Lecture 19: Developing Autonomy for Robots in Teams

Brett Browning

Autonomous Robots 16-200

Carnegie Mellon University in Qatar

Overview

- Logistics
- The plan
- RoboCup research environment
- My research

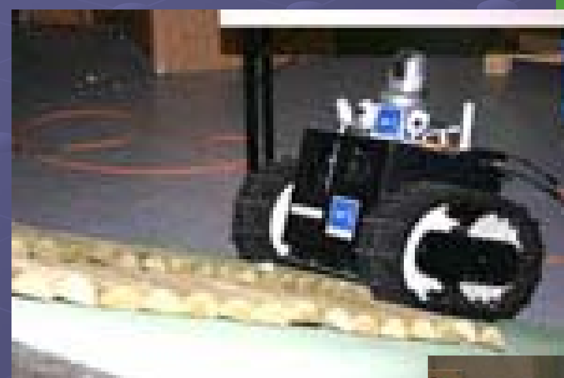
Logistics

- Weekly project meetings
  - You give a 5 minute update on your progress
  - To be held in Thursday Lab slot
  - Additional help also available on Thursday
- Homework #4 is due Thursday at 10.30am
- Change to the syllabus
- Please take care of the batteries!!!

The Plan

- Today: RoboCup and Brett's research
- Wednesday & Next week: Manipulation
- Homework #5
- Research projects
  - Weekly meetings during lab session
  - Demonstration, paper, presentation, poster
  - (Remember, start now!)

RoboCup Robot Soccer



RoboCup Robot Soccer

- Challenge to researchers to improve robot intelligence through friendly competition
  - "By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team", *www.robocup.org*
- First competition in 1997, Nagoya Japan

Small Size Robot League

CMDragons [Browning, Bruce,Bowling, Veloso *et al.* 03]

Legged League

Mid-Size League

Segway Soccer

Segway Soccer [Browning *et al.* 04]

Humanoid League

RoboCup Details

- Annual international competitions
  - Next one in RoboCup 2006 in Bremen, Germany
  - Large and growing larger (2,000 competitors in 2006)
- Teams compete in games of soccer
  - Human referee commands translated by computer
  - Fully autonomous during game
- Research communicated via
  - Technical reports, papers, symposium
  - Code releases

Common Challenges

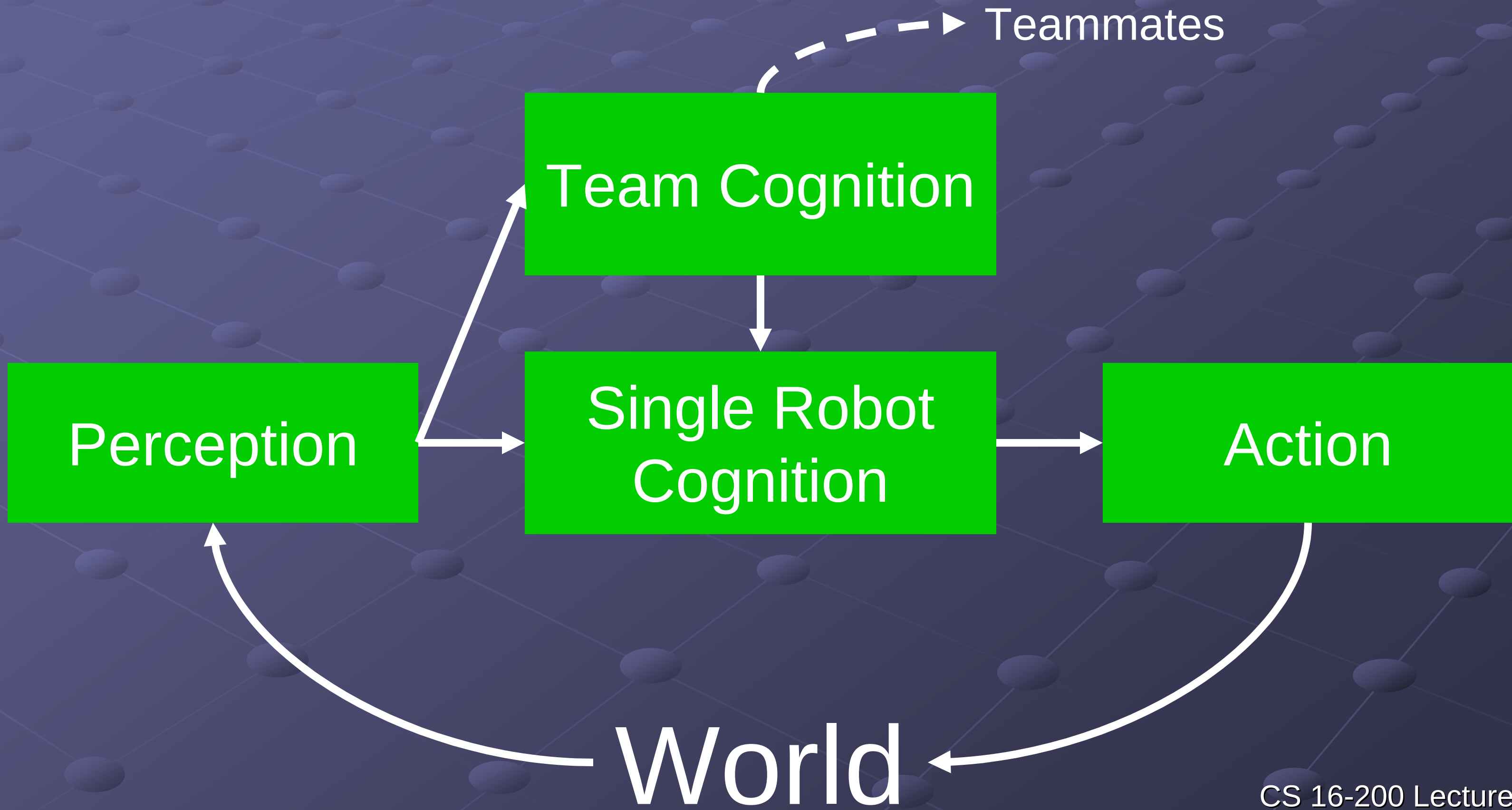
- Autonomous robots with real-time perception
- Operating in a dynamic environment
- Operating in a team with other robots, humans, novel teammates
- Operating with adversaries
  - Creates highly dynamic environment
  - Encourages high performance solutions
  - Encourages strategy adaptation and learning

My Interests

- Real-time vision perception
- Autonomously robot control
  - Individually, and within team
- Effective learning mechanisms
  - Adapt to environment, changes, opponents, task

Lets Focus on Control

- Robot must choose actions to perform its role within the team

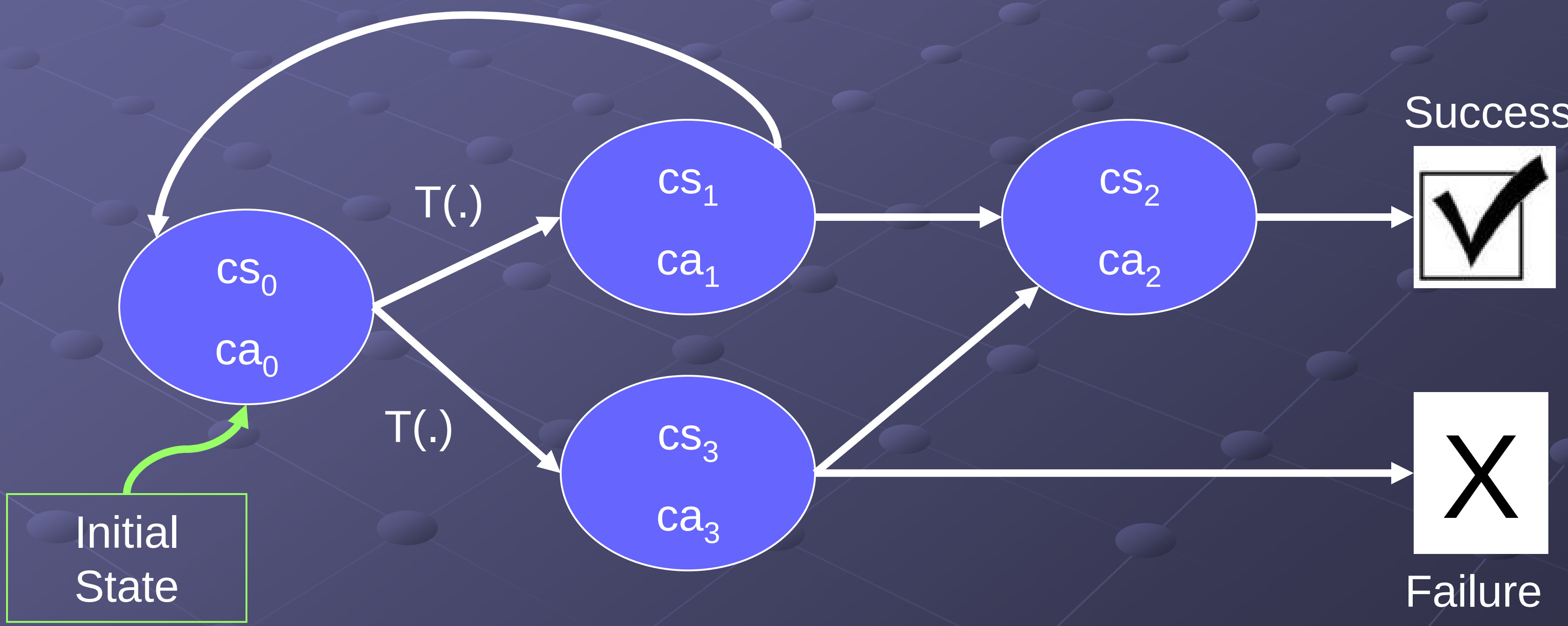


State Machines for Control

- State machines have a long history in robotics, control and AI
  - For behavioral control [Brooks 86, Balch *et al.* 95]
  - Hybrid control [Lynch & Krogh,00]
  - State estimation [Thrun *et al.* 05], many more
- State abstraction provides a powerful mechanism for describing (and implementing) sequences with different modes of control

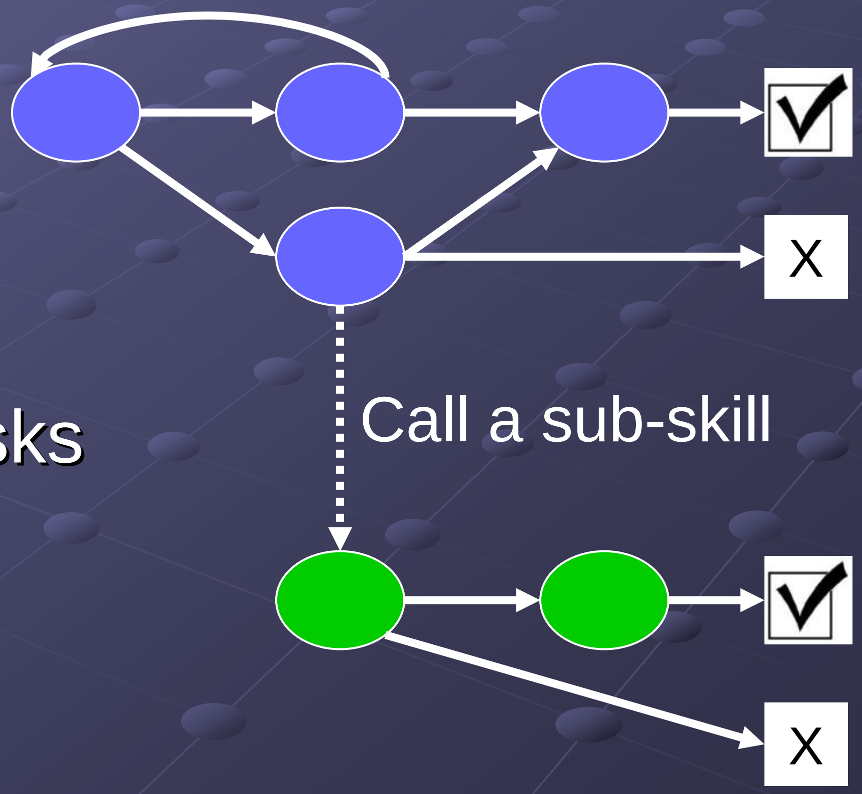
Definition of a State Machine

- PS – Perceptual state
- CS – Control states
- CA – Control actions
- T(PS+) – Transition function
- R – Termination result

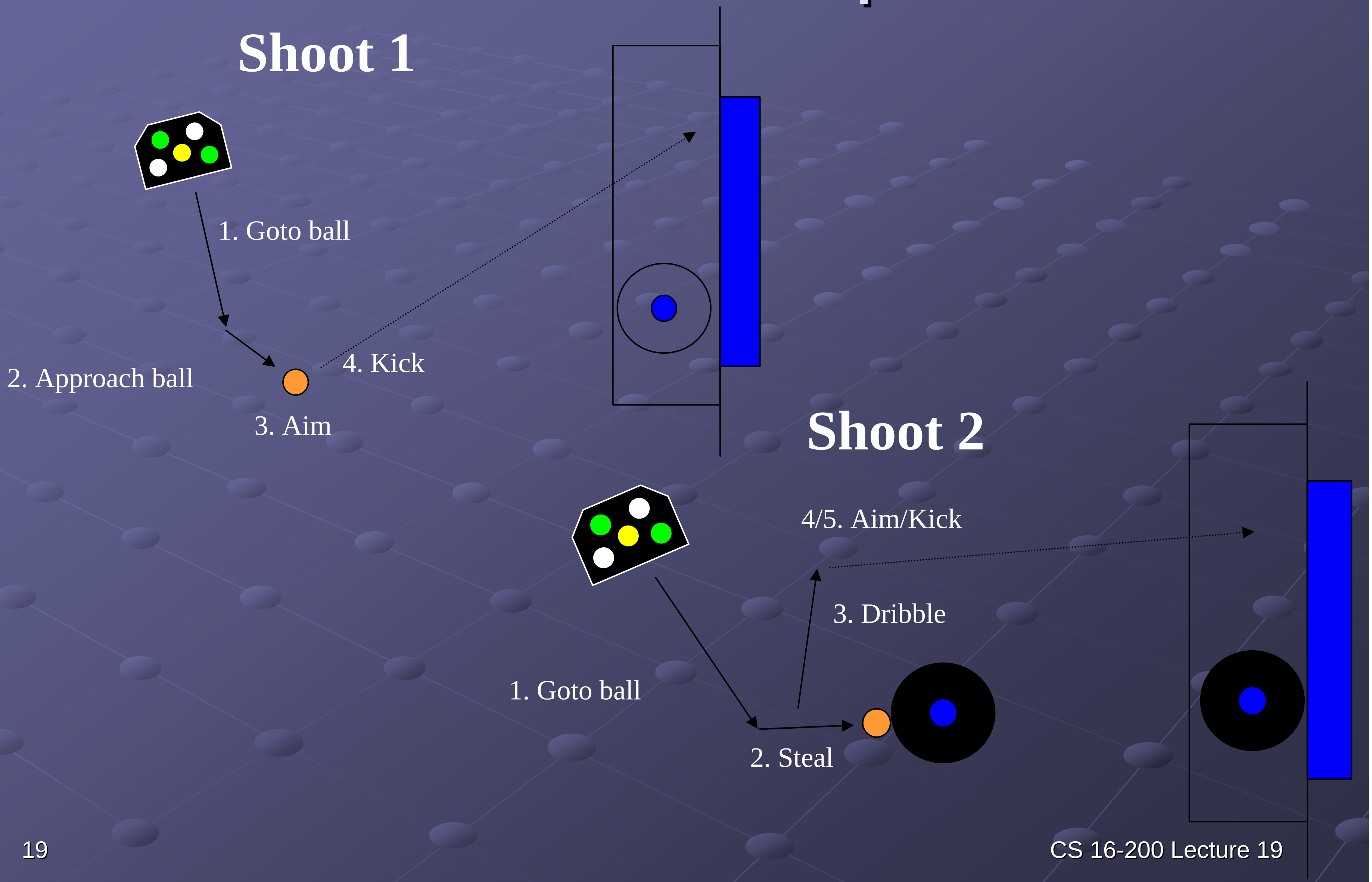


Hierarchies of State Machines for Individual Control

- Really equivalent to a larger state machine
- Allows for state machine reuse (i.e. Macros)
- Allows for natural task decomposition into sub-tasks

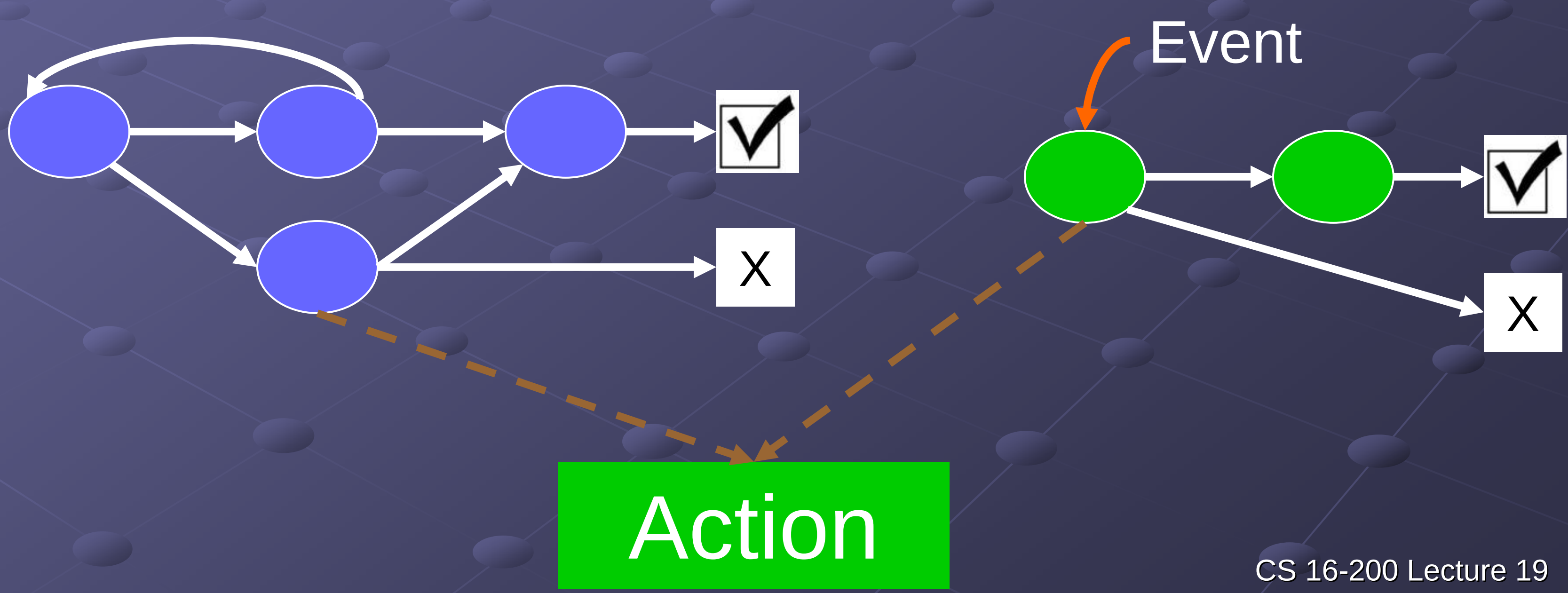


Soccer Example



Parallel Execution of Hierarchies

- Operate state machines in parallel
  - Independent or non-conflicting tasks
  - Requires scheduling for conflicting tasks
- Event driven execution, sleeping states



Skill Kernel

- Call a state machine a skill
- Idea: We can equate management of skills with a multi-threaded operating system

OS Terms	Skill Terms
Thread	Skill
Resource management	Action management
Sleeping	Sleeping
Signals	Events
Scheduling	?
Synchronization	?

Skill Learning

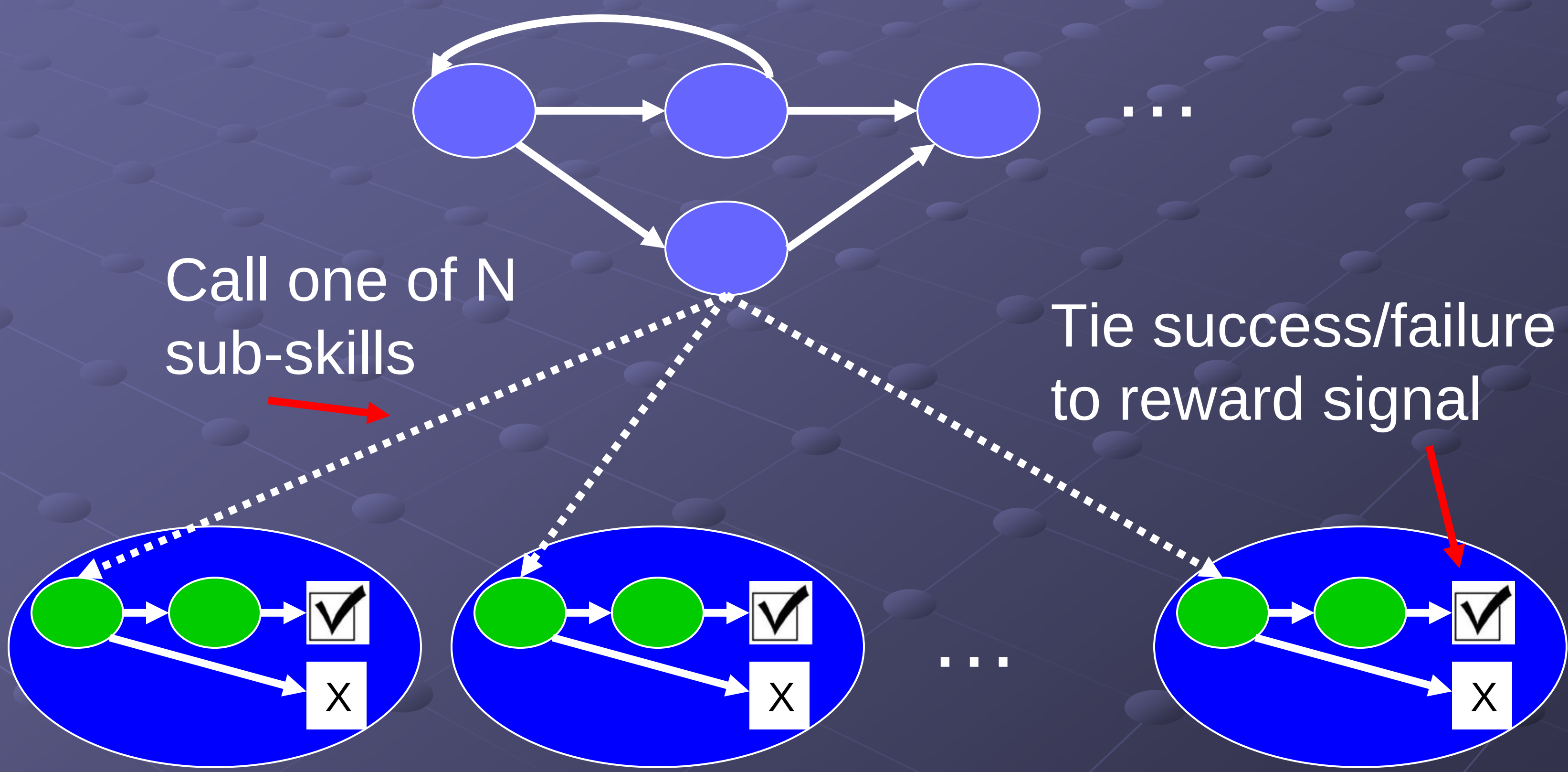
- Key idea:
  - Skill structure and kernel creates natural basis for applying learning
- Learning applicable in three ways
  - Learning control policy in a state
  - Learning state transitions
  - Learning hierarchy/which skills to call

Skill Selection Learning

- Key idea:
  - In a state a skill may call multiple sub-skills to do the same task
  - Learn which sub-skill works best
- Approach
  - Use success/failure history as reward signal
  - Apply an expert systems technique to learn which ‘expert’, or skill, is best

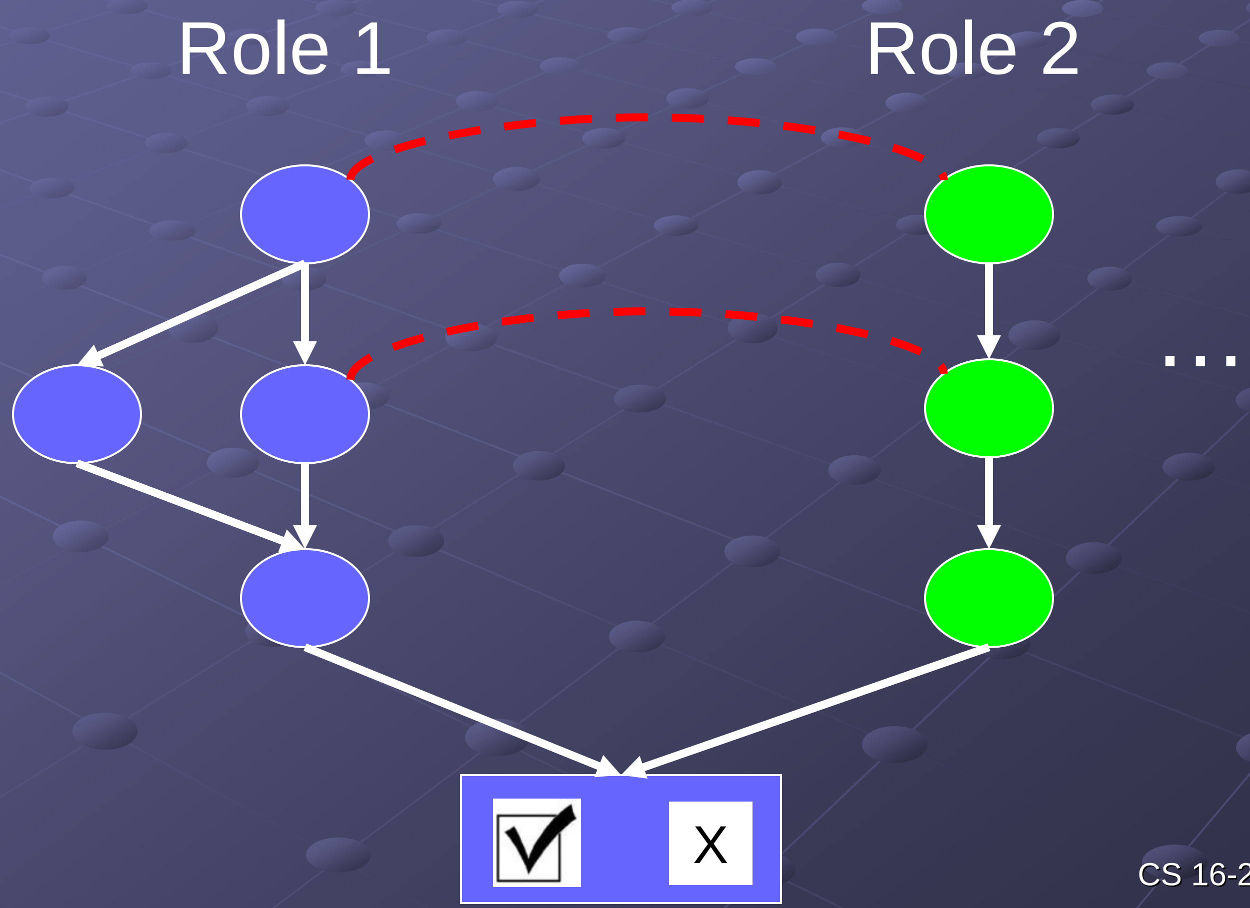
Skill Selection Approach

- Treat a skill as an 'expert'



Coordinating Robots: Plays

- Synchronized state machines



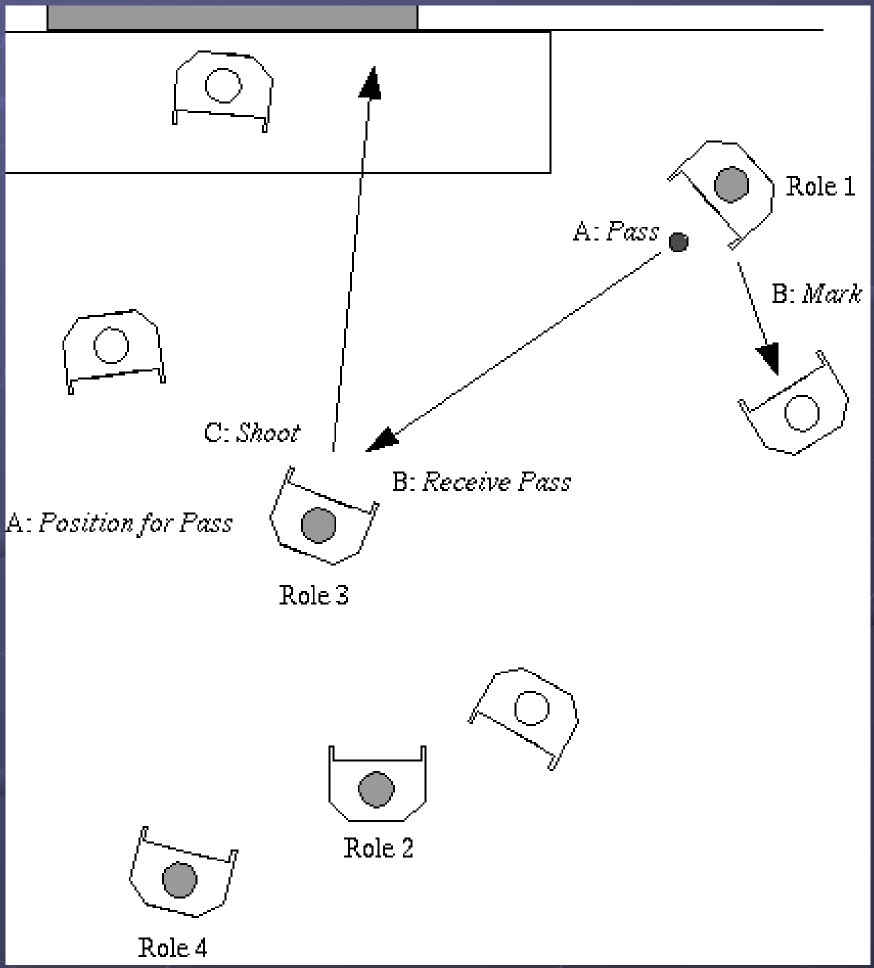
Play Manager

- Manages selection, execution and synchronization for each play
- Beyond synchronization
  - Play selection
  - Dynamic role assignment
  - Monitoring
  - Learning

An Example Play

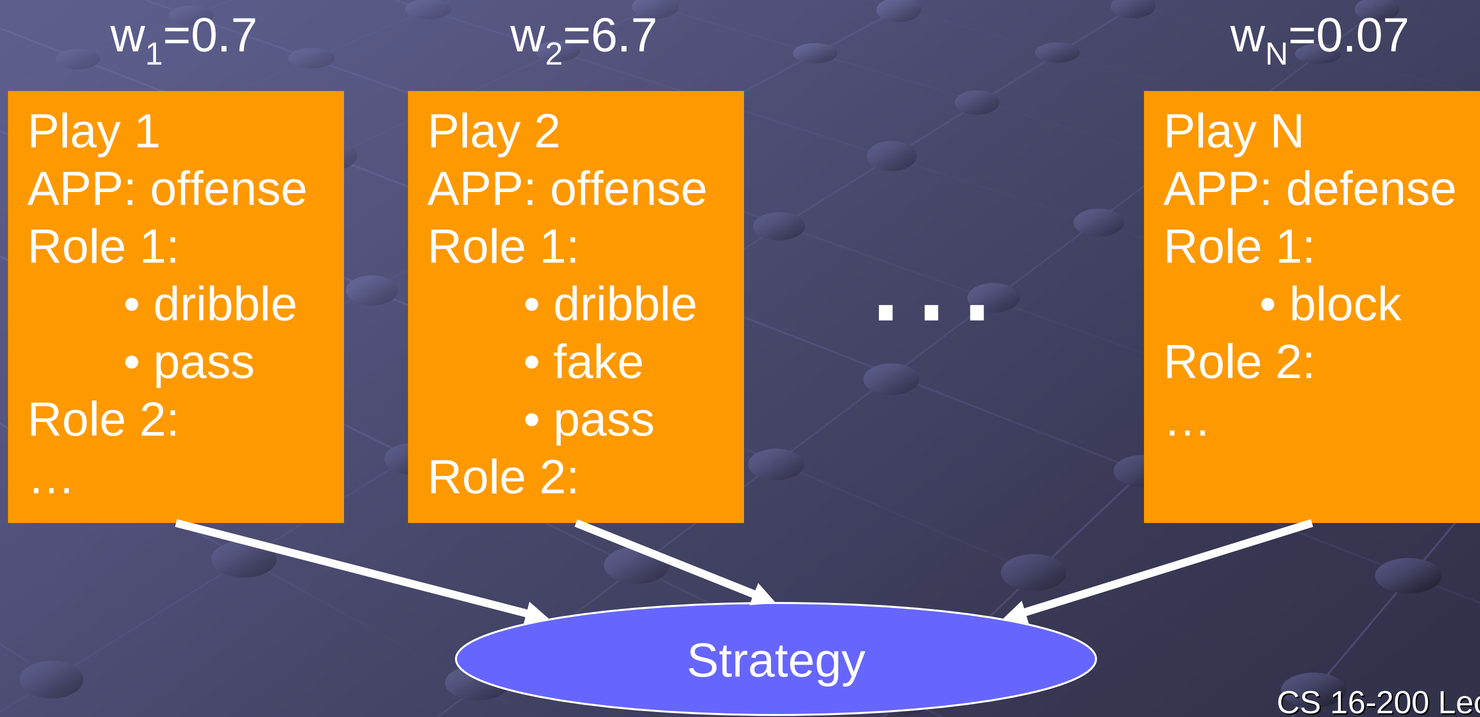
APPLICABLE offense
DONE aborted !offense

ROLE 1
pass 3
mark best_opponent
ROLE 2
block
ROLE 3
pos_for_pass R B 1000 0
receive_pass
shoot A
ROLE 4
defend_lane



Playbook Strategy

- Play Manager handles multiple plays, and select as appropriate for world state
- Learn which plays work better



Summary

- State abstraction is a powerful technique that can be used at many levels from single robot to team coordination
  - Management of state machines equivalent to multi-threaded OS management
  - Provides a natural basis to apply learning
  - Provides a natural mechanism for task decomposition

End of Lecture

See you on Wednesday