Probabilistic Verification of Coordinated Multi-Robot Missions

Sagar Chaki, Joseph Giampapa
Software Engineering Institute, CMU
July 8, 2013

Software Engineering Institute | Carnegie Mellon

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0000502

Motivation

Robots increasingly used to perform a wide variety of tasks

- Involving dangerous of inhospitable situations
- Example: Robotic demining
  - http://www.ri.cmu.edu/research_project_detail.html?project_id=220&menu_id=261

Have to face uncertain situations

- Internal (e.g., sensor accuracy)
- External (e.g., presence of mine

Typically operate in teams

Mission designers control number of teams, size of each team, capability of each robot, etc.

- Currently an ad-hoc process

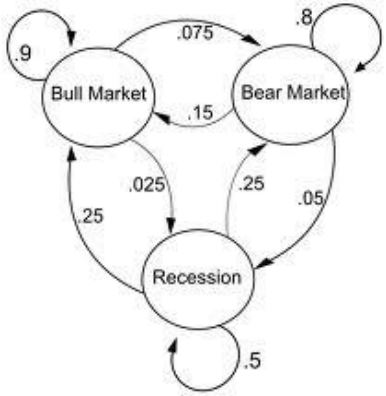
Problem: Analytically find the mission design that optimizes (i.e., maximizes probability or expected value of) some overall system utility

Solution: Use Probabilistic Model Checking

Specifically, model checking (a restricted type of) discrete time Markov chains (DTMCs)

Widely studied over many years, well-understood theory

- For example: C. Baier. On algorithmic verification methods for probabilistic systems. Habilitation Thesis. 1998



Mature tools

- PRISM: http://www.prismmodelchecker.org/

Contributions of this paper

- Identifying a restricted class of probabilistic automata that naturally model coordinated multi-robot missions
- Showing that probabilities and expected rewards for these can be computed compositionally
- Empirical validation on a robotic demining example

Our Focus: Forage and Rescue (FAR) Missions

Robots explore an arena, look for objects, and react in specific ways

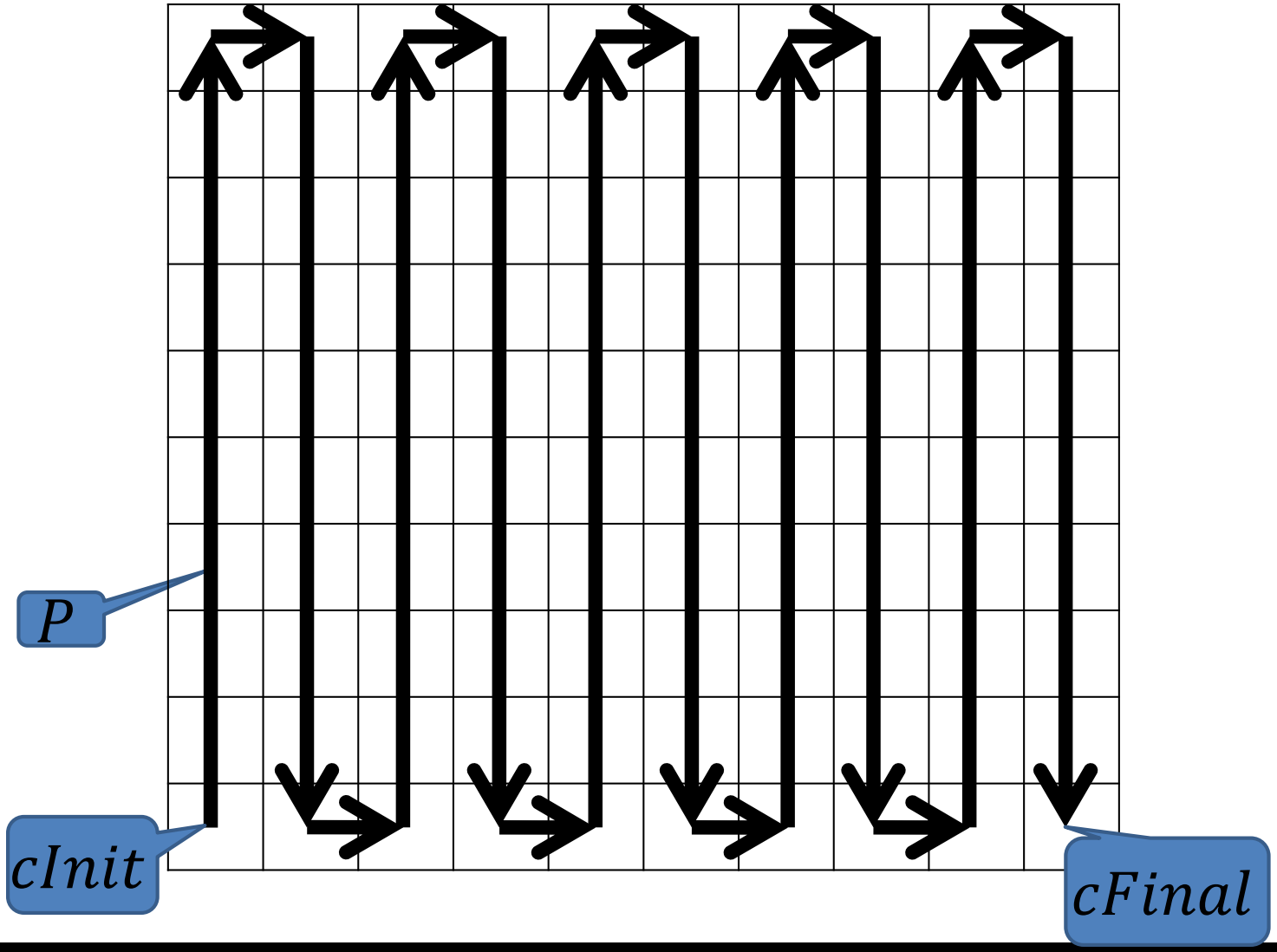
Example: robotic demining

- Two-dimensional array of cells
- Randomly seeded with mines
- T teams each consisting of N robots
- Cells pre-allocated to teams

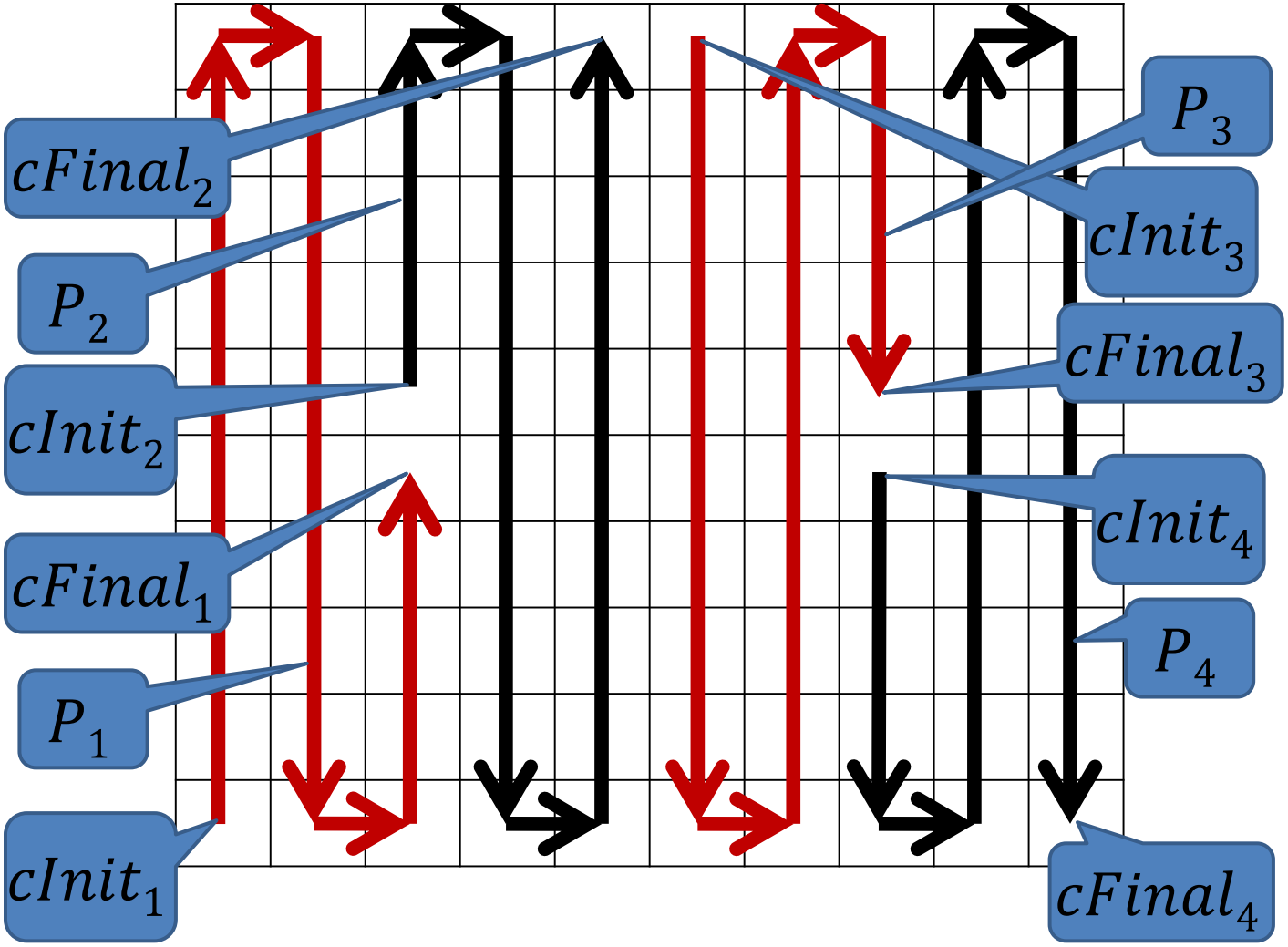
Each team has one leader and zero or more followers. Teams operate independently as follows:

- Each team follows a pre-defined path to explore all cells assigned to it
- In each cell, the leader tries of detect a mine. If a mine is found, the leader tries to defuse it. If it could not be defused, the cell is marked as being mined. The team moves to the next cell in its path.
- If the leader explodes, a new leader is elected via a standard protocol

Example: Robotic Demining with 1 team



Example: Robotic Demining with 4 teams



Sources of Uncertainty

External: Due to the terrain

- The leader sometimes fails to detect a mine
- The time to defuse varies from mine to another
- Team cannot move to next cell due to locomotion issues

Internal: Due to robot capability

- Mine explodes while being defused
- Mine explodes while a cell is being marked

External: Due to communication

- Leader election algorithm fails

Properties

Property 1: Probability of $Success$, where:

$Success$ = Every team covers all the cells allocated to it within a given deadline D without missing a single mine

Property 2: Expected value of $Coverage$, where:

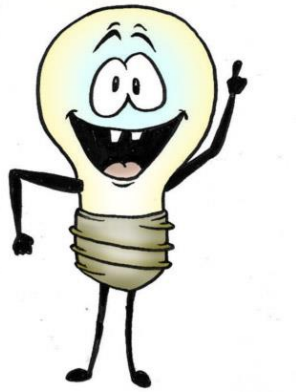
$Coverage$ = Total number of cells covered by all the teams within a given deadline D

Goal: Compute Properties 1 and 2 for various combinations of values of D, T, N and probabilities expressing the uncertainties we consider

First Cut

Model each team as a DTMC. Let model for team i be M_i

The overall mission model is the parallel composition of each team DTMC, i.e., $M = M_1 \parallel \cdots \parallel M_T$



Express properties 1 and 2 using PCTL

Model check using PRISM

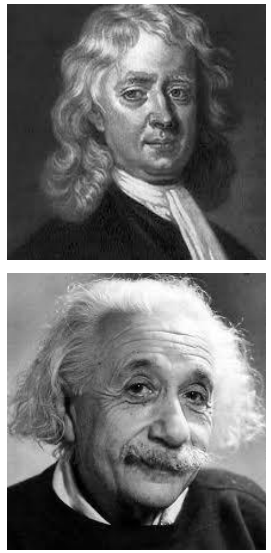
Does not work. Two problems:

1. **Theoretical:** DTMC not closed under parallel composition. Properties 1 and 2 don't make in the composed model.
2. **Practical:** Statespace blows up

Second Cut

Properties 1 and 2 don't make sense if we compose the team models (even partially) asynchronously

But the properties do make sense in the "real world"



Ergo, something is synchronizing the teams … time

- The teams don't have synchronized clocks
- But they march to the tick of the same global clock
- Assuming they are not zooming at relativistic speeds

So, what happens if we just stick to DTMCs with a singleton alphabet

- αPA : each transition labeled by α, which corresponds to an unit of time
- Closed under parallel composition (theoretical problem solved)
- Compositionality results for probabilities and expected rewards (practical problem solved)

Definitions (1)

αPA. $Probabilistic\ Automaton\ M = (S, Init, \Sigma, \delta, AP, \mathcal{L})\ where$

$(i)\ S\ is\ a\ set\ of\ states$

(ii) Init = initial state

(iii) $\Sigma = \{\alpha\} = alphabet$

$(iv)\ \delta\ is\ the\ transition\ relation\ that\ maps\ each\ state\ to\ a\ probability\ distribution\ over\ all\ the\ states$

$(v)\ AP = atomic\ propositions$

$(vi)\ \mathcal{L}\ labels\ states\ with\ sets\ of\ atomic\ propositions.$

Execution. *Sequence of states following the transition relation.*

Cylinder. $If\ \hat{s}\ is\ a\ finite\ execution, then\ Cyl(\hat{s})\ is\ the\ set\ of\ all\ its\ infinite\ extensions.$

Definitions (2)

LTL. *Linear temporal logic. A model is an infinite execution of an αPA defined in the standard way.*

Result. *The set of all executions of a probabilistic automaton M satisfying an LTL formula Ψ is expressible as a countable union of cylinders. The probability P(M, Ψ) of M satisfying Ψ is measurable.*

Parallel Composition. *Synchronous. Result $M_1 \parallel M_2$ is also an αPA.*

Reward Structure. *$R = (\rho, \iota)$ where ρ maps states to rewards and ι maps transitions to rewards. For any αPA state s and reward structure R, the cumulative reward up to k steps $C_{\leq k}(s, R)$ defined in a natural way by taking the weighted (probabilities) of all rewards.*

Result 1: Compositionality of Probabilities

$Theorem\ 2.\ Let\ M_1, \ldots, M_n\ be\ \alpha PA\ with\ disjoint\ atomic\ propositions.$
$Let\ \Psi_1, \ldots, \Psi_n\ be\ LTL\ formulas\ such\ that\ \Psi_i\ is\ over\ AP_i. Then:$

$$P(M_1 \parallel \cdots \parallel M_n, \Psi_1 \wedge \cdots \wedge \Psi_n) = \prod_{i=1}^{n} P(M_i, \Psi_i)$$

Probabilities of satisfying LTL formulas are compositionally computable

- Essentially due to the "independence" of the $M_i's$ and $\Psi_i's$

Moreover, $Success$ for T teams is expressible as $\Psi_1 \wedge \cdots \wedge \Psi_T$ which satisfy the conditions of $Theorem\ 2$

Result 2: Compositionality of Rewards

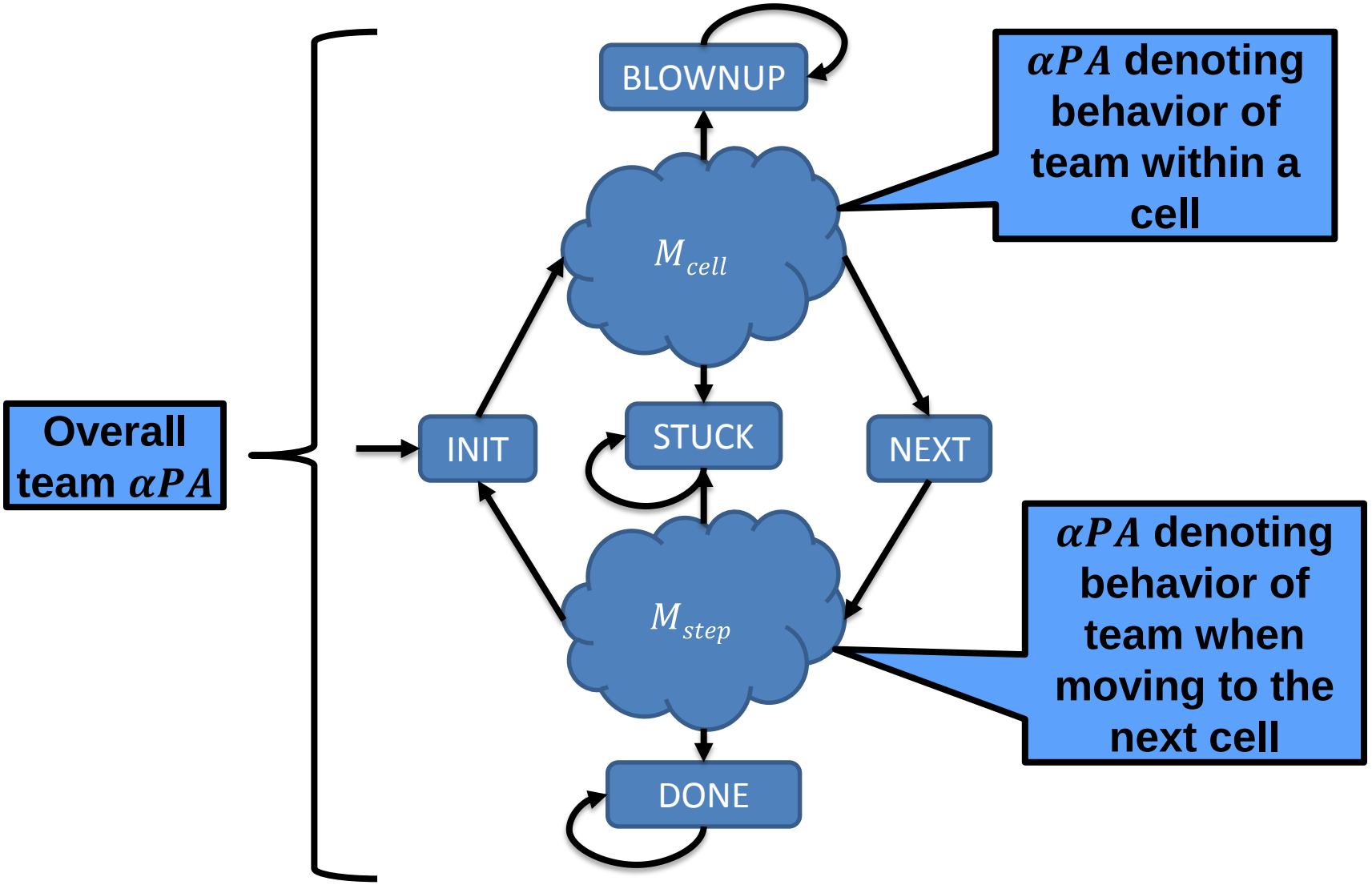
$Theorem\ 4.\ Let\ M_1, \ldots, M_n\ be\ \alpha PA\ with\ disjoint\ atomic\ propositions.$
$Let\ \mathrm{R}_1, \ldots, \mathrm{R}_n\ be\ reward\ structures\ over\ M_1, \ldots, M_n, respectively. Then:$

$$\forall k.\, C_k(M_1 \parallel \cdots \parallel M_n, \mathrm{R}_1 \oplus \cdots \oplus \mathrm{R}_n) = \sum_{i=i}^{n} C_k(M_i, R_i)$$

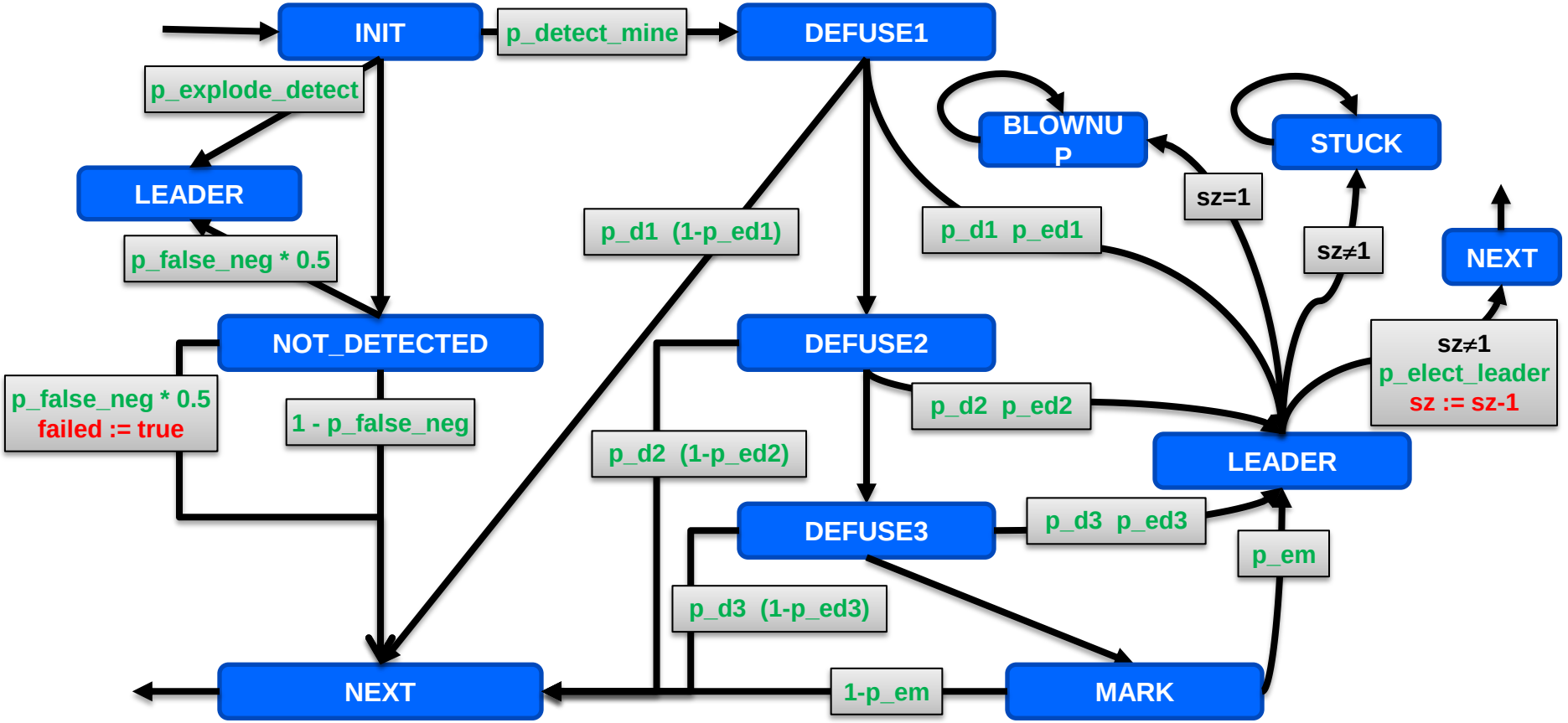
Rewards are compositionally computable

Moreover, $Coverage$ for T teams is expressible as $\mathrm{R}_1 \oplus \cdots \oplus \mathrm{R}_T$ which satisfy the conditions of $Theorem\ 4$

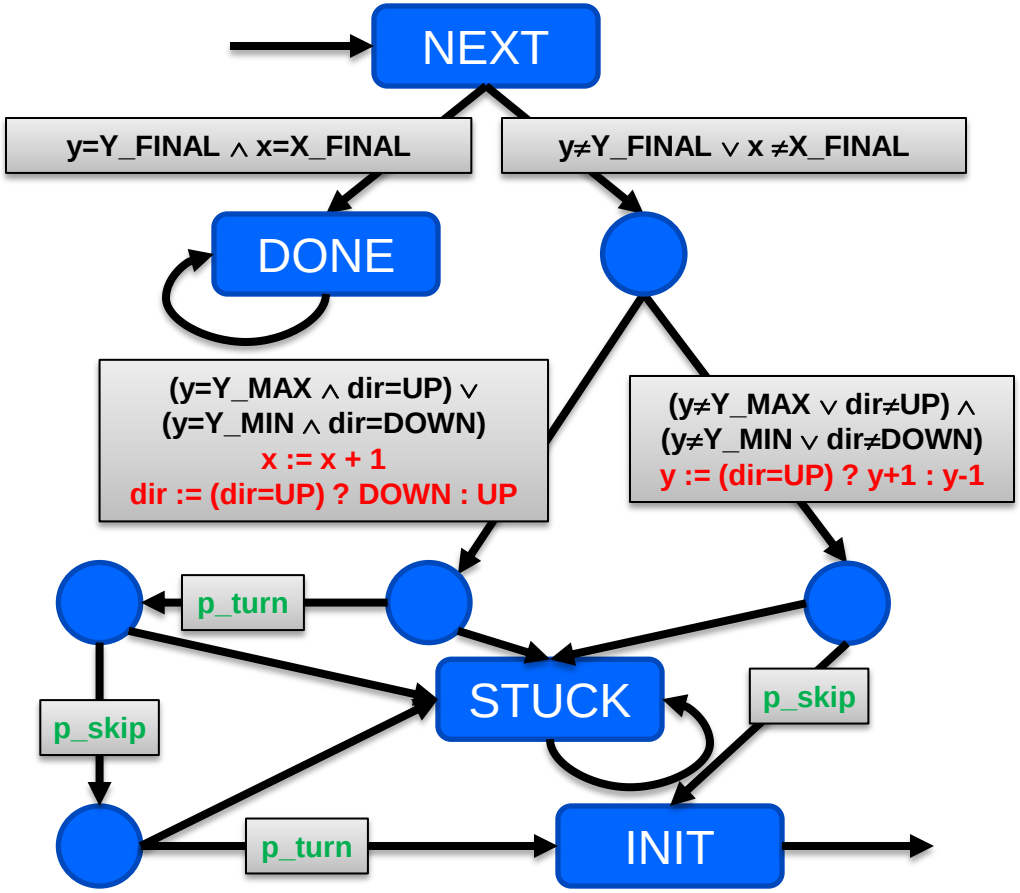
Modeling a team as a αPA

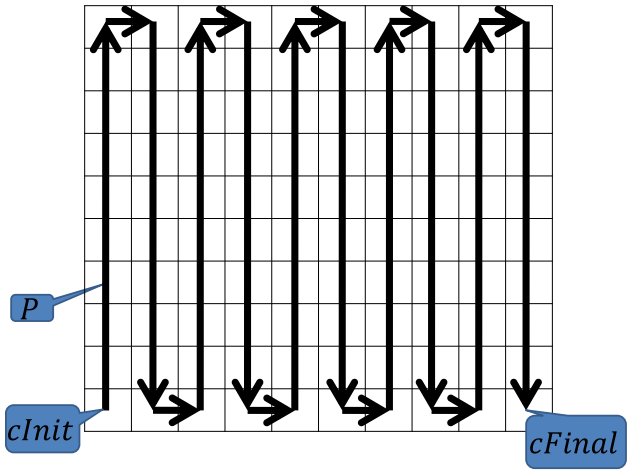


Behavior within a cell : M_{cell}



Moving to next cell : M_{step}





Experimental Results: $Success\ (\boldsymbol{D = 250})$

2 teams

Varying team size

DET=Low
DEF=High
LOC=Low

DET=Detection Capability
DEF=Defusing Capability
LOC=Locomotion Capability
Influence probability distribution

Average Time

T	N	$success_D$								Time
		LLL	LLH	LHL	LHH	HLL	HLH	HHL	HHH	seconds
2	2	0.000	0.000	0.000	0.000	0.013	0.014	0.035	0.035	21
2	3	0.001	0.001	0.003	0.004	0.065	0.066	0.129	0.131	26
2	5	0.018	0.018	0.030	0.031	0.256	0.259	0.355	0.359	38
2	10	0.073	0.074	0.086	0.087	0.386	0.391	0.443	0.449	62
2	15	0.076	0.077	0.087	0.089	0.386	0.391	0.443	0.449	87

Direct verification (using a model containing all the αPAs) with PRISM timed out at 1800 seconds. Compositionality results very useful in practice.

http://www.contrib.andrew.cmu.edu/~schaki/discover/spin13.tgz

Experimental Results: $Success\ (\boldsymbol{D = 250})$

2 teams

Varying team size

DET=Low
DEF=High
LOC=Low

DET=Detection Capability
DEF=Defusing Capability
LOC=Locomotion Capability
Influence probability distribution

Average Time

T	N	$success_D$			Time
		LLL		HLL	seconds
2	2	0.000		0.013	21
2	3	0.001		0.065	26
2	5	0.018		0.256	38
2	10	0.073		0.386	62
2	15	0.076		0.386	87

Prioritize improving DET over DEF and LOC. Got to detect the mine before you can do anything with it.

Experimental Results: $Success\ (\boldsymbol{D = 250})$

2 teams

Varying team size

DET=Low
DEF=High
LOC=Low

DET=Detection Capability
DEF=Defusing Capability
LOC=Locomotion Capability
Influence probability distribution

Average Time

T	N	$success_D$		Time
		LLL	LHL	
2	2	0.000	0.000	
2	3	0.001	0.003	
2	5	0.018	0.030	
2	10	0.073	0.086	
2	15	0.076	0.087	

Prioritize improving DET over DEF and LOC. Got to detect the mine before you can do anything with it.

Experimental Results: $Success\ (\boldsymbol{D = 250})$

2 teams

Varying team size

DET=Low
DEF=High
LOC=Low

DET=Detection Capability
DEF=Defusing Capability
LOC=Locomotion Capability
Influence probability distribution

Average Time

T	N	$success_D$		Time
		LLL	LLH	
2	2	0.000	0.000	
2	3	0.001	0.001	
2	5	0.018	0.018	
2	10	0.073	0.074	
2	15	0.076	0.077	

Prioritize improving DET over DEF and LOC. Got to detect the mine before you can do anything with it.

Experimental Results: $Success\ (\boldsymbol{D = 250})$

30 robots. Varying number of robots and team sizes.

T	N	$success_D$								Time
		LLL	LLH	LHL	LHH	HLL	HLH	HHL	HHH	seconds
2	15								0.449	87
3	10								0.498	46
6	5								0.494	29
10	3								0.441	35
15	2								0.264	48
30	1								0.003	100

$T = 3$ and $N = 10$ is optimal. $Success$ drops off sharply for $N > 5$. Smaller teams have higher likelihood to be completely destroyed or disabled.

Experimental Results: $Coverage\ (D = 250)$

2 teams

Varying team size

DET=Low
DEF=High
LOC=Low

DET=Detection Capability
DEF=Defusing Capability
LOC=Locomotion Capability
Influence probability distribution

Average Time

T	N	DET:DEF:LOC								Time
		LLL	LLH	LHL	LHH	HLL	HLH	HHL	HHH	seconds
2	2	43.3	43.3	48.3	48.4	61.8	62.0	70.6	70.8	7
2	3	60.1	60.2	66.2	66.4	80.8	81.0	89.1	89.3	7
2	5	82.1	82.4	87.5	87.7	97.8	98.1	102.5	102.8	7
2	10	93.5	93.8	96.2	96.5	101.7	102.0	104.6	105.0	7
2	15	93.6	93.9	96.2	96.5	101.7	102.0	104.6	105.0	7

Lesson is the same as for *Success*. Prioritize improving DET over DEF and LOC. Got to detect the mine before you can do anything with it.

Experimental Results: $Coverage\ (D = 250)$

30 robots. Varying number of robots and team sizes.

DET=Detection Capability

DEF=Defusing Capability

LOC=Locomotion Capability

Influence probability distribution

Average Time

T	N	DET:DEF:LOC								Time
		LLL	LLH	LHL	LHH	HLL	HLH	HHL	HHH	seconds
2	15								105.0	7
3	10								109.8	9
6	5								114.9	16
10	3								116.7	25
15	2								116.8	37
30	1								111.9	84

Lesson different from Success. $T = 10$ and $N = 3$ is optimal. More teams have higher likelihood of "covering" more cells, even if they end up getting destroyed.

Related Work

Modeling and verifying **probabilistic systems**

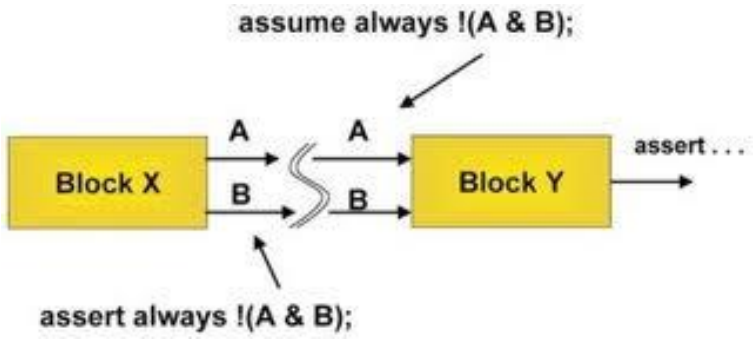
- Pacemakers
- Root-contention protocols
- Biological pathways

Probabilistic verification and **compositionality**

- Compositionality of probabilistic reactive modules

Assume-guarantee reasoning for verifying probabilistic systems

- Learning-based
- Asbtraction-refinement
- Hardware designs



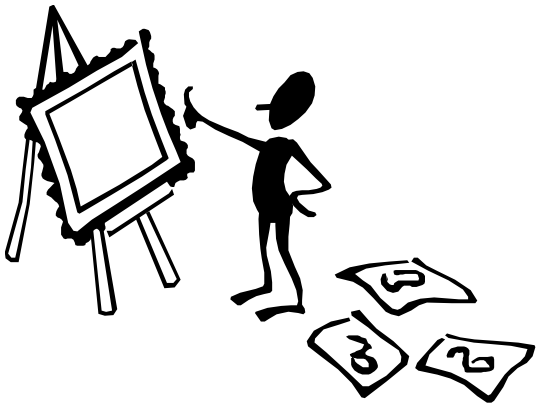
Workshop paper: **preliminary** work

- ARMS'13

Conclusion

This paper

- αPA : restricted but useful version of probabilistic automata
- Compositionality theorems
  - Probabilities of satisfying LTL claims
  - Cumulative rewards
- Empirical validation on a robotic de-mining example



Current Work

- Allowing intra-team coordination
- Field tests to see how predictions made by model checker hold up to reality
  - Estimate the atomistic probabilities, plug into the model
  - Refine model as needed

QUESTIONS?