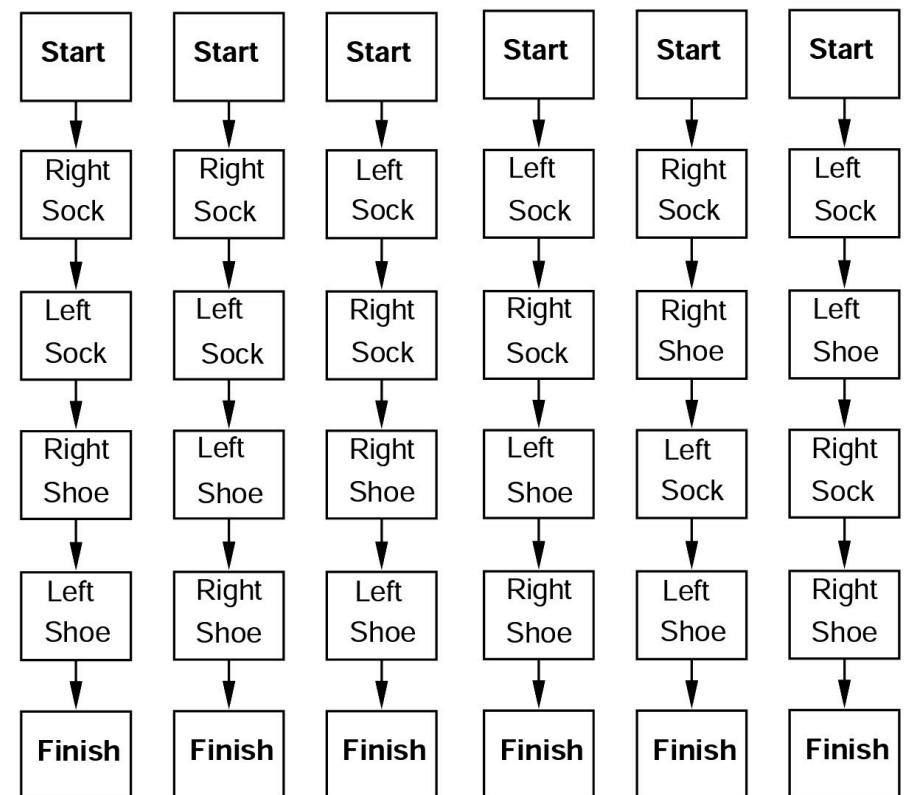
PROF. ROSARIO SORBELLO

# ORDINAMENTO PARZIALE e ORDINAMENTO TOTALE

Partial Order Plan



Total Order Plans:



PROF. ROSARIO SORBELLO



1. **NODO:** PIANO PARZIALE
2. DA UN PIANO INCOMPLETO AD UN PIANO COMPLETO
3. **CONDIZIONE APERTA:** PRECONDIZIONE NON VERIFICATA
4. **OPERATORI:**
  - AGGIUNGI AZIONE PER VERIFICARE UNA CONDIZIONE APERTA
  - AGGIUNGI LINK DA UNA AZIONE AD UNA CONDIZIONE APERTA
  - ORDINA UNA AZIONE RISPETTO UN'ALTRA



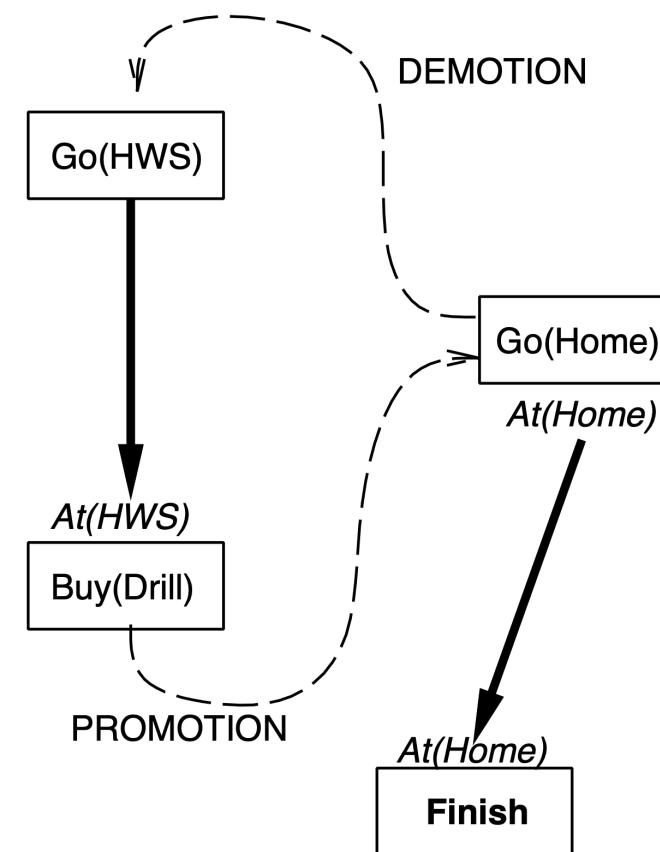
- **PIANO COMPLETO:** TUTTE LE PRECONDIZIONI SONO VERIFICATE
- **PRECONDIZIONE VERIFICATA:** UNA AZIONE LA VERIFICA E NESSUNA AZIONE LA “MINACCIA”
- **RISULTATO:** INSIEME DI AZIONI CON VINCOLI CAUSALI



Go(Home) minaccia At(HWS)

Prima di Go(HWS)

Dopo Buy(Drill)

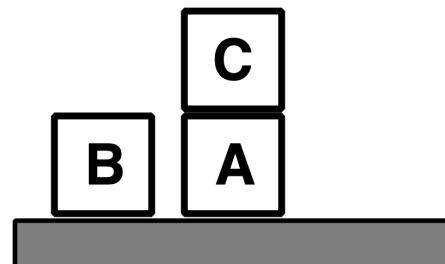


PROF. ROSARIO SORBELLO

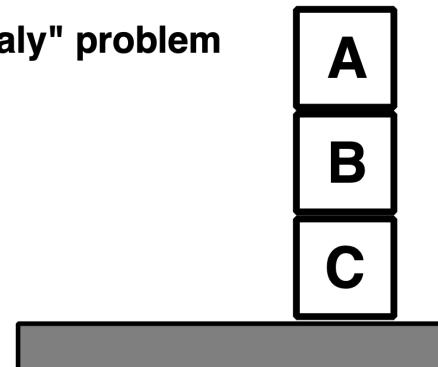


## ANOMALIA DI SUSSMAN RIVISTA

"Sussman anomaly" problem



Start State



Goal State

*Clear(x) On(x,z) Clear(y)*

*PutOn(x,y)*

*~On(x,z) ~Clear(y)  
Clear(z) On(x,y)*

*Clear(x) On(x,z)*

*PutOnTable(x)*

*~On(x,z) Clear(z) On(x,Table)*

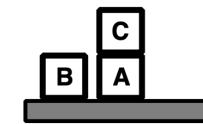


**ROBOTICS LAB**

## PIANO INIZIALE

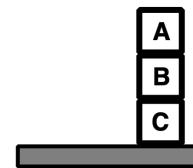
START

*On(C,A) On(A,Table) Cl(B) On(B,Table) Cl(C)*



*On(A,B) On(B,C)*

FINISH

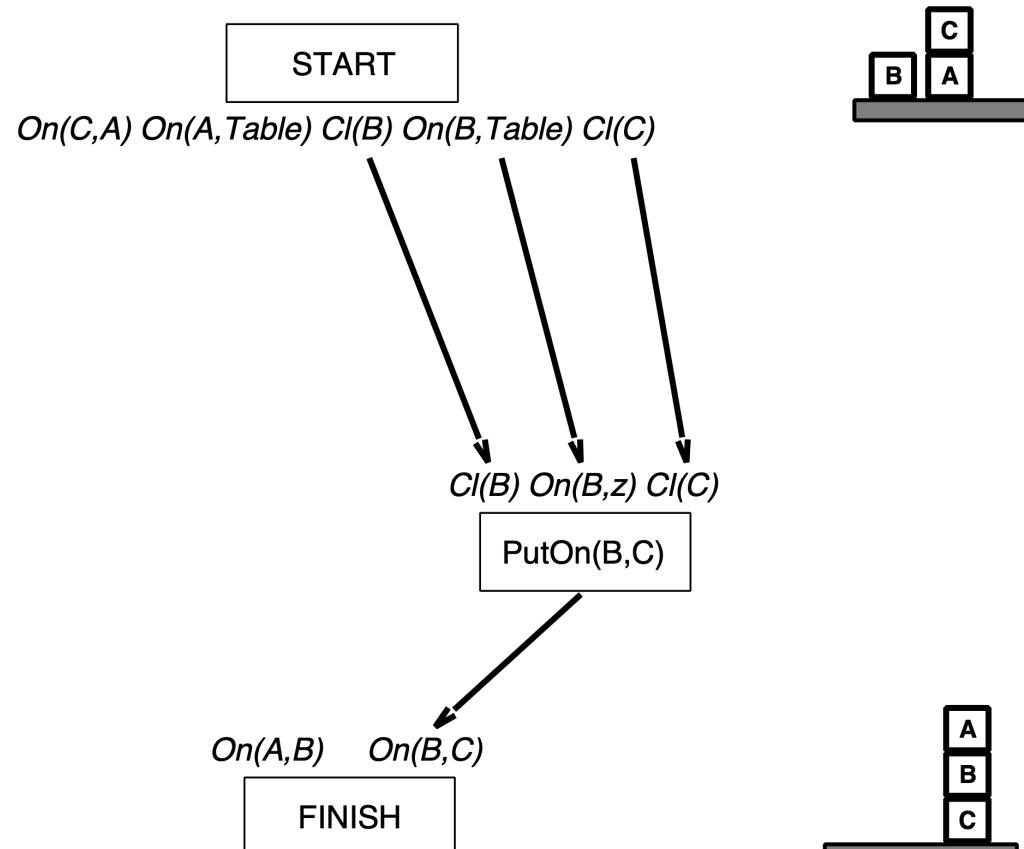


PROF. ROSARIO SORBELLO



ROBOTICS LAB

## PIANO PARZIALE

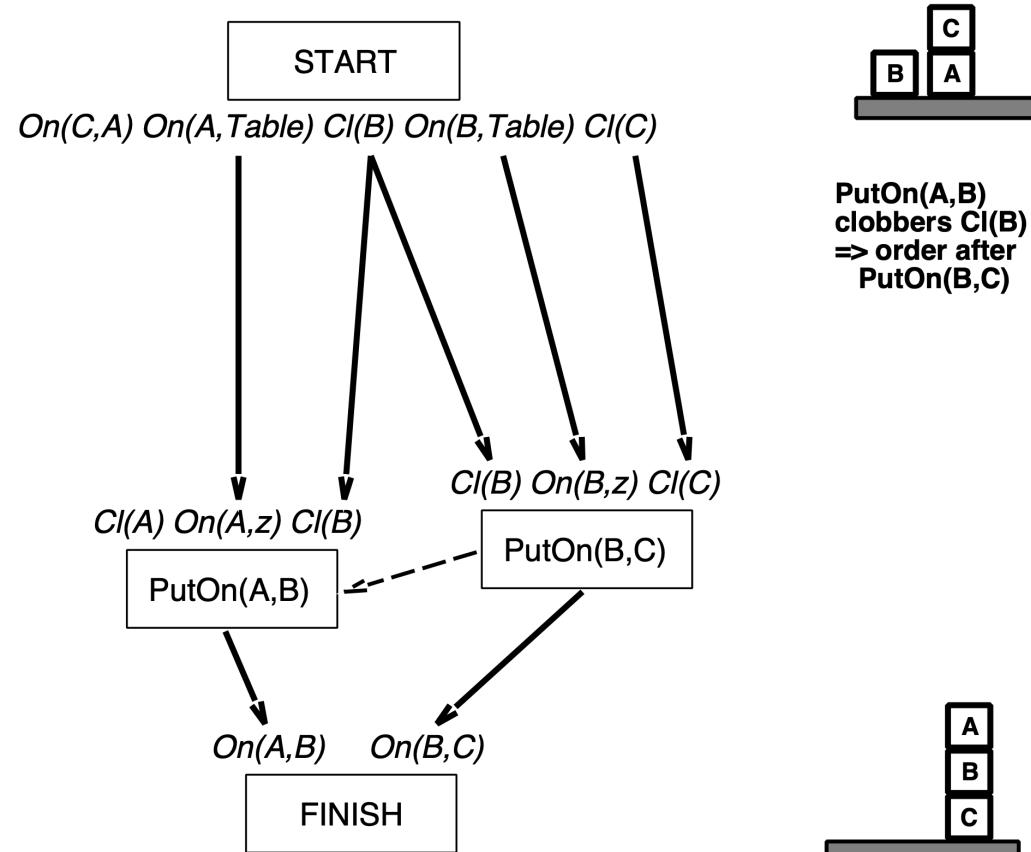


PROF. ROSARIO SORBELLO



ROBOTICS LAB

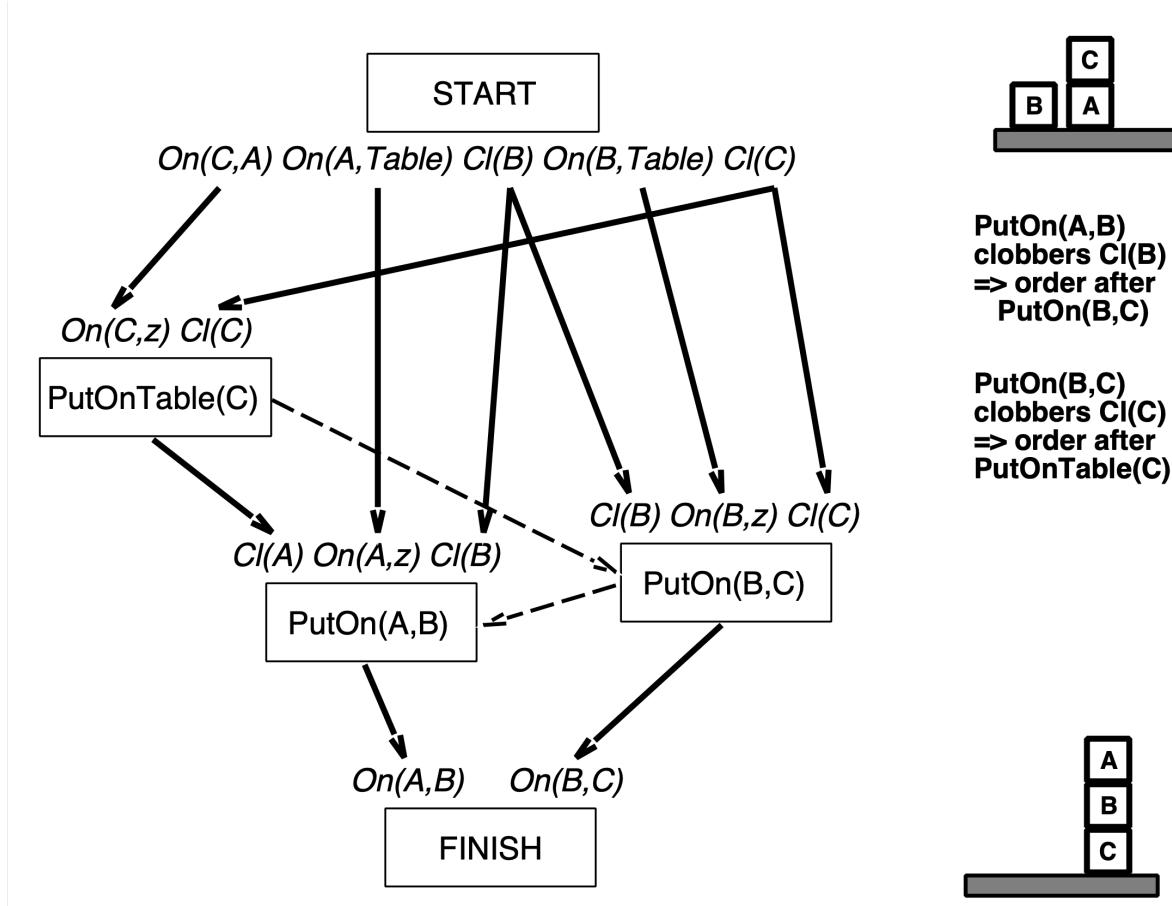
## NUOVO PIANO PARZIALE CON «MINACCIA»



PROF. ROSARIO SORBELLO



## PIANO COMPLETO



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## PIANIFICAZIONE NEL MONDO REALE

---

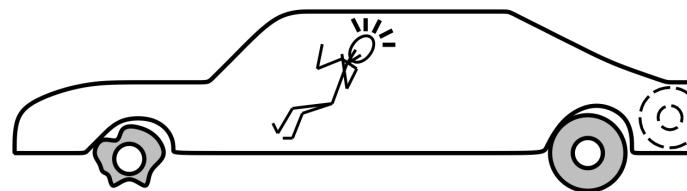
- AZIONI POSSONO FALLIRE
- INFORMAZIONI INIZIALI INCORRETTE
- NON SI POSSONO ANALIZZARE TUTTE LE PRECONDIZIONI
- NON SI POSSONO PREVEDERE TUTTI GLI EFFETTI DELLE AZIONI

PROF. ROSARIO SORBELLO



ROBOTICS LAB

## MONDO REALE



START

$\neg Flat(Spare)$   $Intact(Spare)$   $Off(Spare)$   
 $On(Tire1)$   $Flat(Tire1)$

$On(x)$   $\neg Flat(x)$

FINISH

$On(x)$

**Remove(x)**

$Off(x)$   $ClearHub$

$Off(x)$   $ClearHub$

**Puton(x)**

$On(x)$   $\neg ClearHub$

$Intact(x)$   $Flat(x)$

**Inflate(x)**

$\neg Flat(x)$

PROF. ROSARIO SORBELLO



- INFORMAZIONI SUL MONDO: AZIONI DI “SENSING”
- SOTTOPIANI PER OGNI CONTINGENZA:  
Check(Tire1),  
If(Intact(Tire1),  
    Inflate(Tire1),  
    Call(ACI))
- DISPENDIOSO: PREVEDERE CONTROLLI PER TUTTI I PROBLEMI POSSIBILI



- AZIONI DI “SENSING”: CONTROLLO MEDIANTE SENSORI SE UNA CONDIZIONE È VERIFICATA
- PREVEDERE COSA FARE IN ENTRAMBI I CASI
- IF(P, PIANO1, PIANO2)

CheckTire(x)

*KnowsIf(Intact(x))*



**ROBOTICS LAB**

## ESEMPIO

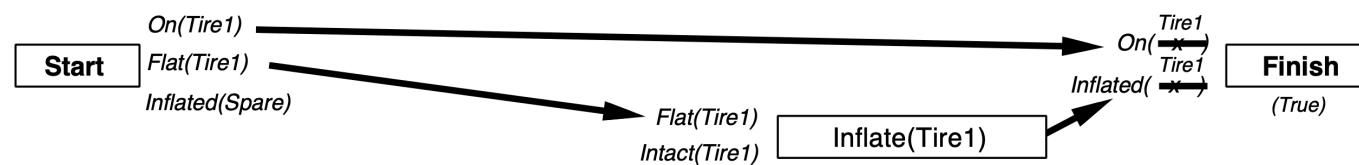


PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## ESEMPIO

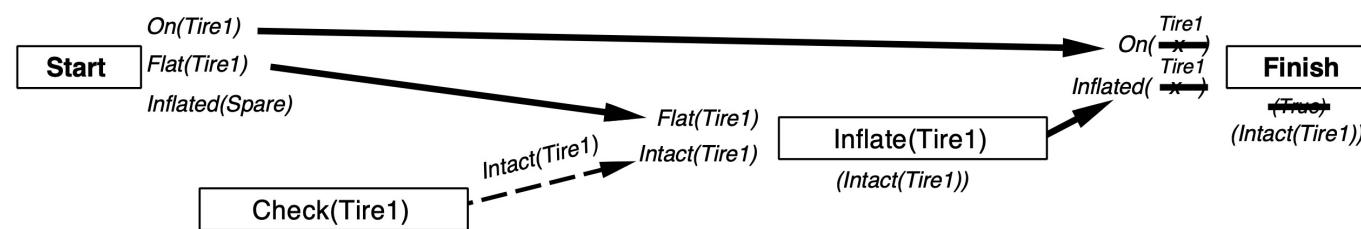


PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## ESEMPIO

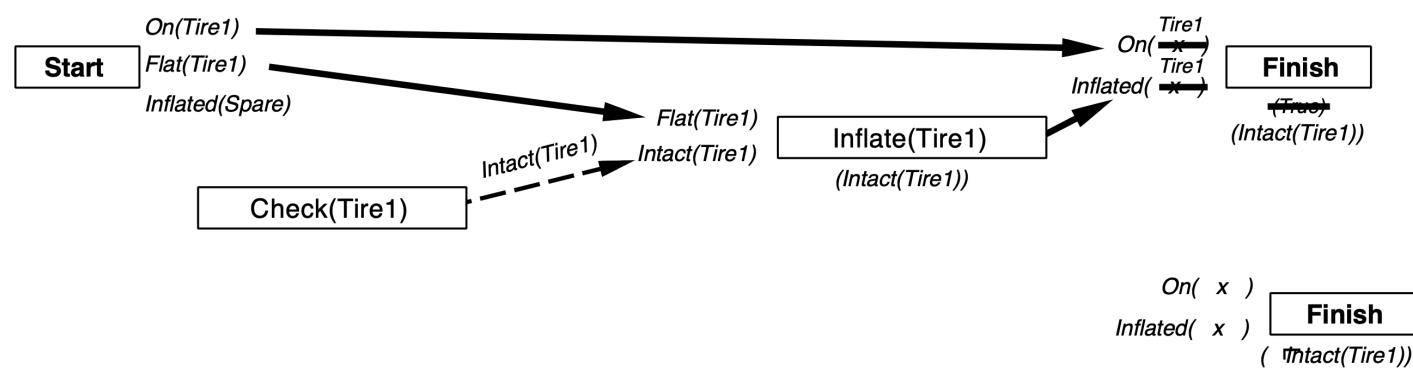


PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## ESEMPIO

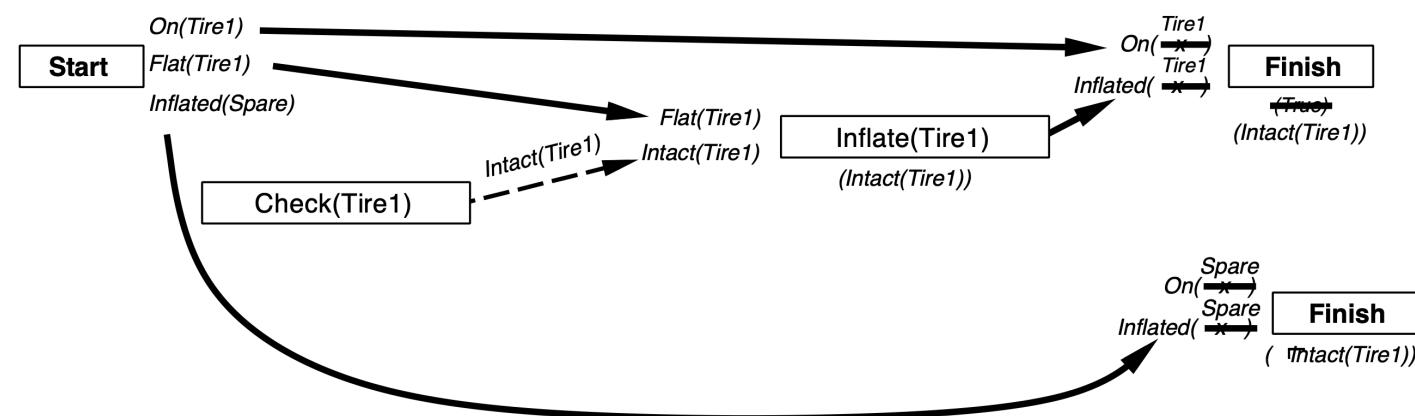


PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

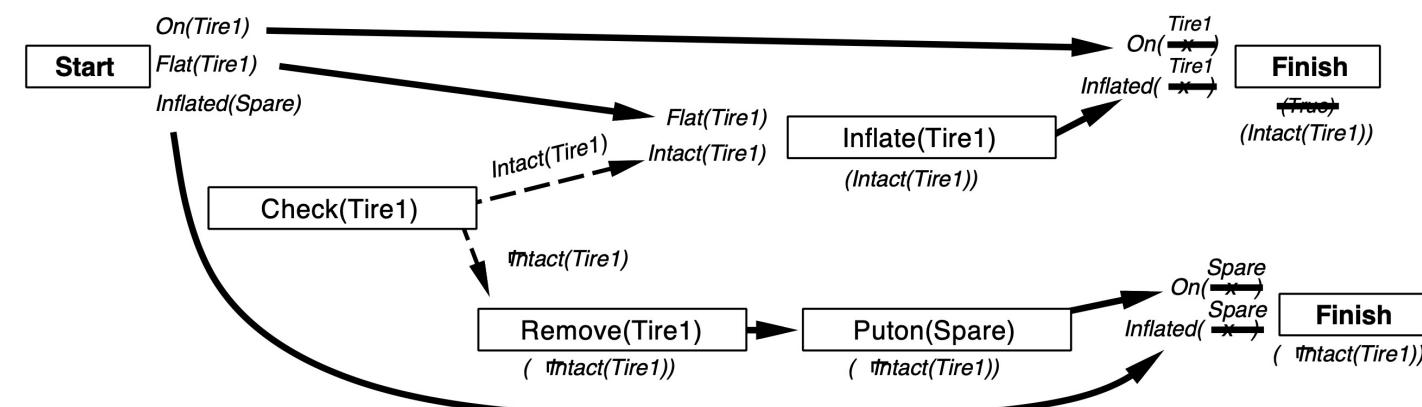
## ESEMPIO



PROF. ROSARIO SORBELLO



## ESEMPIO



PROF. ROSARIO SORBELLO

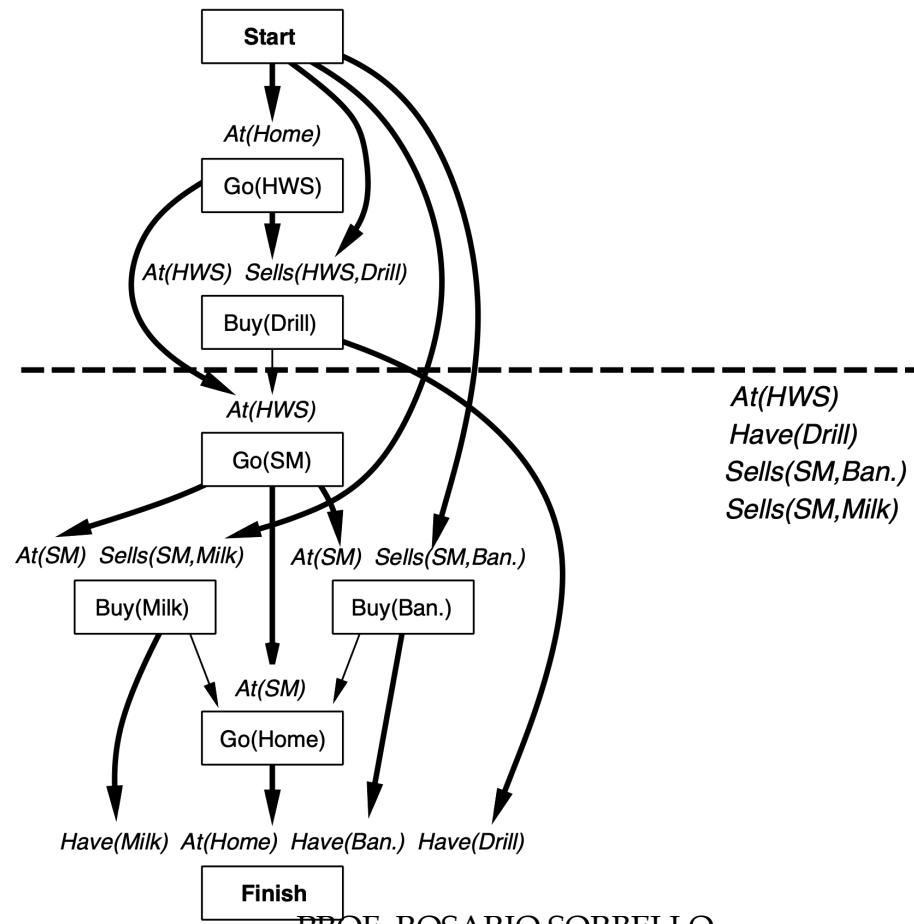


- GENERAZIONE DEL PIANO
- VERIFICA DEI PROGRESSI DURANTE L'ESECUZIONE
- MONITORAGGIO DELL'ESECUZIONE DEL PIANO:
  - ERRORE SU PRECONDIZIONI DEL PIANO RESTANTE (LINK CAUSALI DELLE AZIONI EFFETTUATE)
- MONITORAGGIO DELLA SINGOLA AZIONE:
  - ERRORE SE LA PRECONDIZIONE DELLA PROSSIMA AZIONE NON È VERIFICATA (LINK CAUSALE DI UNA AZIONE GIÀ EFFETTUATA)

PROF. ROSARIO SORBELLO



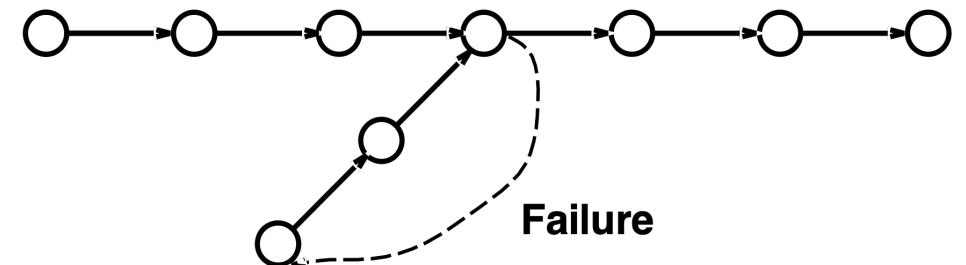
## PRECONDIZIONI DEL PIANO RESTANTE



PROF. ROSARIO SORBELLO



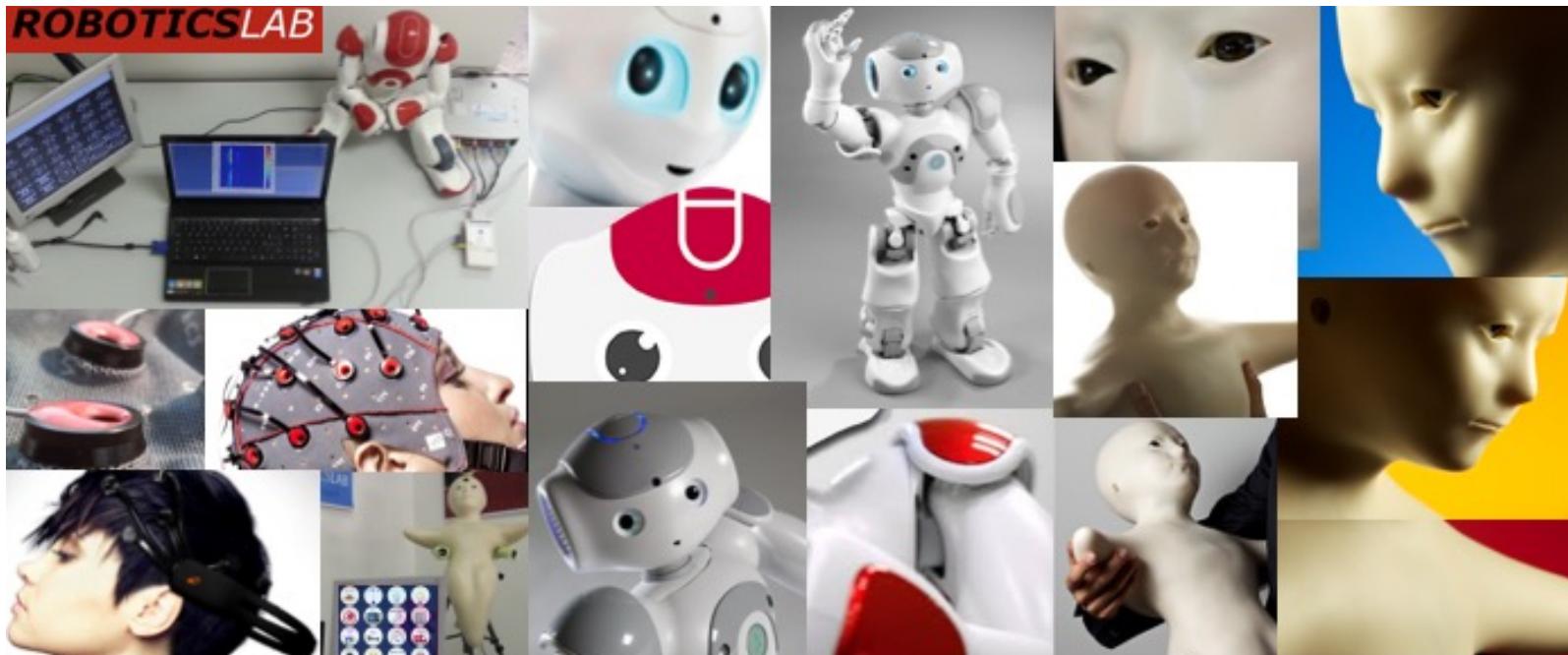
- GENERAZIONE DI UN NUOVO PIANO PER RAGGIUNGERE IL GOAL
- PRECONDIZIONI NON VERIFICATE
- NUOVO PIANO “DA ZERO”
- PIANO “PATCH” PER TORNARE AL PIANO PRINCIPALE



PROF. ROSARIO SORBELLO



OSAKA UNIVERSITY



ATR

g-tec

## ARCHITETTURE IBRIDE

PROF. ROSARIO SORBELLO

5° LEZIONE - PALERMO 24 OTTOBRE 2022



**ROBOTICS LAB**

# Obiettivi

---

- Capire le limitazioni dei due approcci
- Capire i principi ispiratori
- Capire le caratteristiche di interfaccia
- Studiare esempi concreti

PROF. ROSARIO SORBELLO



## Assunzioni dei sistemi puramente reattivi

---

- L'ambiente manca di consistenza temporale e stabilità
- La percezione immediata del robot è adeguata
- E' difficile localizzare il robot relativamente ad un modello del mondo
- La conoscenza simbolica ha valore scarso o nullo



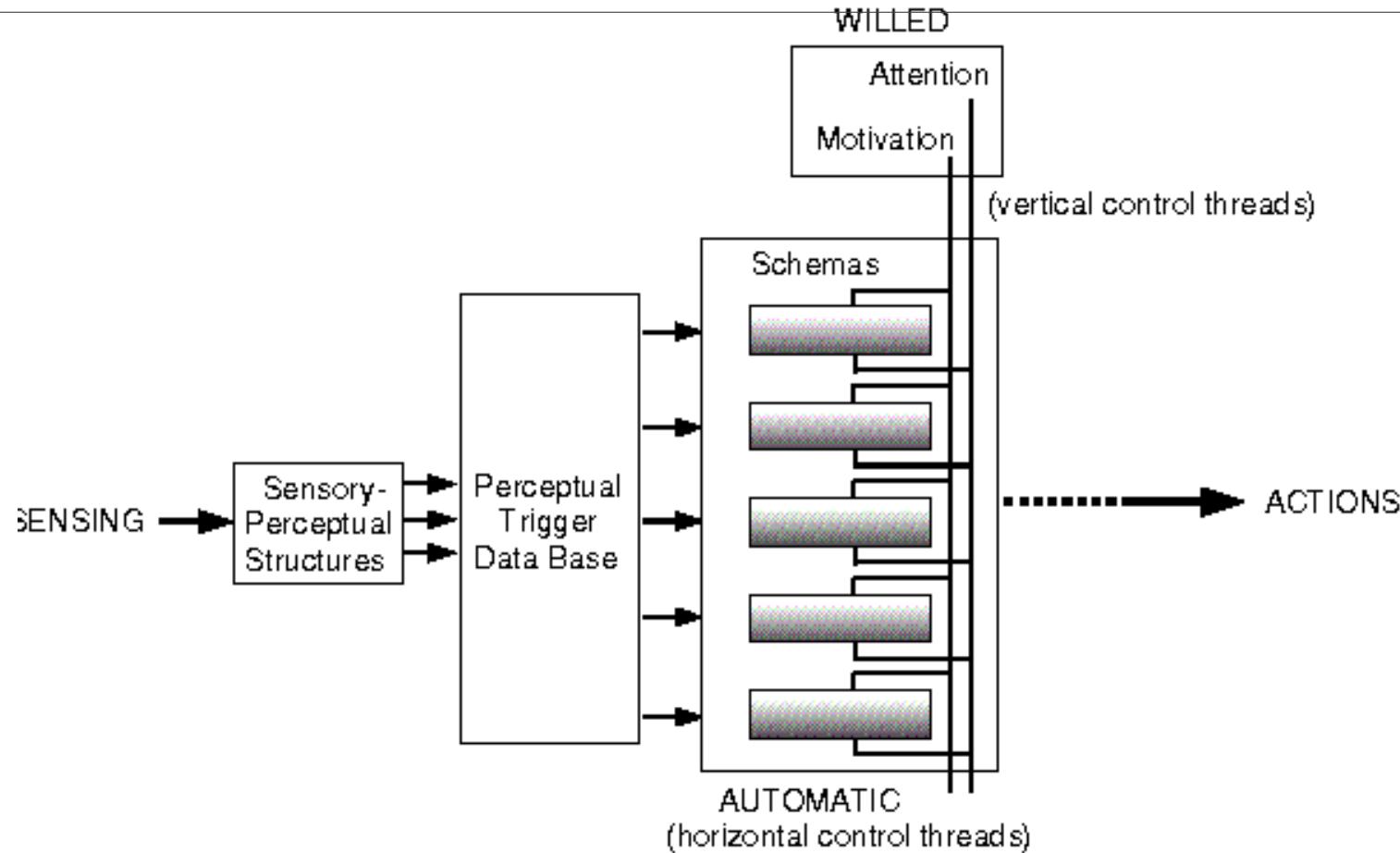
# Modello di Norman-Shallice

---

- Controllo volontario
  - Pianificazione, decisioni
  - Riparazione di guasti
  - Azioni nuove o poco note
  - Azioni difficili o pericolose
  - Superamento di schemi prefissati
- Controllo automatico
  - Eseguito senza coscienza
  - Attività multiple in parallelo (schemi)
  - Non necessita attenzione alla partenza



# Modello di Norman-Shallice



PROF. ROSARIO SORBELLO



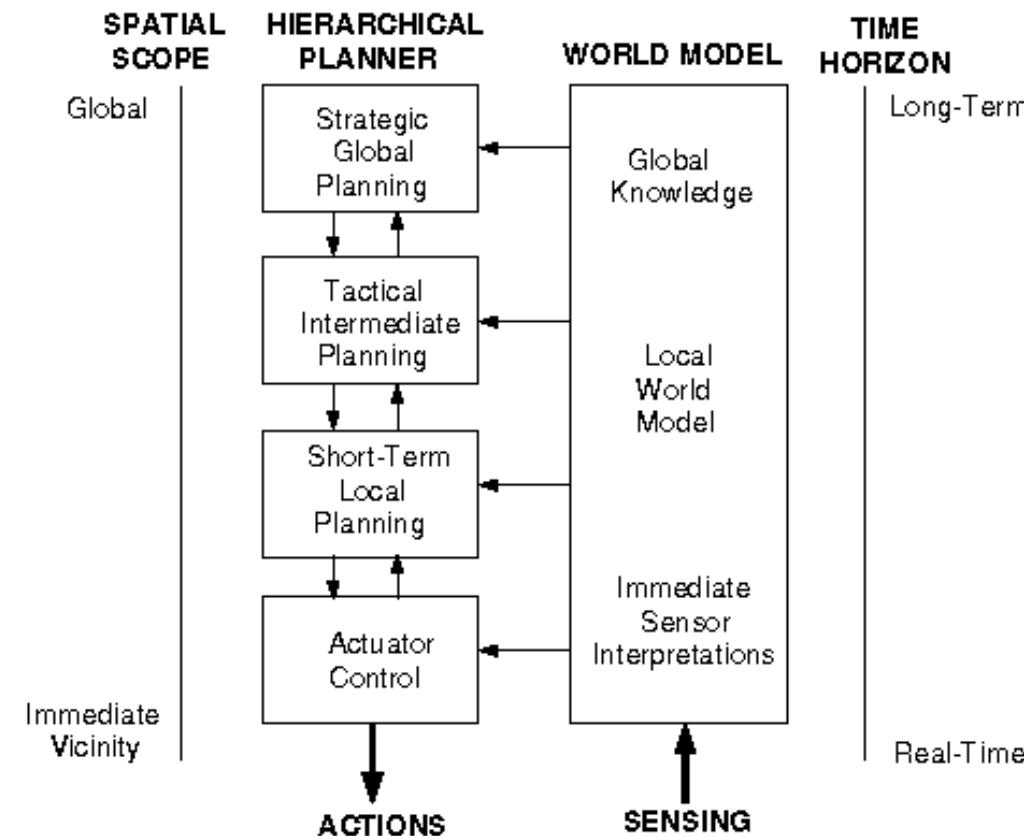
## Modello N-S: Connessioni tra deliberazione e reattività

---

- Gli schemi automatici sono modulati dall'attenzione che si sviluppa dal controllo deliberativo
- Gli schemi competono tra loro
- La selezione degli schemi è la funzione principale del controllo deliberativo
- Evidenze psicologiche a supporto



# Pianificazione Deliberativa/Gerarchica



PROF. ROSARIO SORBELLO



## Relativi Punti di Forza

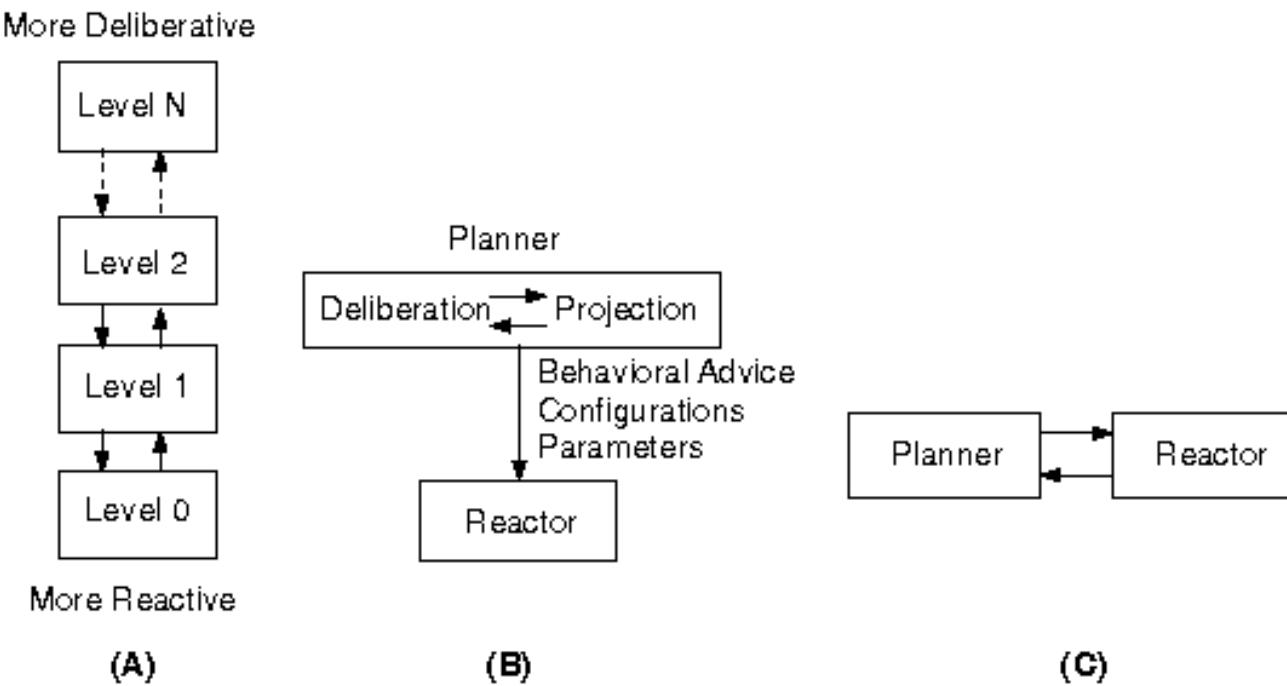
---

- **PIANIFICATORI DELIBERATIVI**
  - SONO BASATI SULLA CONOSCENZA DEL MONDO
  - POSSONO INTEGRARE NUOVA CONOSCENZA
  - HANNO AMPIE PROSPETTIVE E SCOPI
- **SISTEMI BASATI SU COMPORTAMENTI**
  - PERMETTONO LO SVILUPPO MODULARE
  - FUNZIONAMENTO IN TEMPO REALE NEL MONDO REALE
  - CRESCITA INCREMENTALE
  - STRETTO ACCOPPIAMENTO CON DATI SENSORIALI



# Pianificazione e sistemi reattivi

- Integrazione gerarchica
- Pianificazione che guida i comportamenti reattivi
- Accoppiamento pianificazione-comportamenti



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Principali strategie di interfaccia

---

- Selezione: Pianificatore visto come configurazione
- Consulenza: Pianificatore visto come generatore di consigli
- Adattamento: Pianificatore visto come adattamento del controllore
- Posposizione: Pianificatore visto come processo a bassa priorità

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# AuRA

---

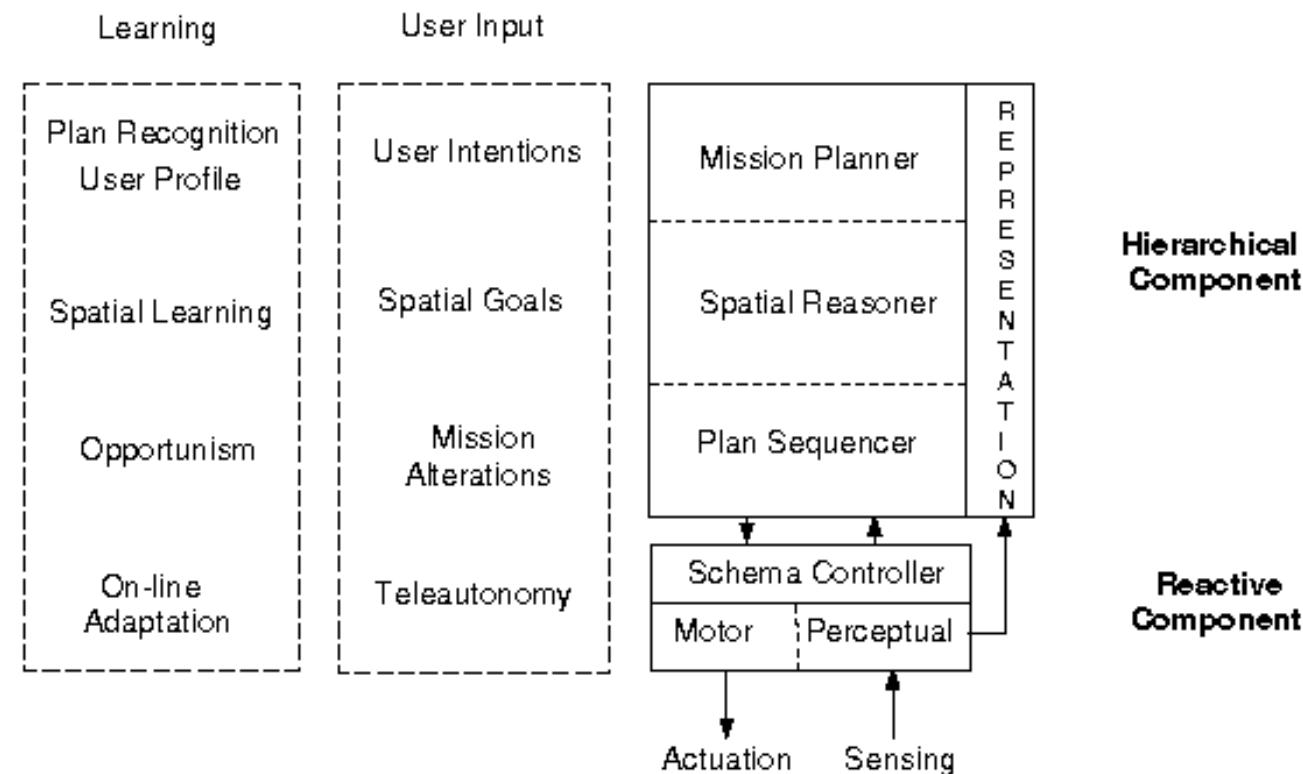
- Arkin-1986
- Pianificazione vista come configurazione
- Pianificatore iniziale A\* integrato con controllore basato su schemi
- Permette la modularità, flessibilità, adattabilità

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# AuRA



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Atlantis

---

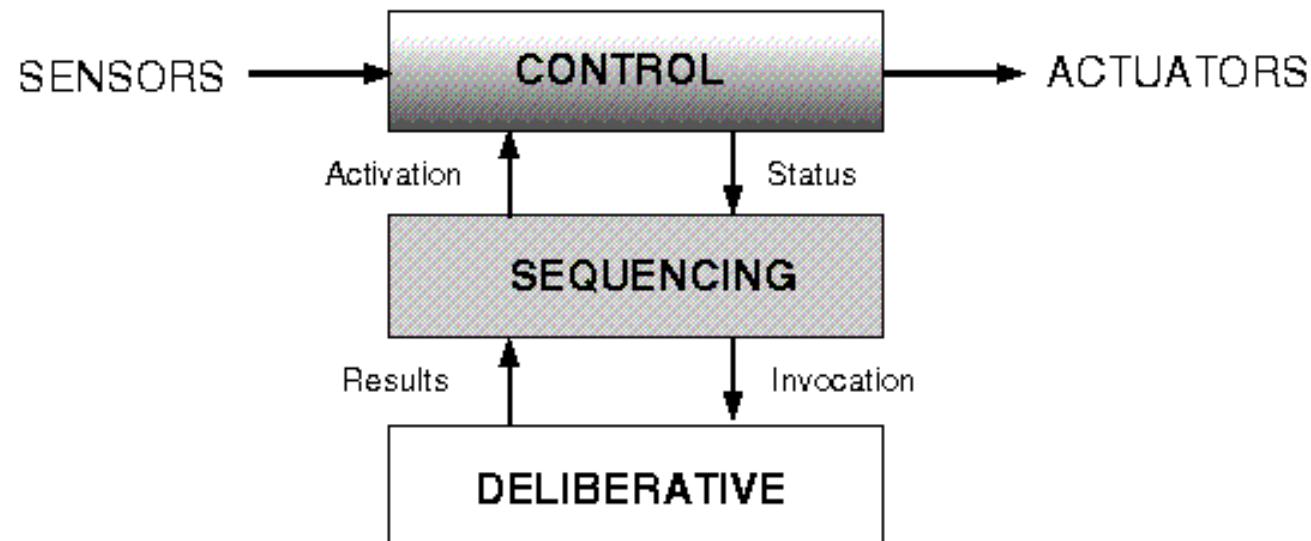
- Gat - 1991
- Tre livelli: controllore, sequenziatore, deliberatore
- Reattività e deliberazione asincrone ed eterogenee
- Implementato in alfa
- Pianificatore come generatore di consigli non obbligatori
- Nozione di guasto
- Provato sui rover della NASA

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Architettura Atlantis



PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# Rover NASA Robby

---



PROF. ROSARIO SORBELLO



# Architettura Planner-Reactor

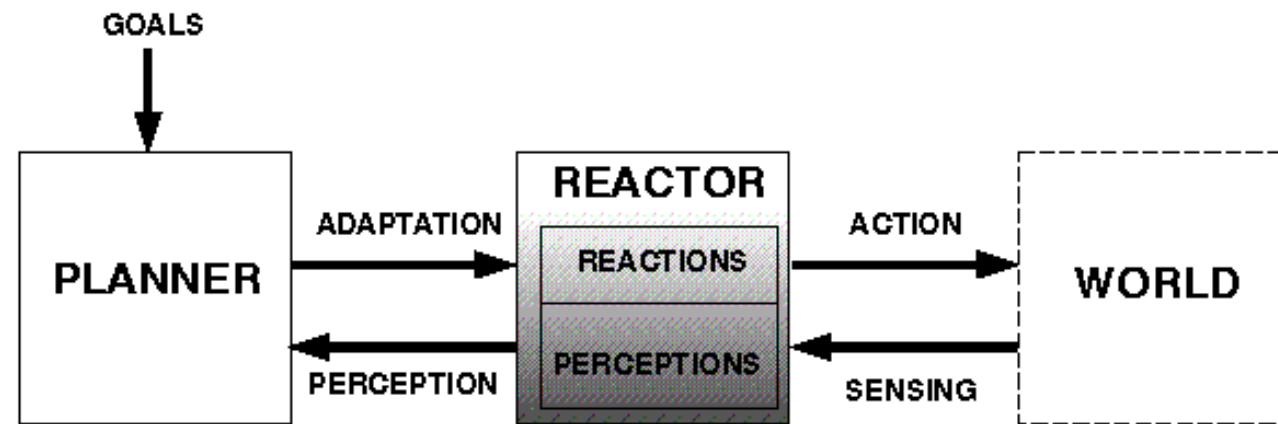
---

- Lyons 1992
- Modifica continua del sistema di controllo reattivo
- Pianificazione come forma di adattamento del sistema reattivo
- Adattamento on-line invece che deliberazione off-line
- La pianificazione è utilizzata per rimuovere eventuali errori di funzionamento
- Usa il modello RS
- Testata sia in celle di assemblaggio che nella pianificazione della presa



**ROBOTICS LAB**

# Architettura Planner-Reactor

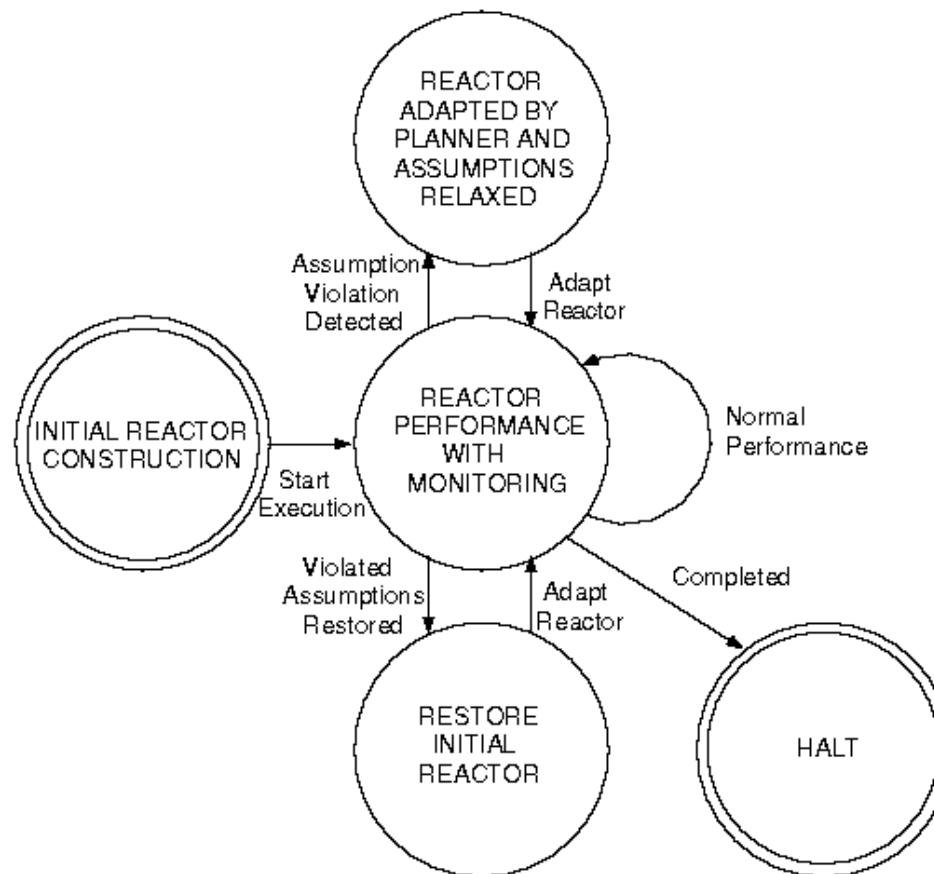


PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

# FSA dell'Architettura Planner-Reactor



PROF. ROSARIO SORBELLO



ROBOTICS LAB

# Procedural Reasoning System

- Georgeff e Lansky 1987
- Posposizione della pianificazione vista
- Provata sul robot SRI Flakey



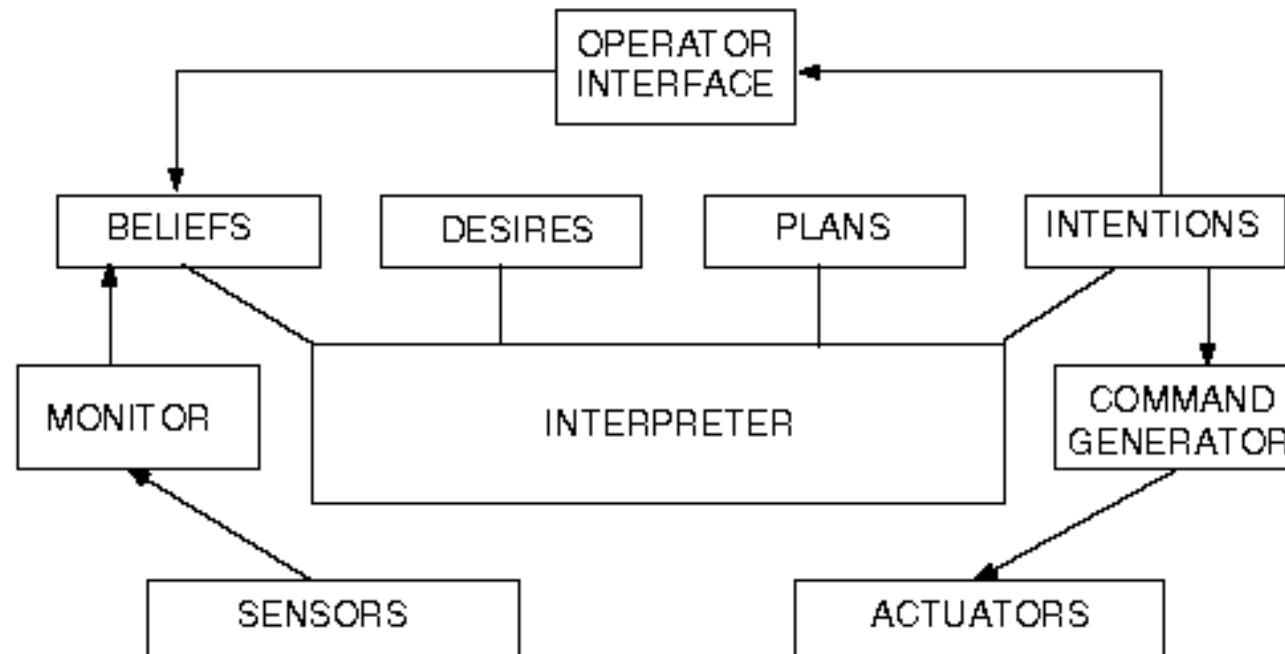
a priorità

PROF. ROSARIO SORBELLO



ROBOTICS LAB

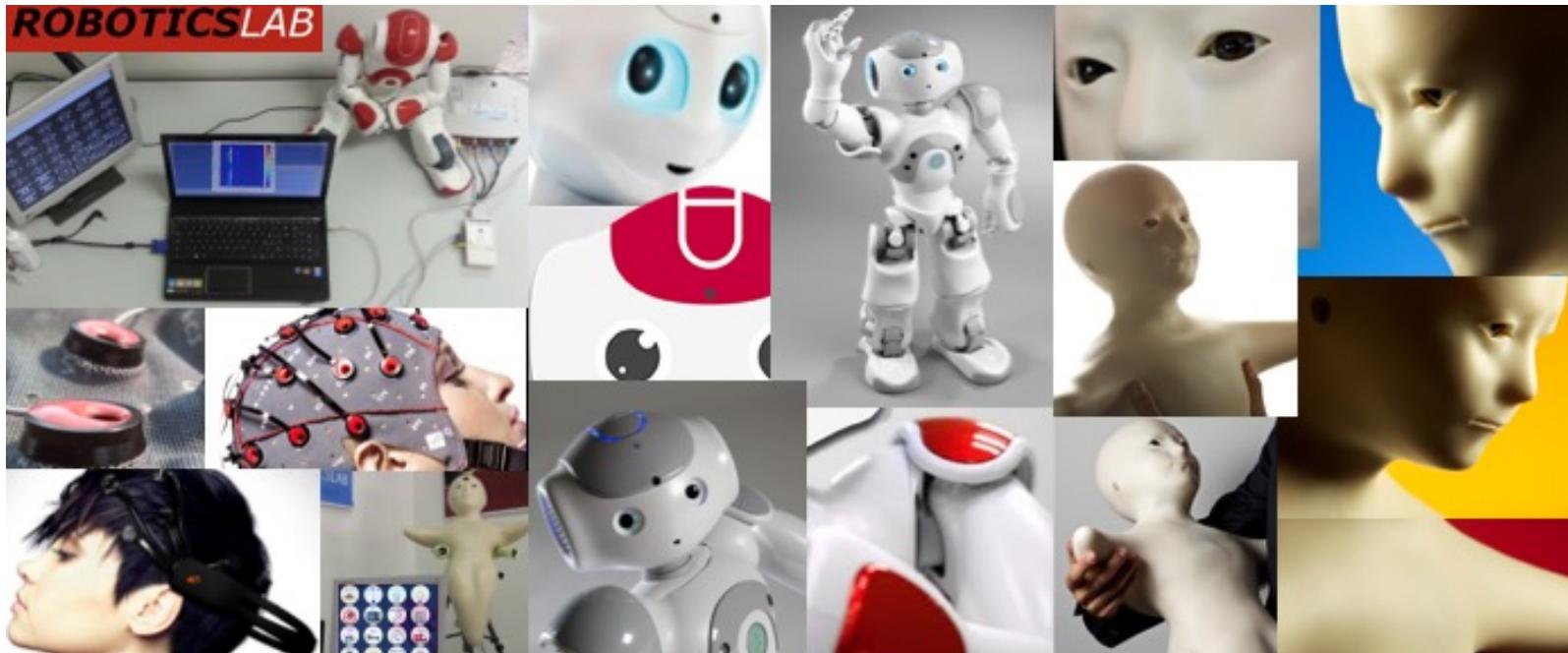
# PRS



PROF. ROSARIO SORBELLO



OSAKA UNIVERSITY



ATR

g-tec

## PIANIFICATORE STRIPS

PROF. ROSARIO SORBELLO

5° LEZIONE - PALERMO 24 OTTOBRE 2022



- Acronimo per Stanford Research Institute Problem Solver (STRIPS)
- Antenato degli attuali sistemi di pianificazione (primi anni '70)
- Ha un proprio linguaggio per la rappresentazione di azioni, con una sintassi molto più semplice del Situation Calculus (meno espressività, ma più efficienza).
- **Planner lineare** basato su ricerca **backward** per la costruzione di piani
- Assume che lo stato iniziale sia completamente noto (**Closed World Assumption**)

Numerose implementazioni LISP

<http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/planning/systems/idm/0.html>

PROF. ROSARIO SORBELLO



- Rappresentazione dello stato
  - Insieme di fluent che valgono nello stato  
Esempio: *on(b,a)*, *clear(b)*, *clear(c)*, *ontable(c)*
- Rappresentazione del goal
  - Insieme di fluent (simile allo stato)
  - Si possono avere variabili  
Esempio: *on(X,a)*



- Rappresentazione delle azioni (3 liste)
  - PRECONDIZIONI: fluent che devono essere veri per applicare l'azione
  - DELETE List: fluent che diventano falsi come risultato dell'azione
  - ADD List: fluent che diventano veri come risultato dell'azione

Esempio *Move(X, Y, Z)*

Precondizioni: *on(X, Y)*, *clear(X)*, *clear(Z)*

Delete List: *clear(Z)*, *on(X, Y)*

Add list: *clear(Y)*, *on(X, Z)*

A volte ADD e DELETE list sono rappresentate come **EFFECT** list con atomi positivi e negativi

Esempio *Move(X, Y, Z)*

Precondizioni: *on(X, Y)*, *clear(X)*, *clear(Z)*

Effect List:  $\neg\text{clear}(Z)$ ,  $\neg\text{on}(X, Y)$ , *clear(Y)*, *on(X, Z)*

Il frame problem è risolto con la **STRIPS Assumption**: tutto ciò che non è specificato nella ADD e DELETE list resta immutato<sup>4</sup>



## **pickup(X)**

PRECOND: ontable(X), clear(X), handempty

DELETE: ontable(X), clear(X), handempty

ADD: holding(X)

## **putdown(X)**

PRECOND: holding(X)

DELETE: holding(X)

ADD: ontable(X), clear(X), handempty



### **stack(X,Y)**

PRECOND: holding(X), clear(Y)

DELETE: holding(X), clear(Y)

ADD: handempty, on(X,Y), clear(X)

### **unstack(X,Y)**

PRECOND: handempty, on(X,Y), clear(X)

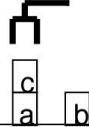
DELETE: handempty, on(X,Y), clear(X)

ADD: holding(X), clear(Y)



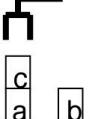
■ Utilizza due strutture dati:

- stack di goal / azioni
- descrizione S dello stato corrente

state	stack di goal
 clear(b), clear(c) on(c,a), ontable(a) ontable(b), handempty	on(c,b) $\wedge$ on(a,c)

■ Funziona backward (goal-oriented)

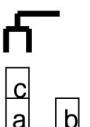
■ Partendo dallo stato obiettivo (goal), lo separa in singoli congiunti se si tratta di un AND, ad esempio:

state	stack di goal
 clear(b), clear(c) on(c,a), ontable(a) ontable(b), handempty	on(a,c) on(c,b) on(c,b) $\wedge$ on(a,c)



- I goal soddisfatti nello stato corrente sono tolti (pop) dallo stack, altrimenti ...
- Seleziona una regola che menziona il top goal atomico nella sua ADD list e la inserisce nello stack (con eventuali sostituzioni), insieme alle sue precondizioni da soddisfare:

state	clear(c) $\wedge$ holding(a) <b>stack(a,c)</b>
clear(b), clear(c)	
on(c,a), ontable(a)	on(c,b)
ontable(b)	on(c,b) $\wedge$ on(a,c)
handempty	



**stack(X,Y)**  
PRECOND:  
holding(X), clear(Y)  
DELETE:  
holding(X), clear(Y)  
ADD:  
handempty, on(X,Y),  
clear(X)

- Quando ***l'elemento affiorante dello stack è un'azione*** (eventualmente istanziata) significa che le sue precondizioni sono state soddisfatte e quindi ***la esegue***, modificando lo stato corrente (*simulando la modifica*)



# PSEUDO-CODIFICA ALGORITMO STRIPS

Inizializza stack con la congiunzione di goal final

while **stack** non è vuoto do

if `top(stack) = A and A ⊆ S` (si noti che A puo essere un `and` di goal o un atomo)

then `pop(A)` e applica la sostituzione  $\theta$  sullo **stack**

else if top(**stack** ) = a % atomico

then

- seleziona regola R con  $a \in \text{Addlist}(R)$
  - $\text{pop}(a)$ ,  $\text{push}(R)$ ,  $\text{push}(\text{Precond}(R))$ ;

else if top(stack) = a<sub>1</sub>  $\wedge$  a<sub>2</sub>  $\wedge$  ...  $\wedge$  a<sub>n</sub> % congiunzione

(\*) then push(a1), ..., push(an)

**else if** top(stack) = R % azione

then `pop(R)` e applica `R` trasformando `S`

(\*) si noti che l'ordine con cui i sottogoal vengono inseriti nello stack rappresenta un punto di scelta non deterministica. La congiunzione rimane sullo stack e verrà ri-verificata dopo - interacting goals)



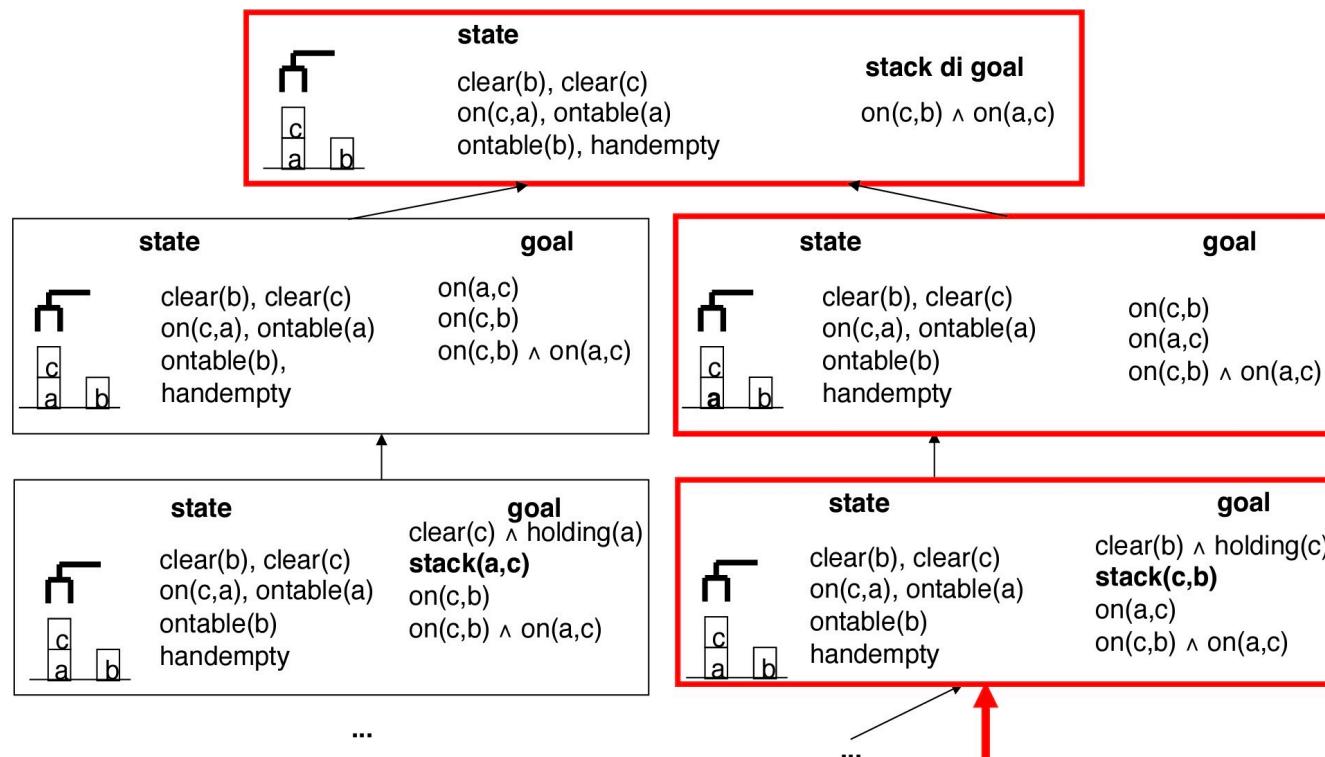
## CONSIDERAZIONI SULL'ALGORITMO

---

1. Il goal è il contenuto iniziale dello stack (la prima pila di obiettivi)
2. Suddividere il problema in sottoproblemi: ciascuno per un componente dell'obiettivo originale. Tali sottoproblemi possono interagire
3. Abbiamo tanti possibili ordini di soluzione.
4. Ad ogni passo del processo di risoluzione si cerca di risolvere il goal in cima alla pila.
5. Quando si ottiene una sequenza di operatori che lo soddisfa la si applica alla descrizione corrente dello stato ottenendo una nuova descrizione.
6. Si cerca poi di soddisfare l'obiettivo che è in cima alla pila partendo dalla situazione prodotta dal soddisfacimento del primo obiettivo.
7. Il procedimento continua fino allo svuotamento della pila.
8. Quando in cima alla pila si incontra una congiunzione si verifica che tutte le sue componenti siano effettivamente soddisfatte nello stato attuale. Se una componente non è soddisfatta (problema dell'interazione tra goal è spiegato più avanti) si reinserisce nella pila e si continua.



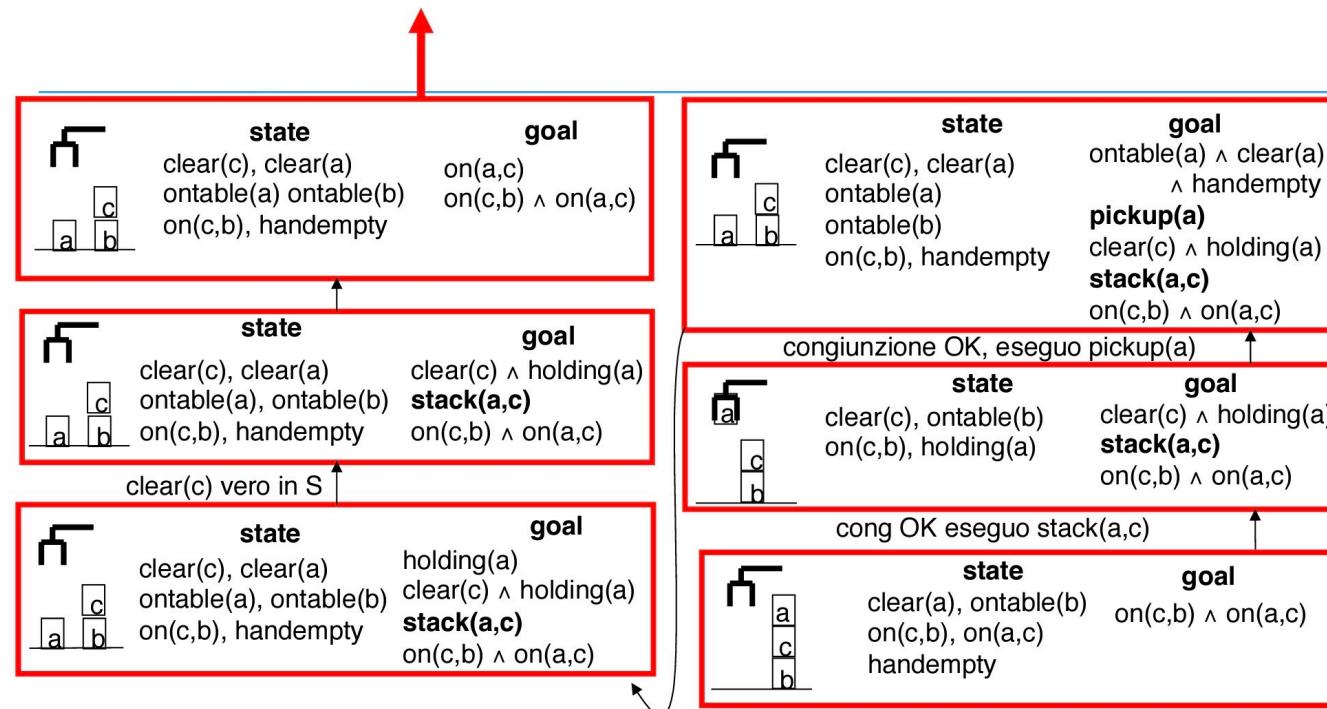
## ESEMPIO DI ALGORITMO STRIPS



PROF. ROSARIO SORBELLO



## ESEMPIO DI ALGORITMO STRIPS



Ricostruendo da configurazione iniziale  
a finale ho una soluzione:

1. unstack(c,a)

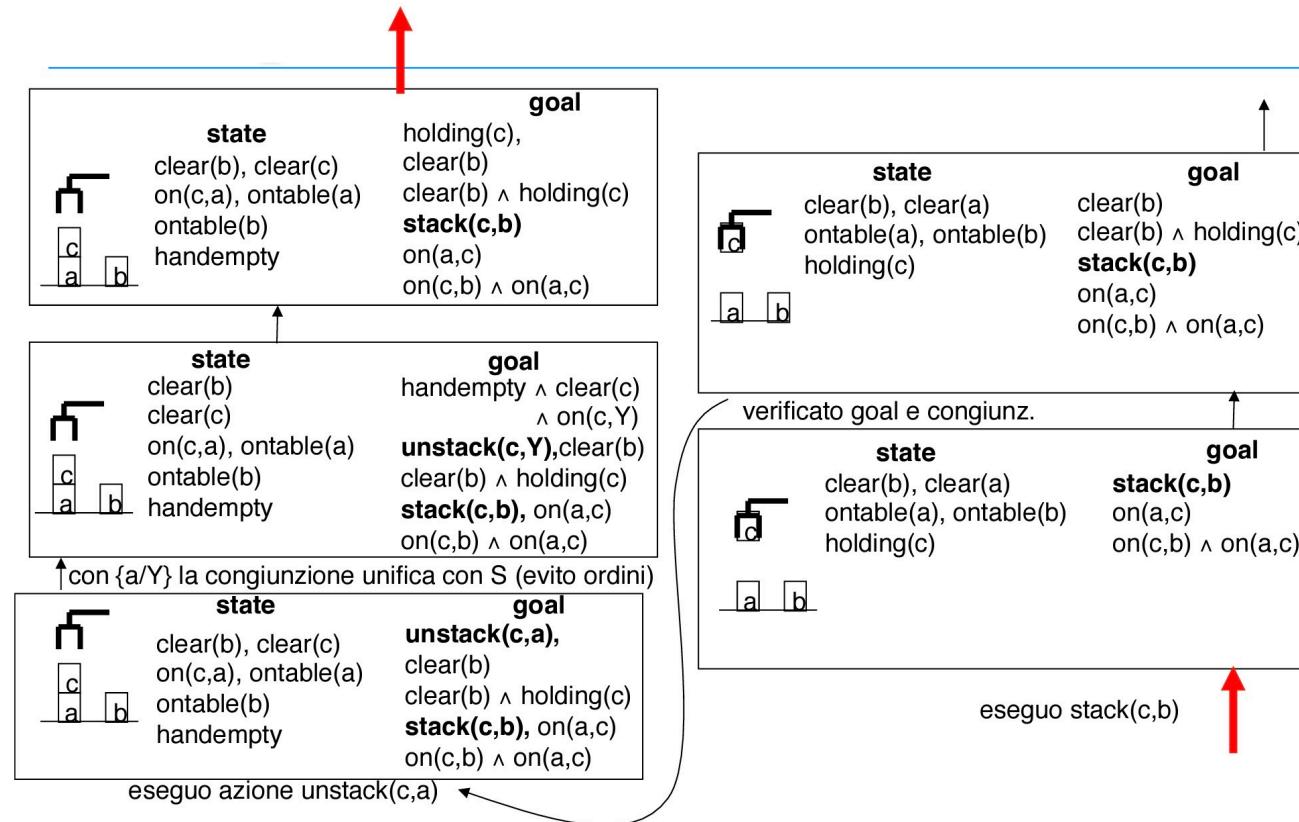
2. stack(c,b)

3. pickup(a)

4. stack(a,c)

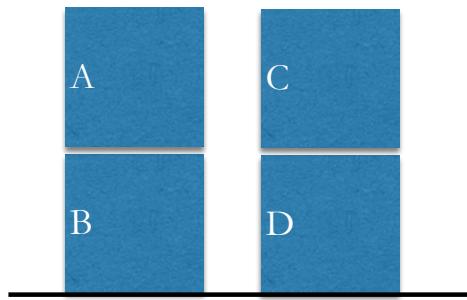


## ESEMPIO DI ALGORITMO STRIPS





- STATO INIZIALE E GOAL ESPRESSI MEDIANTE FORMULE:



```
(setq ab-cd
  '((block a)
    (block b)
    (block c)
    (block d)
    (clear a)
    (on a b)
    (on b table)
    (clear c)
    (on c d)
    (on d table))))
```



- AZIONI DESCritte MEDIANTE CAMBIAMENTI DI STATO
1. **PRECONDIZIONI:** FORMULE DA VERIFICARE PER ESEGUIRE L'AZIONE
  2. **ADD-LIST:** FORMULE CHE DIVENTANO VERE DOPO L'ESECUZIONE DELL'AZIONE
  3. **DELETE-LIST:** FORMULE NON PIÙ VERE DOPO L'ESECUZIONE DELL'AZIONE
  4. **FILTER:** FORMULE CHE PERMETTONO IL FILTRAGGIO DI AZIONI



## MUOVI IL BLOCCO SUL TAVOLO

---

```
(strips-op move-block-to-table
  :preconditions  ( (clear ?x)
                      (on ?x ?y) )
  :filter         ( (block ?x)
                      (block ?y)
                      (:not-equal ?x ?y) )
  :add-list       ( (on ?x table)
                      (clear ?y) )
  :delete-list    ( (on ?x ?y) )
  :command-string "move ?x from ?y to the table"
  :variables      (?x ?y)
)
```



## MUOVI IL BLOCCO DAL TAVOLO A

---

```
(strips-op move-block-from-table
  :filter      ( (block ?x)
                  (block ?y)
                  (:not-equal ?x ?y) )
  :preconditions  ( (on ?x table)
                    (clear ?x)
                    (clear ?y) )
  :add-list      ( (on ?x ?y) )
  :delete-list   ( (clear ?y) )
  :command-string "move ?x from the table to ?y"
  :variables     (?x ?y)
)
```



```
(strips-op move-block-to-block
  :filter      ( (block ?x)
                  (block ?y)
                  (block ?z)
                  (:not-equal ?x ?y)
                  (:not-equal ?z ?y)
                  (:not-equal ?x ?z) )
  :preconditions ( (on ?x ?y)
                  (clear ?x)
                  (clear ?z) )
  :add-list      ( (on ?x ?z)
                  (clear ?y) )
  :delete-list   ( (on ?x ?y)
                  (clear ?z) )
  :command-string "move ?x from ?y to ?z"
  :variables     (?x ?y ?z)
)
```

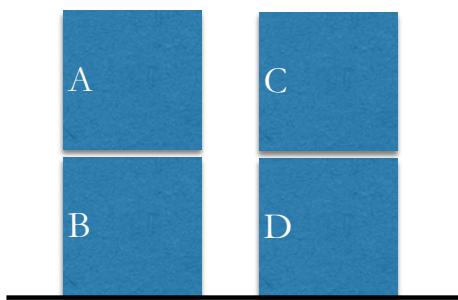


- Input: stato  $s$ ; goal  $g$   
Output: sequenza di azioni per trasformare  $s$  in  $g$
- Calcola i predicati da verificare  $d = g - s$
- Se  $d$  è vuoto, termina e stampa la sequenza di azioni
- Scegli l'azione  $a$  per cui la *add-list* contiene il maggior numero di formule contenute in  $d$
- Chiama ricorsivamente *Strips* sulle precondizioni di  $a$
- Calcola il nuovo stato  $s$  ottenuto applicando  $a$
- Chiama ricorsivamente *Strips* sul nuovo stato  $s$

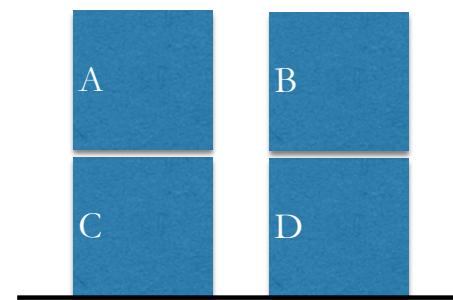


## ESEMPIO

```
(strips  
ab-cd  
' ((on a c)  
    (on b d)))
```



stato iniziale



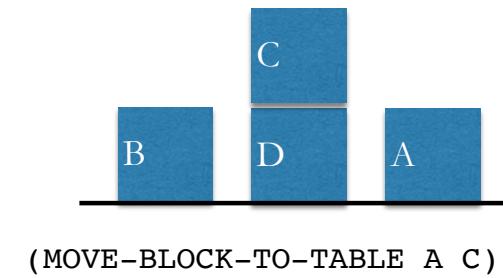
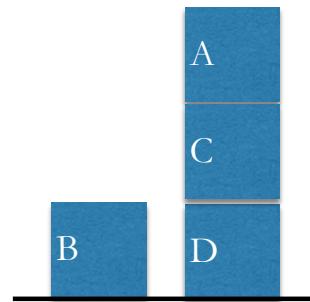
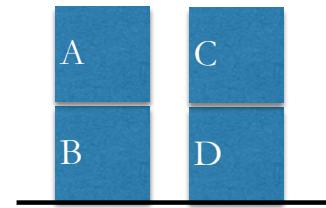
goal

PROF. ROSARIO SORBELLO

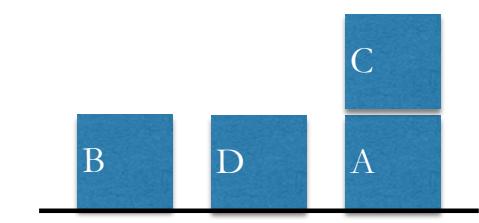


**ROBOTICS LAB**

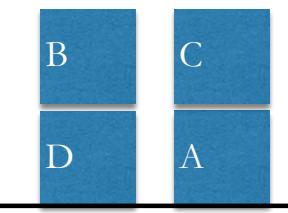
## ESEMPIO



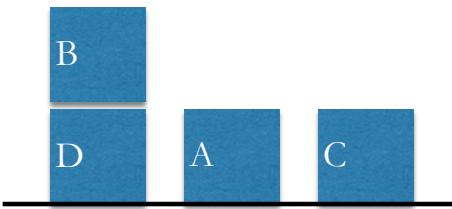
(MOVE-BLOCK-TO-BLOCK A B C)



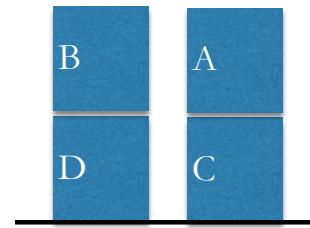
(MOVE-BLOCK-TO-BLOCK C D A)



(MOVE-BLOCK-FROM-TABLE B D)



(MOVE-BLOCK-TO-TABLE C A)

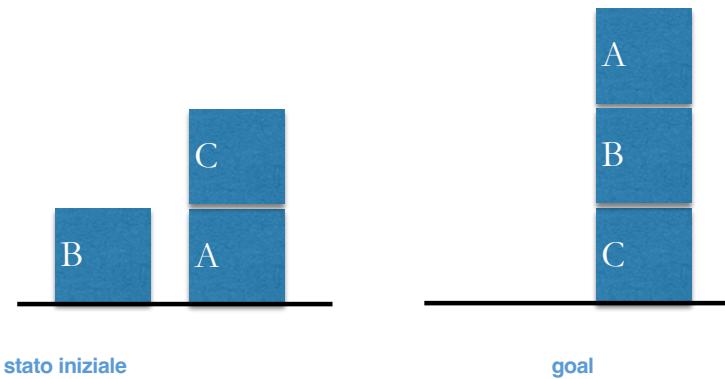


(MOVE-BLOCK-FROM-TABLE A C)

PROF. ROSARIO SORBELLO



## ANOMALIA DI SUSSMAN

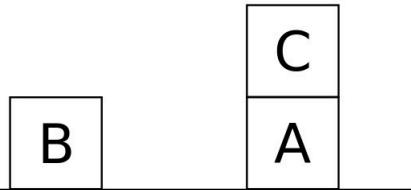


```
(strips
  '((block a)
    (block b)
    (block c)
    (on b table)
    (clear b)
    (on a table)
    (on c a)
    (clear c))
   '((on b c)(on a b)))
```

- PROBLEMA DEGLI ALGORITMI NON INTERLACCIATI COME STRIPS CHE DIVIDONO UN PROBLEMA COME CATENA DI AND E IN TANTI SUB-GOALS
- IL GOAL È COMPOSTO DA DUE SUBGOAL (ON B C) E (ON A B) CHE SI INTERFERISCONO A VICENDA



## ANOMALIA DI SUSSMAN



Tuttavia, i pianificatori non interacciati in genere separano l'obiettivo (pila A sopra B sopra C) in obiettivi secondari, come:

1. prendi A in cima a B
2. prendi B in cima a C

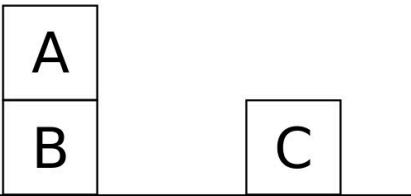
Supponiamo che il pianificatore inizi perseguiendo l'obiettivo 1. La soluzione semplice è spostare C fuori strada, quindi spostare A sopra B. Ma mentre questa sequenza raggiunge l'obiettivo 1, l'agente non può ora perseguire l'obiettivo 2 senza annullare l'obiettivo 1, poiché entrambi A e B deve essere spostato sopra C:

PROF. ROSARIO SORBELLO

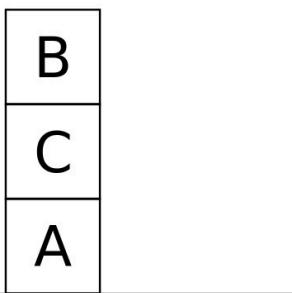


**ROBOTICS LAB**

## ANOMALIA DI SUSSMAN



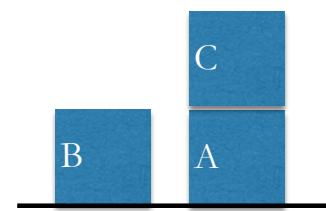
Se invece il pianificatore inizia con l'Obiettivo 2, la soluzione più efficiente è muovere B. Ma ancora una volta, il pianificatore non può perseguire l'Obiettivo 1 senza annullare l'Obiettivo 2:



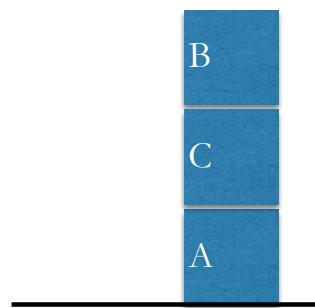
PROF. ROSARIO SORBELLO



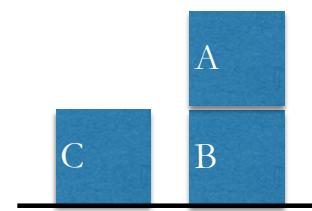
## ANOMALIA DI SUSSMAN



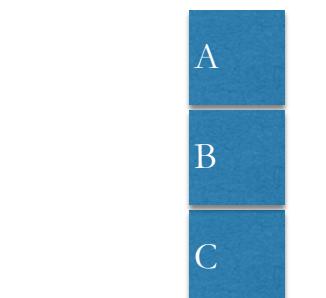
stato iniziale



(on b c)



(on a b)



goal

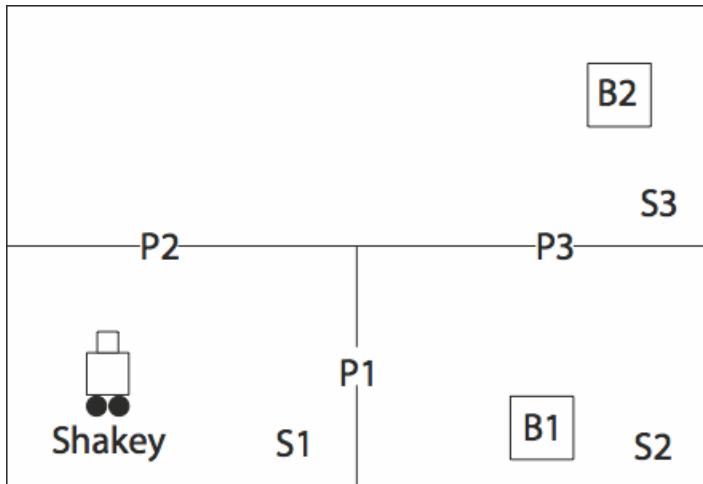
```
((MOVE-BLOCK-FROM-TABLE B C)
(MOVE-BLOCK-TO-TABLE B C)
(MOVE-BLOCK-TO-TABLE C A)
(MOVE-BLOCK-FROM-TABLE A B)
(MOVE-BLOCK-TO-TABLE A B)
(MOVE-BLOCK-FROM-TABLE B C)
(MOVE-BLOCK-FROM-TABLE A B))
```

PROF. ROSARIO SORBELLO



ROBOTICS LAB

## SHAKEY ROBOT



```
(setq mondo
      '((stanza s1)
        (stanza s2)
        (stanza s3)
        (porta p1)
        (porta p2)
        (porta p3)
        (passaggio p1 s1 s2)
        (passaggio p1 s2 s1)
        (passaggio p2 s1 s3)
        (passaggio p2 s3 s1)
        (passaggio p3 s2 s3)
        (passaggio p3 s3 s2)))

(setq scatole
      '((scatola b1)
        (scatola b2)))

(setq iniziale
      '((in shakey s1)
        (in b1 s2)
        (in b2 s3)))

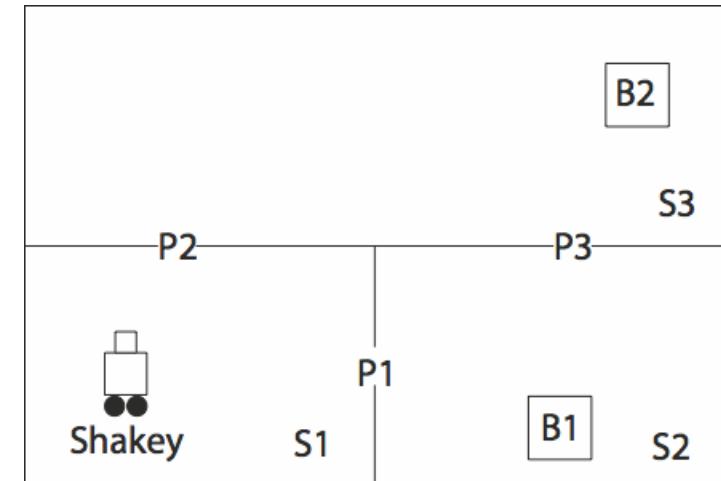
(setq sit1
      (concatenate 'list
                  mondo scatole iniziale))
```

PROF. ROSARIO SORBELLO



## AZIONI DEL ROBOT SHAKEY

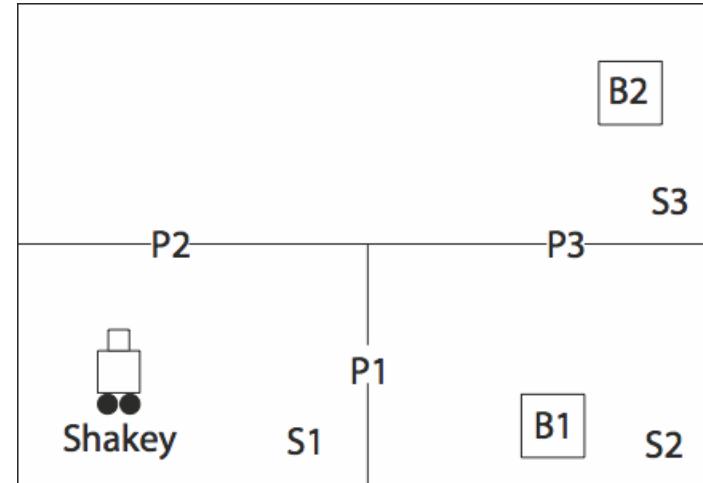
```
(strips-op andare
  :filter ((stanza ?x)
            (stanza ?y)
            (:not-equal ?x ?y)
            (passaggio ?p ?x ?y))
  :preconditions ((in shakey ?x))
  :add-list ((in shakey ?y))
  :delete-list ((in shakey ?x))
  :command-string "muoviti dalla stanza ?x alla stanza ?y"
  :variables (?x ?y))
```





## AZIONI DEL ROBOT SHAKEY

```
(strips-op spingere
  :filter ((scatola ?b)
            (stanza ?x)
            (stanza ?y)
            (:not-equal ?x ?y)
            (passaggio ?p ?x ?y))
  :preconditions ((in ?b ?x)
                  (in shakey ?x))
  :add-list ((in ?b ?y) (in shakey ?y))
  :delete-list ((in ?b ?x) (in shakey ?x))
  :command-string "spingi la scatola ?b dalla stanza ?x alla stanza ?y"
  :variables (?b ?x ?y))
```

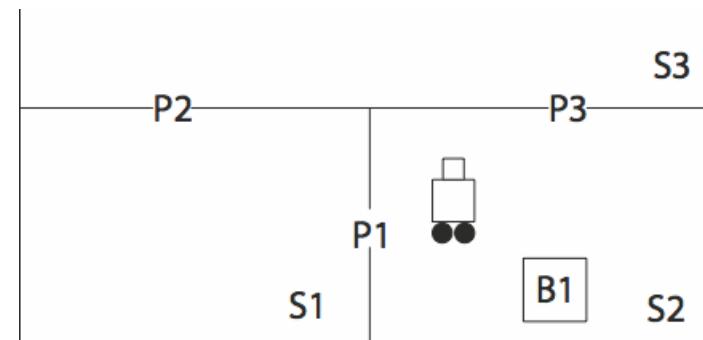
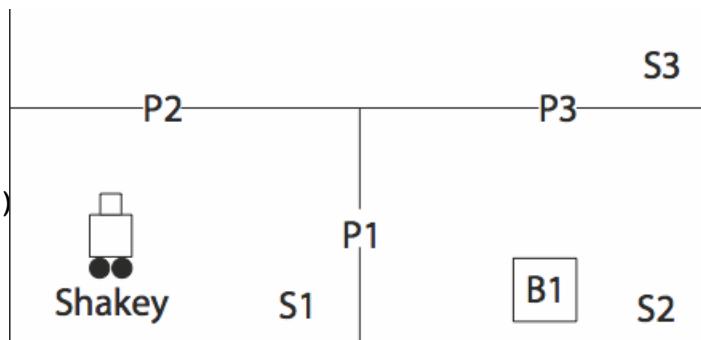




**ROE**(strips-op andare

```
:filter ((stanza ?x)
         (stanza ?y)
         (:not-equal ?x ?y)
         (passaggio ?p ?x ?y))
:preconditions ((in shakey ?x))
:add-list ((in shakey ?y))
:delete-list ((in shakey ?x))
:command-string "muoviti dalla stanza ?x alla stanza ?y"
:variables (?x ?y))
```

```
anza s1)
anza s2)
anza s3)
rta p1)
rta p2)
rta p3)
ssaggio p1 s1 s2,
ssaggio p1 s2 s1)
ssaggio p2 s1 s3)
ssaggio p2 s3 s1)
ssaggio p3 s2 s3)
ssaggio p3 s3 s2))
```



```
le
atola b1)
atola b2)))
```

```
ale
shakey s1)
b1 s2)
b2 s3)))
```

```
(strips sit1 '((in shakey s2)))
((ANDARE S1 S2))
```

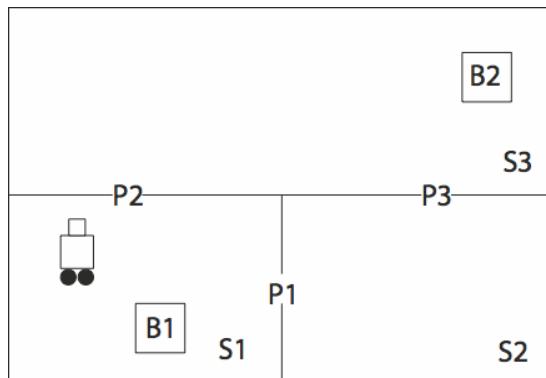
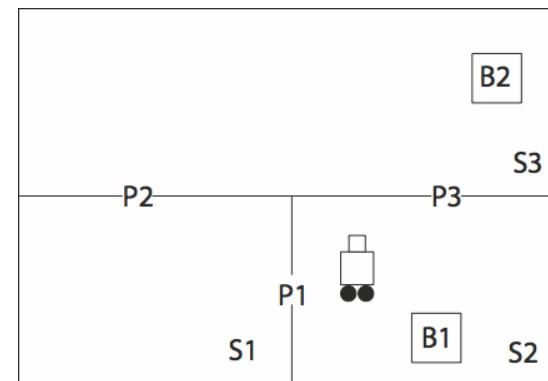
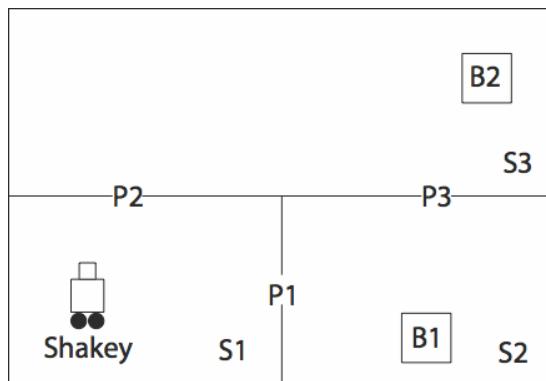
```
oncatenate 'list
mondo scatole iniziale))
```

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## TEST 1



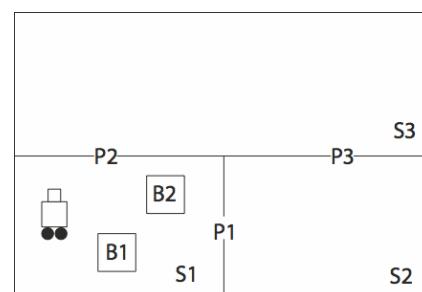
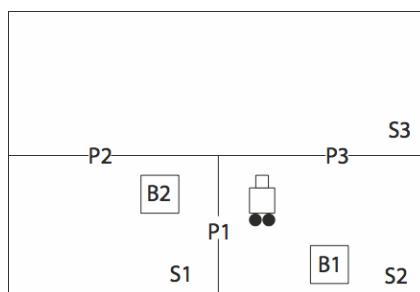
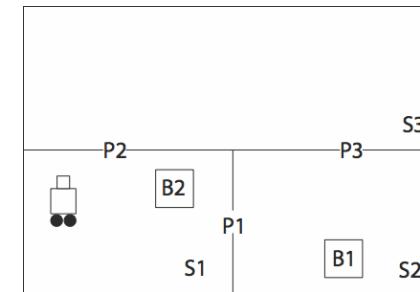
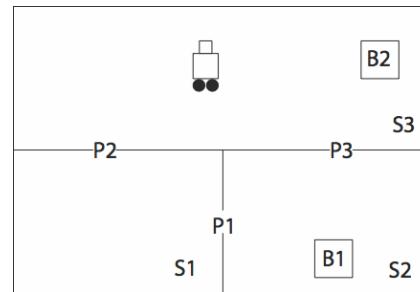
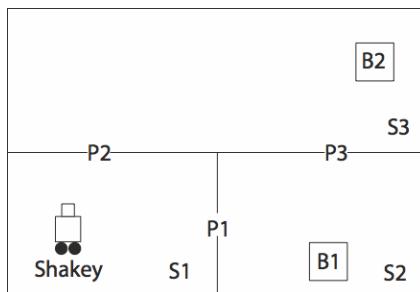
```
(strips sit1 '((in b1 s1)))  
((ANDARE S1 S2)  
 (SPINGERE B1 S2 S1))
```

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## TEST 2



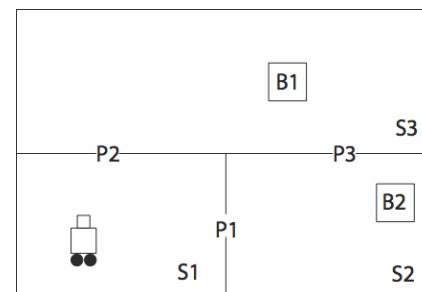
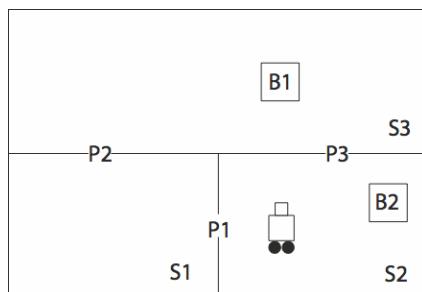
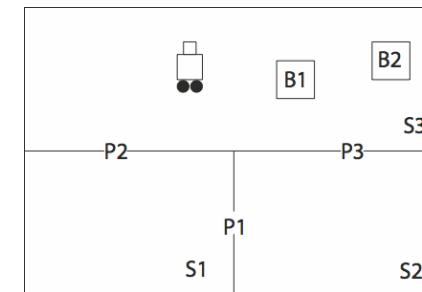
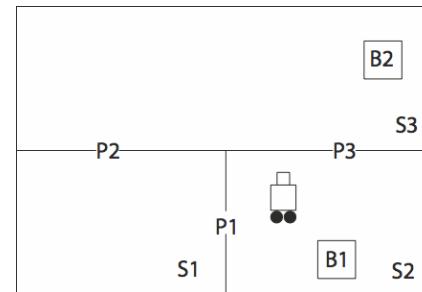
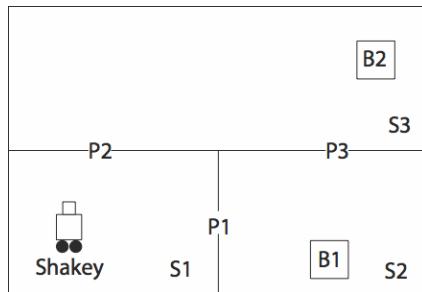
```
(strips sit1 '((in b1 s1)
                (in b2 s1)
                (in shakey s1)))
((ANDARE S1 S3)
 (SPINGERE B2 S3 S1)
 (ANDARE S1 S2)
 (SPINGERE B1 S2 S1))
```

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## TEST 3

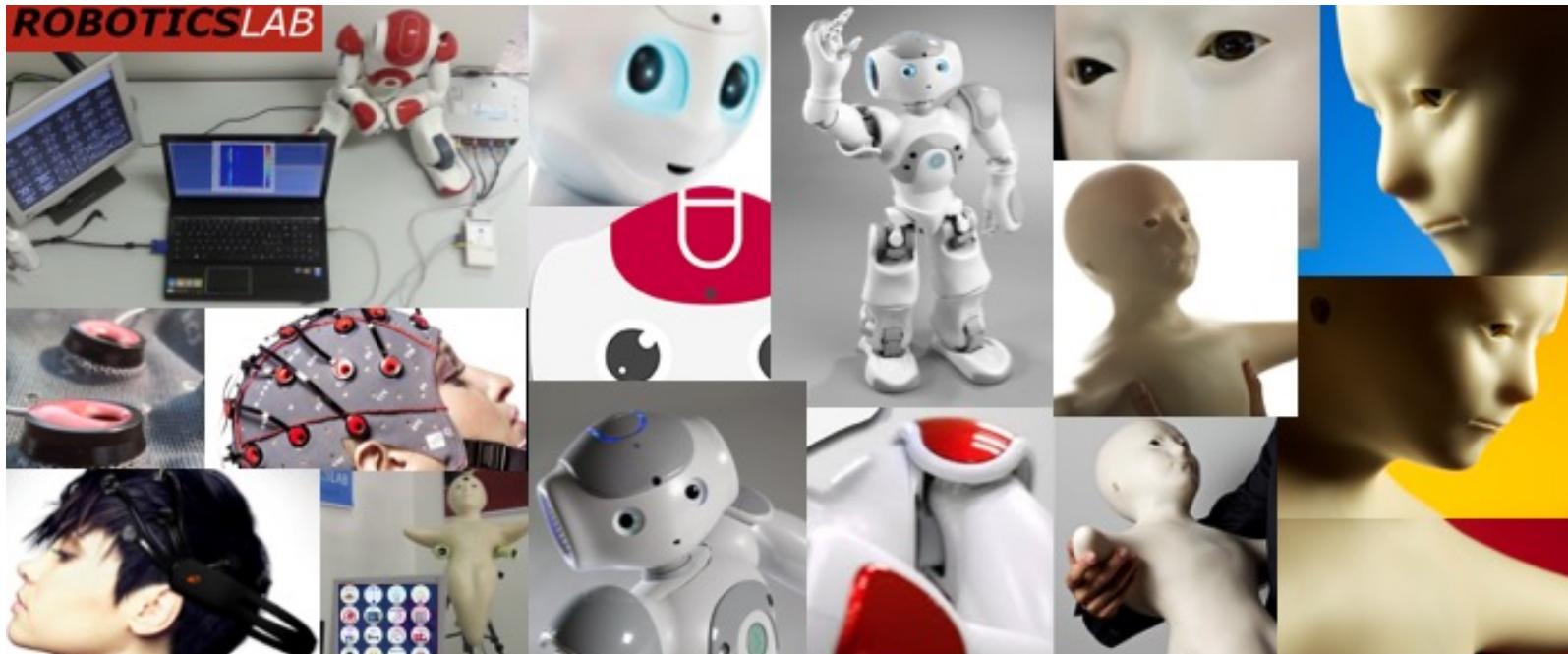


```
(strips sit1 '((in b1 s3)
                (in b2 s2)
                (in shakey s1)))
 ((ANDARE S1 S2)
  (SPINGERE B1 S2 S3)
  (SPINGERE B2 S3 S2)
  (ANDARE S2 S1))
```

PROF. ROSARIO SORBELLO



OSAKA UNIVERSITY



ATR

g-tec

## ROBOTICA PROBABILISTICA

PROF. ROSARIO SORBELLO

8° LEZIONE – PALERMO 2 NOVEMBRE 2022



- **IDEA DI BASE:** RAPPRESENTAZIONE ESPLICITA DELL'INCERTEZZA DOVUTA AI SENSORI E AGLI ATTUATORI ATTRAVERSO LA TEORIA DELLA PROBABILITÀ
- **PERCEZIONE:** STIMA DELLO STATO DELL'AMBIENTE
- **AZIONE:** STIMA DELLE TRANSIZIONI DI STATO DELL'AMBIENTE



### Rappresentazione dell'incertezza usando la probabilità

**Percezione** = stima dello stato (problema stima)

**Azione** = massimizzazione utilità (teoria decisioni)

Percezione: mappa misure dei sensori nella rappresentazione interna dello stato del robot e dell'ambiente (stato).

Sensori sono rumorosi, l'ambiente è parzialmente osservabile, non prevedibile, dinamico, modello inaccurato etc.

**Belief** = Probabilità dello stato date azioni ed osservazioni



**ROBOTICS LAB**

## ASSIOMI DELLA TEORIA DELLA PROBABILITÀ

---

- $\Pr(A)$  DENOTA LA PROBABILITÀ CHE LA PROPOSIZIONE A SIA VERA

### ASSIOMI

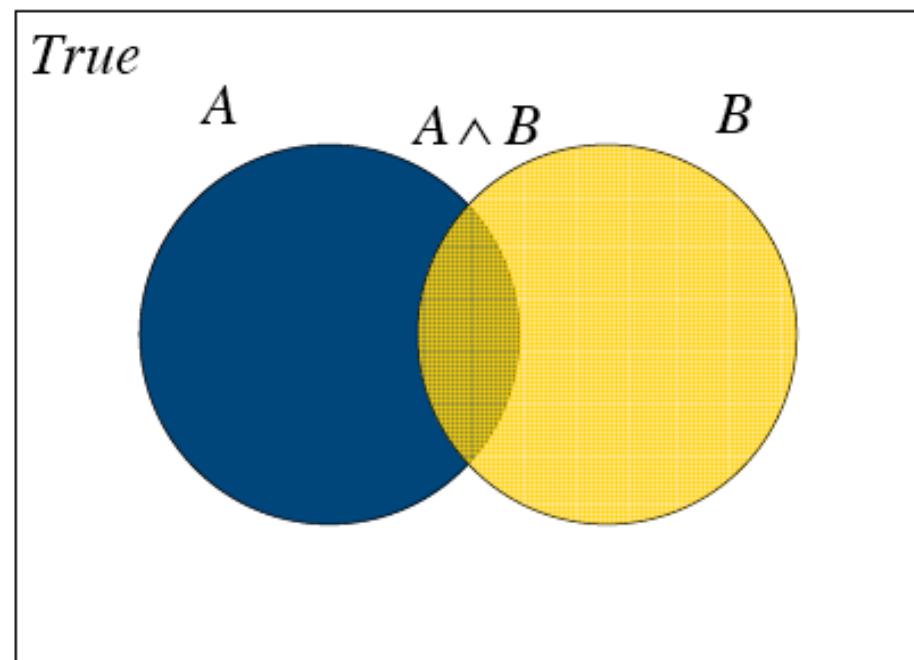
1. POSITIVITA'  $0 \leq \Pr(A) \leq 1$
2. CERTEZZA  $\Pr(\text{True}) = 1$        $\Pr(\text{False}) = 0$
3. UNIONE  $\Pr(A \vee B) = \Pr(A) + \Pr(B) - \Pr(A \wedge B)$

PROF. ROSARIO SORBELLO



ROBOTICS LAB

## ASSIOMA UNIONE



$$\Pr(A \vee B) = \Pr(A) + \Pr(B) - \Pr(A \wedge B)$$

PROF. ROSARIO SORBELLO



CONSEGUENZE ASSIOMI PROBABILITÀ

$$\Pr(A \vee \neg A) = \Pr(A) + \Pr(\neg A) - \Pr(A \wedge \neg A)$$

$$\Pr(True) = \Pr(A) + \Pr(\neg A) - \Pr(False)$$

$$1 = \Pr(A) + \Pr(\neg A)$$

$$\Pr(\neg A) = 1 - \Pr(A)$$

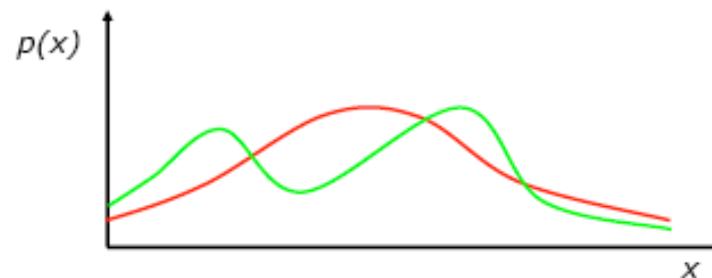


- X DENOTA UNA VARIABILE **CASUALE (ALEATORIA)**
- X PUÒ ASSUMERE UN NUMERO FINITO DI VALORI  
in  $\{x_1, x_2, \dots, x_n\}$
- $P(X=x_i)$  o  $P(x_i)$  È LA **PROBABILITÀ** CHE LA VARIABILE CASUALE X ASSUMA IL VALORE  $x_i$
- $P(\cdot)$  è la funzione **massa di probabilità**
- Es.  $P(\text{Stanza}) = \{0.7, 0.2, 0.08, 0.02\}$



- X assume valori nel continuo
- $p(x)$  è la **funzione di densità di probabilità (pdf)**

$$P(x \in (a, b)) = \int_a^b p(x) dx$$



PROF. ROSARIO SORBELLO



- **PROBABILITÀ CONGIUNTA DI 2 VARIABILI RANDOMICHE:**

$$P(X = x, Y = y) = P(x, y)$$

- SE X E Y SONO INDIPENDENTI ALLORA:

$$P(x, y) = P(x)P(y)$$

- $P(X | Y)$  È LA PROBABILITÀ **CONDIZIONALE** DI X DATO Y:

$$P(x | y) = \frac{P(x, y)}{P(y)}$$

$$P(x, y) = P(x | y)P(y)$$

- SE X E Y SONO INDIPENDENTI ALLORA:

$$P(x | y) = P(x)$$

PROF. ROSARIO SORBELLO



## LEGGE DELLA PROBABILITA' TOTALE

CASO DISCRETO

$$\sum_x P(x) = 1$$

$$P(x) = \sum_y P(x,y)$$

$$P(x) = \sum_y P(x|y)P(y)$$

CASO CONTINUO

$$\int p(x)dx = 1$$

$$p(x) = \int p(x,y)dy$$

$$p(x) = \int p(x|y)p(y)dy$$

PROF. ROSARIO SORBELLO



## TEOREMA DI BAYES

$$p(A_i|B) = \frac{p(A_i)p(B|A_i)}{p(B)} = \frac{p(A_i)p(B|A_i)}{\sum_{i=1}^N p(A_i)p(B|A_i)}$$

Il teorema di Bayes consente quindi di determinare a posteriori, la probabilità che una specifica causa  $A_i$  abbia determinato un effetto  $B$  conoscendo a priori la probabilità che si verifichi la causa stessa  $A_i$  e la probabilità che si verifichi l'effetto  $B$  quando è presente la causa  $A_i$ . In breve

$$p(A_i|B) = \frac{p(A_i)p(B|A_i)}{p(B)} = \frac{p(A_i)p(B|A_i)}{\sum_{i=1}^N p(A_i)p(B|A_i)}$$

Probabilità a posteriori di  $A_i$

Probabilità a priori di  $A_i$

Probabilità che la causa  $A_i$  abbia determinato l'effetto  $B$  (causa dato l'effetto)

Probabilità l'effetto  $B$  si verifica sapendo che l'evento  $A_i$  sia verificato (effetto data la causa)



## FORMULA DI BAYES

CONNETTE LA PROBABILITA' CONDIZIONALE  $P(x|y)$  CON IL SUO INVERSO  $P(y|x)$

$$P(x,y) = P(x|y)P(y) = P(y|x)P(x)$$
$$\Rightarrow \quad \begin{array}{c} \text{PROBABILITA'} \\ \downarrow \\ P(x|y) = \frac{P(y|x)P(x)}{P(y)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}} \end{array} \quad \begin{array}{c} \text{PRIORITA'} \\ \downarrow \\ \text{EVIDENZA O FATTO} \end{array}$$

SE X E' LA QUANTITA' CHE VOGLIAMO INFERIRE DA Y:

1.  $P(x)$  LA DISTRIBUZIONE DI PROBABILITA' PRIORITARIA
2.  $y$  I NOSTRI DATI (MISURA DEI VALORI SENSORIALI)
3.  $P(y)$  LA PROBABILITA' CHE SI VERIFICHINO CERTI DATI
4.  $P(x|y)$  DISTRIBUZIONE DI PROBABILITA' POSTERIORE

PROF. ROSARIO SORBELLO

NORMALIZZAZIONE

---

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} = \eta P(y|x)P(x)$$

$$\eta = P(y)^{-1} = \frac{1}{\sum_x P(y|x)P(x)}$$

$P(y)$  NON DIPENDE DA  $X$ , QUINDI IL FATTORE  $P(y)^{-1}$  DELL'EQUAZIONE VIENE DENOMINATO VALORE DI NORMALIZZAZIONE E VIENE DENOTATO CON LA LETTERA GRECA  $\eta$

$$P(x|y) = \frac{P(y|x)P(x)}{\boxed{P(y)}}$$

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## REGOLA DI BAYES CON CONOSCENZA IN BACKGROUND

---

$$P(x|y,z) = \frac{P(y|x,z)P(x|z)}{P(y|z)}$$

PROF. ROSARIO SORBELLO



- DALLA LEGGE DELLA PROBABILITÀ TOTALE

$$P(x) = \int P(x, z) dz$$

$$P(x) = \int P(x | z) P(z) dz$$

$$P(x | y) = \int P(x | y, z) P(z) dz$$

AAA

INDIPENDENZA CONDIZIONALE

---

$$P(x|y,z) = \frac{P(y|x,z)P(x|z)}{P(y|z)}$$

- EQUIVALENTE A:  $P(x,y|z) = P(x|z)P(y|z)$

- OPPURE A:

$$P(x|z) = P(x|z,y)$$

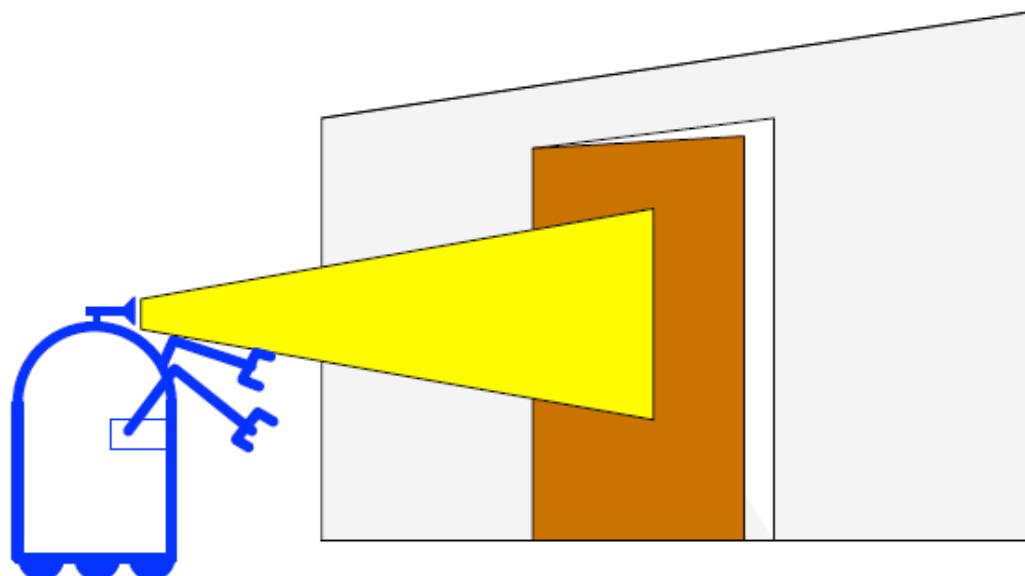
UNA VARIABILE **y** NON PORTA INFORMAZIONI SULLA VARIABILE **x** SE UN'ALTRA VARIABILE **z** E' NOTA.

$$P(y|z) = P(y|z,x)$$



## ESEMPIO DI STIMA DELLO STATO

- IL ROBOT OTTIENE LA MISURA Z. Z MISURA SENSORIALE PER CAPIRE SE LA PORTA SIA APERTA. **QUANTO È  $P(\text{OPEN} | Z)$ ?**
- QUAL'È LA PROBABILITÀ CHE SIA VERAMENTE APERTA DATA LA MISURA DEL VALORE Z



PROF. ROSARIO SORBELLO



- $P(OPEN | Z)$  È PROBABILITÀ DIAGNOSTICO
- $P(Z | OPEN)$  È PROBABILITÀ CAUSALE
  
- LA REGOLA DI BAYES PERMETTE DI USARE PROBABILITÀ CAUSALE PER OTTENERE PROBABILITÀ DIAGNOSTICA

$$P(open | z) = \frac{P(z | open)P(open)}{P(z)}$$

PROF. ROSARIO SORBELLO

ESEMPIO

---

$$P(z | open) = 0.6$$

$$P(z | \neg open) = 0.3$$

$P(open) = P(\neg open) = 0.5$  Probabilità iniziale uguale per i 2 stati (open,close)

$$P(open | z) = \frac{P(z | open)P(open)}{P(z | open)P(open) + P(z | \neg open)P(\neg open)}$$

$$P(open | z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = 0.67$$

La misura di  $z$  aumenta la probabilità che la porta sia aperta



## COMBINAZIONE DI EVIDENZA

---

- Il robot ottiene una nuova osservazione  $z_2$
- Come si integra questa informazione?
- In generale, come stimare  $P(x | z_1, z_2, \dots, z_n)$ ?



$$P(x | z_1, z_2, \dots, z_n) = \frac{P(z_n | x, z_1, z_2, \dots, z_{n-1}) P(x | z_1, z_2, \dots, z_{n-1})}{P(z_n | z_1, z_2, \dots, z_{n-1})}$$

**Assunzione di Markov:** lo stato attuale  $x$  riassume tutta la storia passata del sistema, quindi  $z_n$  è indipendente da  $z_1, z_2, \dots, z_{n-1}$  ma dipende solo da  $x$ .

$$\begin{aligned} P(x | z_1, z_2, \dots, z_n) &= \frac{P(z_n | x) P(x | z_1, z_2, \dots, z_{n-1})}{P(z_n | z_1, z_2, \dots, z_{n-1})} \\ &= \eta P(z_n | x) P(x | z_1, z_2, \dots, z_{n-1}) \\ &= \eta \prod_{i=1 \dots n} P(z_i | x) P(x) \end{aligned}$$

PROF. ROSARIO SORBELLO

ESEMPIO: 2° MISURA

---

$$P(z_2 | open) = 0.5$$

$$P(z_2 | \neg open) = 0.6$$

$$P(open | z_1) = 0.67$$

$$P(open | z_2, z_1) = \frac{P(z_2 | open)P(open | z_1)}{P(z_2 | open)P(open | z_1) + P(z_2 | \neg open)P(\neg open | z_1)}$$

$$P(open | z_2, z_1) = 0.625$$

La nuova misura  $z_2$  abbassa la probabilità che la porta sia aperta



- IL MONDO È **DINAMICO** A CAUSA DI:
  - AZIONI EFFETTUATE DAL **ROBOT**
  - AZIONI EFFETTUATE DA **ALTRI AGENTI**
  - **TEMPO** CHE PASSA
- COME TENERE CONTO DELLE **AZIONI**?



- IL ROBOT MUOVE LE **RUOTE** PER SPOSTARSI
- IL ROBOTI USA IL **MANIPOLATORE** PER AFFERRARE DEGLI OGGETTI
- L'**AMBIENTE** SI EVOLVE NEL TEMPO
- LE AZIONI NON SONO MAI EFFETTUATE CON **CERTEZZA ASSOLUTA**
- IN CONTRASTO CON LE MISURE, LE AZIONI **AUMENTANO L'INCERTEZZA**



- Per incorporare il risultato **dell'azione**  $u$  nella corrente credenza  $x$  (**belief**) dello stato usiamo la pdf condizionale:

$$P(x|x', u)$$

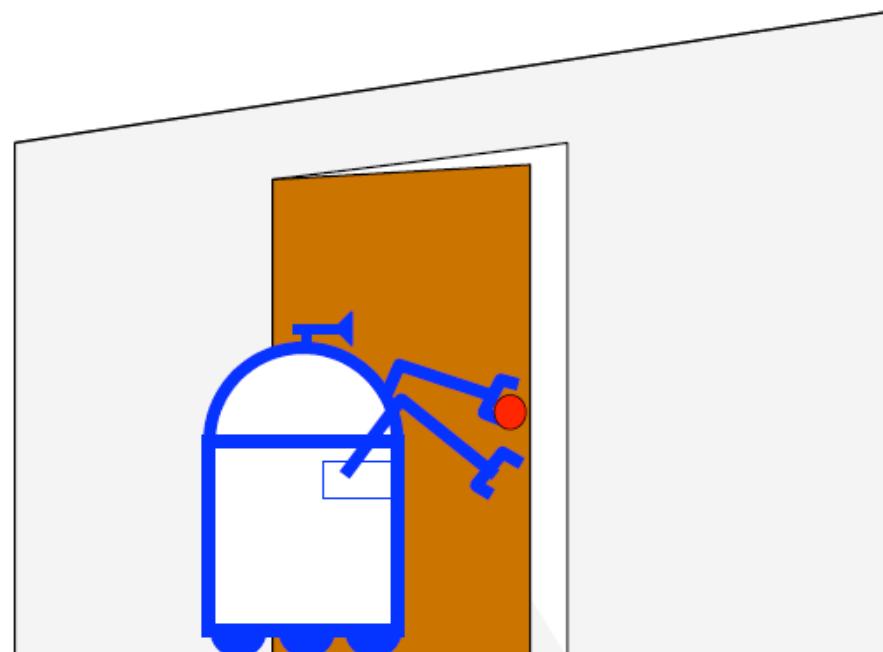
- Questo termine specifica la **pdf** della **transizione** dallo stato  $x'$  a  $x$  per effetto della **esecuzione** dell'azione  $u$



**ROBOTICS LAB**

## ESEMPIO: CHIUSURA DELLA PORTA

---

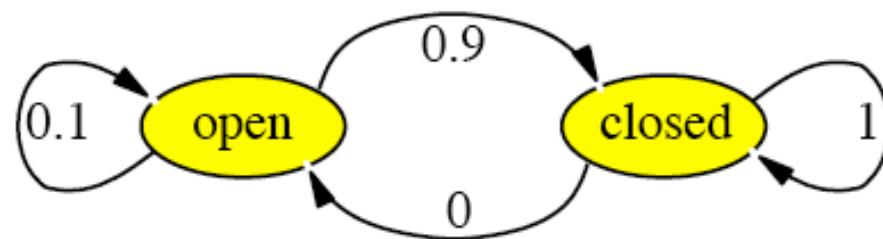


PROF. ROSARIO SORBELLO



## TRANSIZIONI DI STATO

- $P(x|x', u)$  dove  $u = \text{"close door"}$



- SE LA PORTA È APERTA, L’AZIONE “CLOSE DOOR” HA SUCCESSO NEL 90% DEI CASI



**ROBOTICS LAB**

## INTEGRAZIONE DEL RISULTATO DELLE AZIONI

---

$$P(x|u) = \int P(x|x',u)P(x')dx'$$
$$P(x|u) = \sum P(x|x',u)P(x')$$

PROF. ROSARIO SORBELLO



**ROBOTICS LAB**

## ESEMPIO

---

$$\begin{aligned} P(\text{closed} | u) &= \sum P(\text{closed} | x', u) P(x') \\ &= P(\text{closed} | \text{open}, u) P(\text{open}) + P(\text{closed} | \text{closed}, u) P(\text{closed}) \\ &= 0.9375 \end{aligned}$$

$$\begin{aligned} P(\text{open} | u) &= \sum P(\text{open} | x', u) P(x') \\ &= P(\text{open} | \text{open}, u) P(\text{open}) + P(\text{open} | \text{closed}, u) P(\text{closed}) \\ &= 0.0625 \\ &= 1 - P(\text{closed} | u) \end{aligned}$$

PROF. ROSARIO SORBELLO



- Dati:

- La catena di osservazioni z e di azioni u:  $d_t = \{u_1, z_1, u_2, z_2, \dots, u_t, z_t\}$

- Il modello del sensore:  $P(z|x)$

- Il modello dell'attuatore:  $P(x|x', u)$

- La probabilità a priori dello stato:  $P(x)$

- Si vuole ottenere:

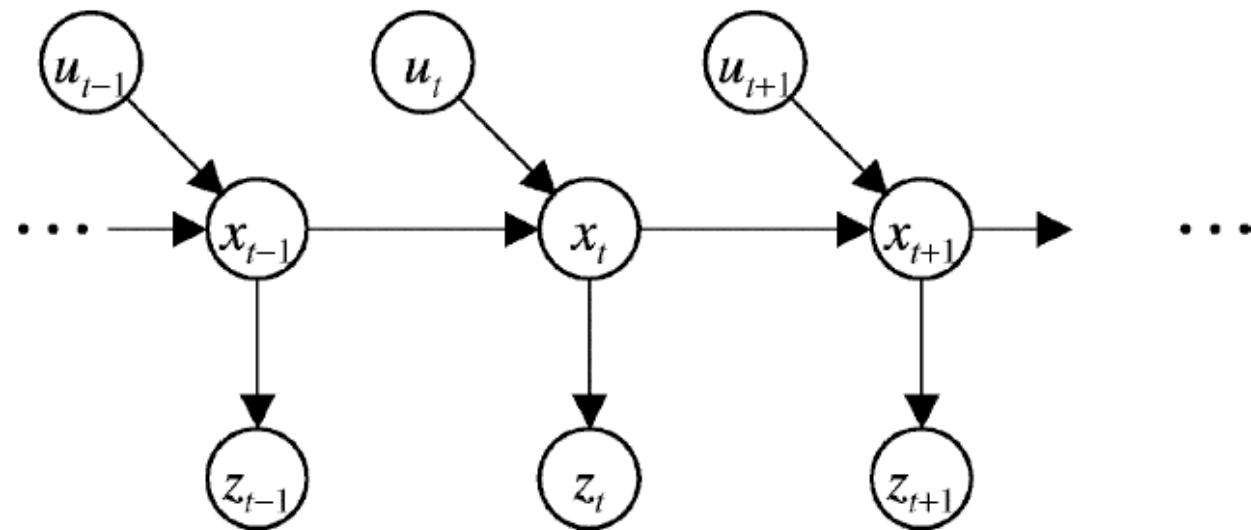
- La stima X dello stato del sistema dinamico.

- La stima dello stato è il Belief:

$$Bel(x_t) = P(x_t | u_1, z_1, u_2, z_2, \dots, u_t, z_t)$$

PROF. ROSARIO SORBELLO

## ASSUNZIONE DI MARKOV



$$p(z_t | x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t | x_t)$$

$$p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t | x_{t-1}, u_t)$$

PROF. ROSARIO SORBELLO

FILTRI DI BAYES

---

$$\begin{aligned} \boxed{Bel(x_t)} &= p(x_t | u_1, z_1, \dots, u_t, z_t) \\ &= \eta p(z_t | x_t, u_1, z_1, \dots, u_t) p(x_t | u_1, z_1, \dots, u_t) && \text{Bayes} \\ &= \eta p(z_t | x_t) p(x_t | u_1, z_1, \dots, u_t) && \text{Markov} \\ &= \eta p(z_t | x_t) \int p(x_t | x_{t-1}, u_1, z_1, \dots, u_t) p(x_{t-1} | u_1, z_1, \dots, u_t) dx_{t-1} && \text{Prob. totale} \\ &= \eta p(z_t | x_t) \int p(x_t | x_{t-1}, u_t) p(x_{t-1} | u_1, z_1, \dots, u_t) dx_{t-1} && \text{Markov} \\ &= \eta p(z_t | x_t) \int p(x_t | x_{t-1}, u_t) p(x_{t-1} | u_1, z_1, \dots, z_{t-1}) dx_{t-1} && \text{Markov} \\ &\boxed{= \eta p(z_t | x_t) \int p(x_t | x_{t-1}, u_t) Bel(x_{t-1}) dx_{t-1}} \end{aligned}$$



**ROBOTICS LAB**

## FILTRI DI BAYES

---

- FILTRO DI KALMAN
- PARTICLE FILTER
- MODELLI MARKOVIANI
- RETI BAYESIANE DINAMICHE
- .....

PROF. ROSARIO SORBELLO



- LA **REGOLA DI BAYES** PERMETTE DI CALCOLARE LA PROBABILITÀ DELLO STATO, DIFFICILE DA OTTENERE IN ALTRO MODO
- L'**AGGIORNAMENTO BAYESIANO RICORSIVO** BASATO SULLA **ASSUNZIONE DI MARKOV** PERMETTE DI COMBINARE EVIDENZE
- I **FILTRI DI BAYES** PERMETTONO DI STIMARE LO STATO DI UN SISTEMA DINAMICO