Probabilistic CTL* - The Deductive Way

Lea Hiendl
Supervisor: Christoph Matheja

RWTH Aachen University
Software Modeling and Verification Group
Probabilistic Programming Seminar 2016

lea.hiendl@rwth-aachen.de

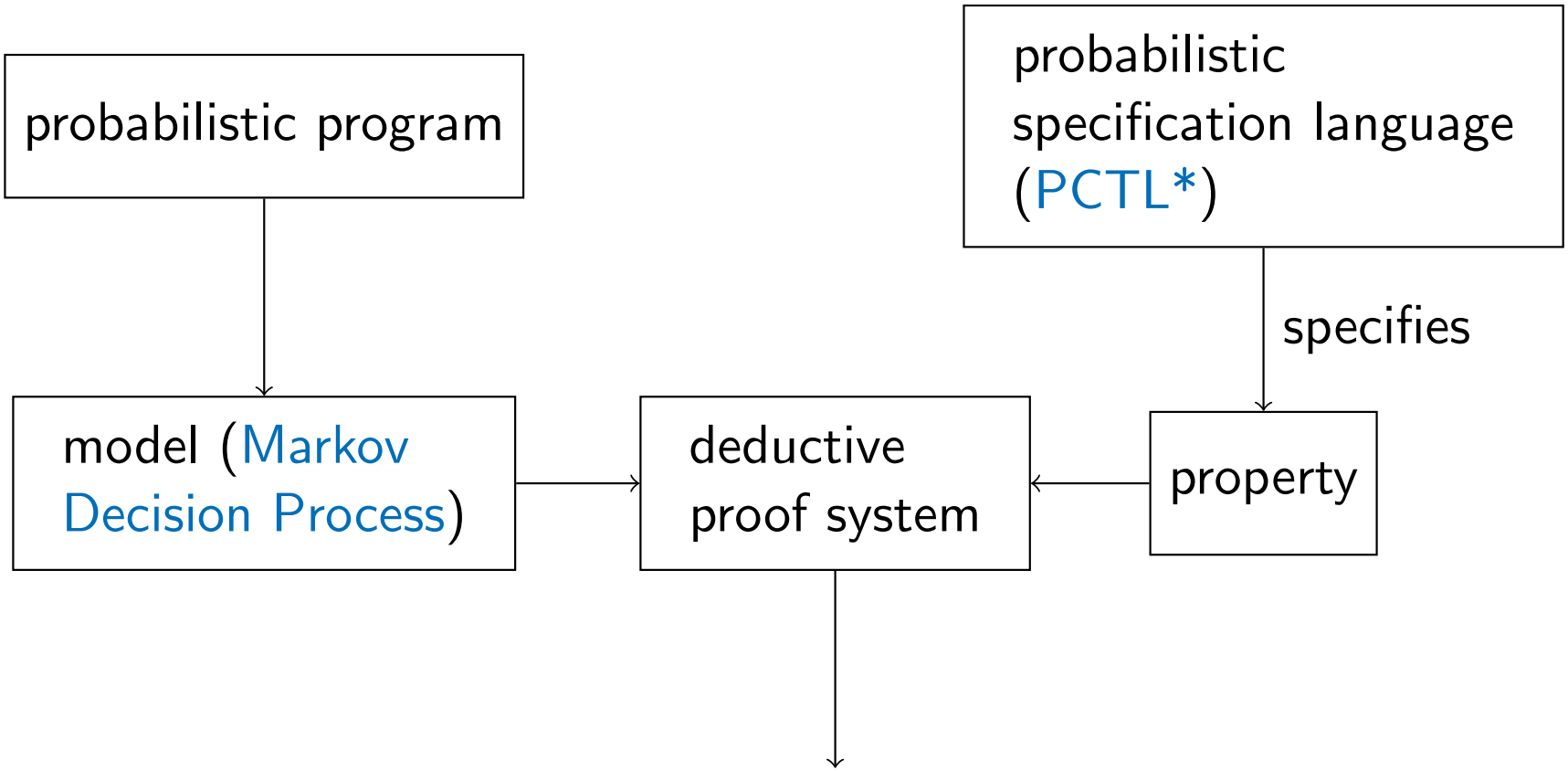
29th June, 2016

Motivation

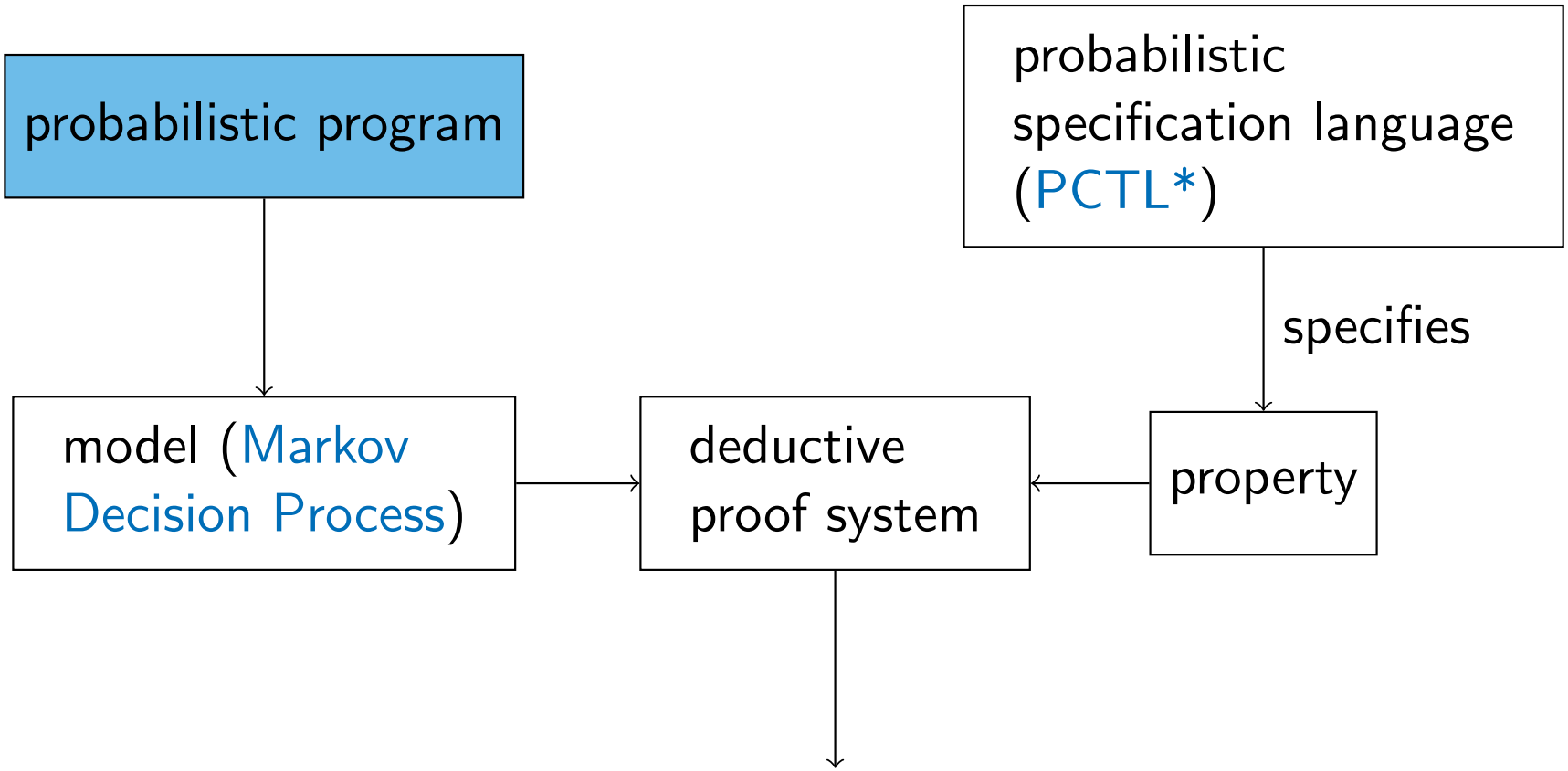
- PCTL*: specification language for model checking
- probabilistic model checking for finite state models is well-studied
- open research area: infinite state models
- "Probabilistic CTL* - The Deductive Way" by Dimitrova et al.:

$\rightarrow$ a sound (deductive) proof system that also works in infinite state spaces

Overview


probabilistic program
probabilistic specification language (PCTL*)
specifies
model (Markov Decision Process)
deductive proof system
property
Does the program satisfy the property?

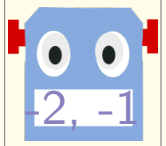
Overview


probabilistic program
probabilistic specification language (PCTL*)
specifies
model (Markov Decision Process)
deductive proof system
property
Does the program satisfy the property?

Robot Example

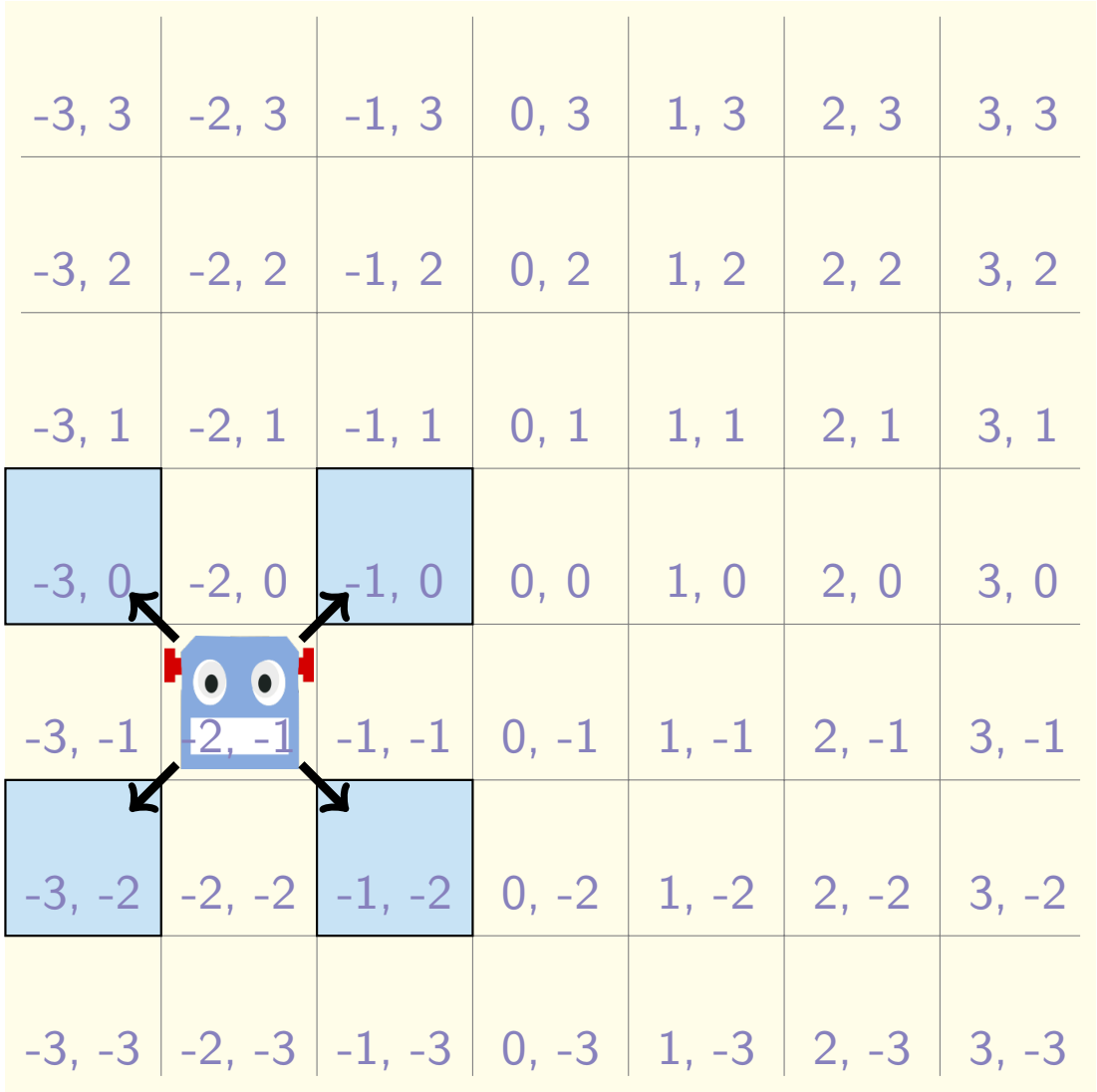
-3, 3	-2, 3	-1, 3	0, 3	1, 3	2, 3	3, 3
-3, 2	-2, 2	-1, 2	0, 2	1, 2	2, 2	3, 2
-3, 1	-2, 1	-1, 1	0, 1	1, 1	2, 1	3, 1
-3, 0	-2, 0	-1, 0	0, 0	1, 0	2, 0	3, 0
-3, -1	-2, -1	-1, -1	0, -1	1, -1	2, -1	3, -1
-3, -2	-2, -2	-1, -2	0, -2	1, -2	2, -2	3, -2
-3, -3	-2, -3	-1, -3	0, -3	1, -3	2, -3	3, -3

Robot Example

-3, 3	-2, 3	-1, 3	0, 3	1, 3	2, 3	3, 3
-3, 2	-2, 2	-1, 2	0, 2	1, 2	2, 2	3, 2
-3, 1	-2, 1	-1, 1	0, 1	1, 1	2, 1	3, 1
-3, 0	-2, 0	-1, 0	0, 0	1, 0	2, 0	3, 0
-3, -1		-1, -1	0, -1	1, -1	2, -1	3, -1
-3, -2	-2, -2	-1, -2	0, -2	1, -2	2, -2	3, -2
-3, -3	-2, -3	-1, -3	0, -3	1, -3	2, -3	3, -3

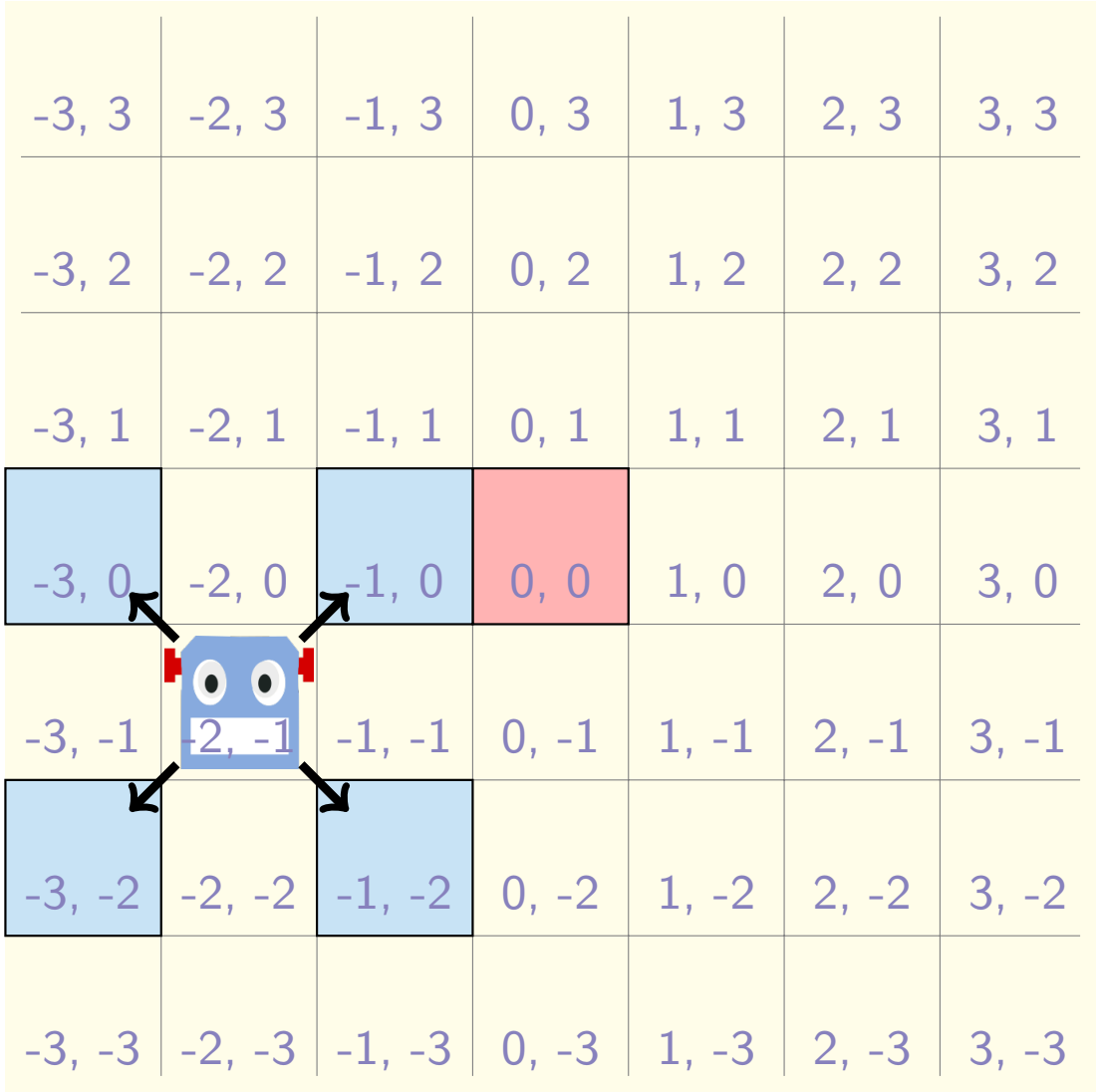
- robot at arbitrary starting position

Robot Example



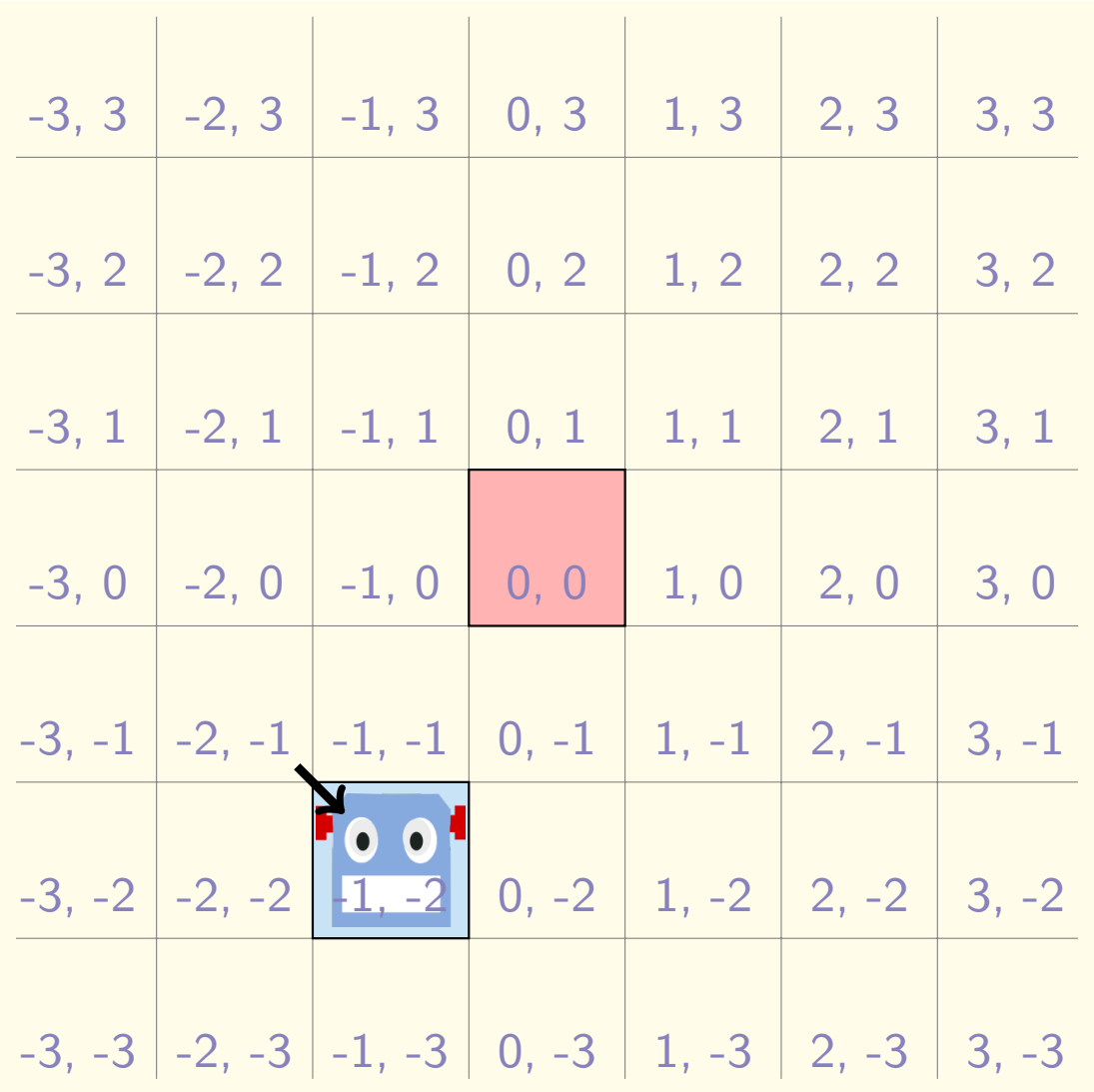
- robot at arbitrary starting position
- at every step: move one tile diagonally

Robot Example



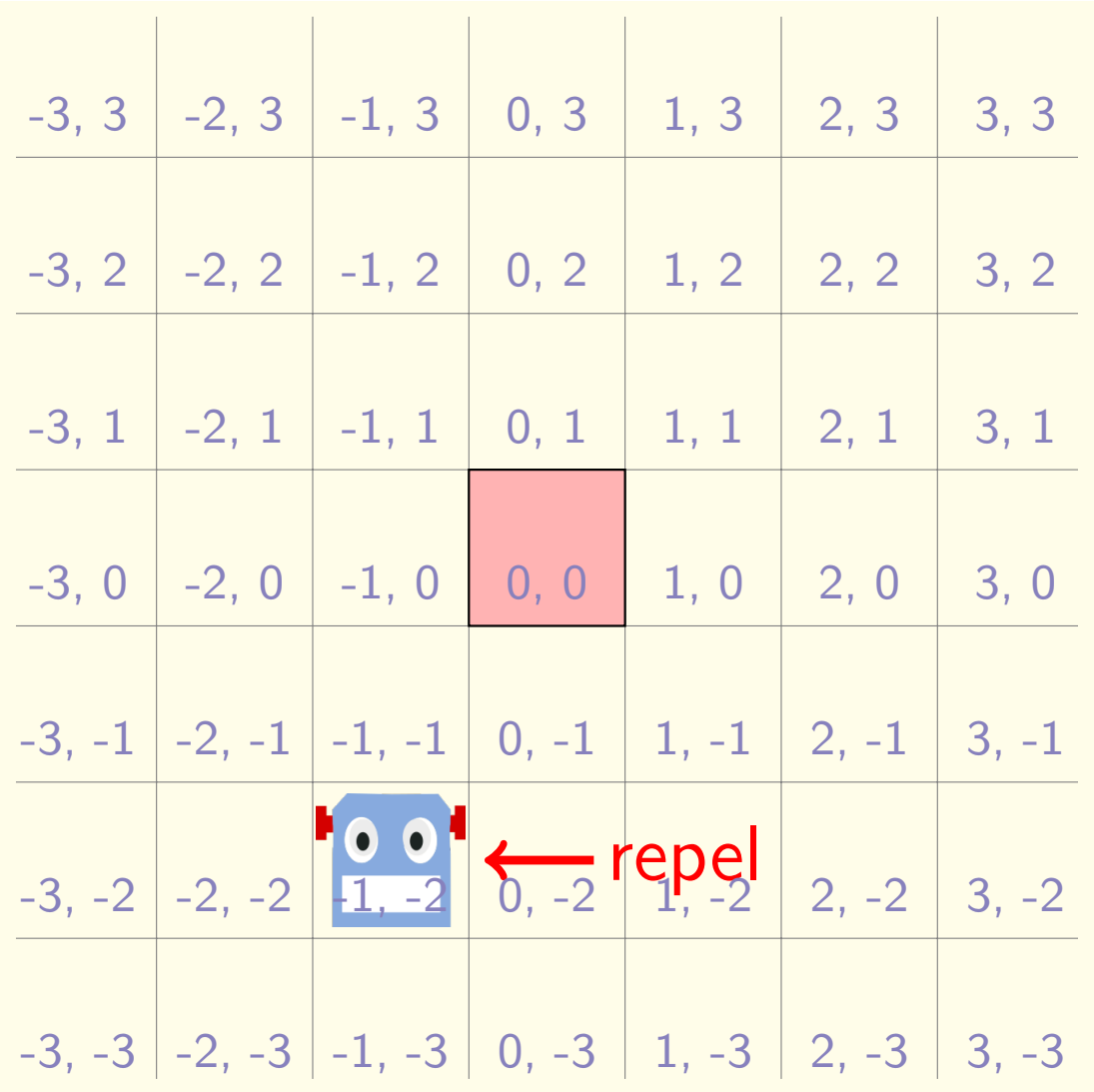
- robot at arbitrary starting position
- at every step: move one tile diagonally
- goal: visit the origin (0,0) infinitely often

Robot Example



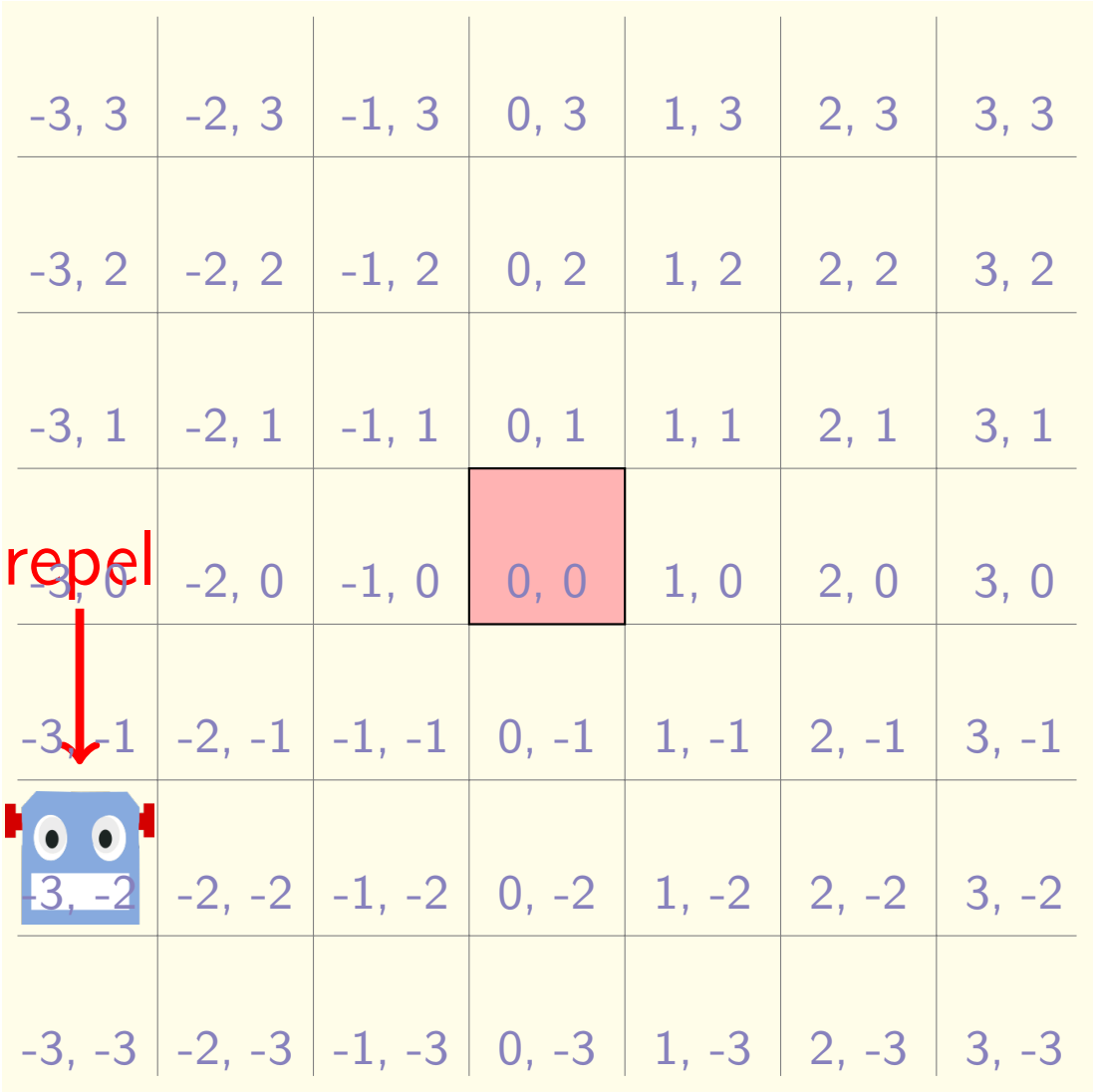
- robot at arbitrary starting position
- at every step: move one tile diagonally
- goal: visit the origin (0,0) infinitely often

Robot Example



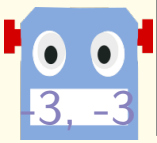
- robot at arbitrary starting position
- at every step: move one tile diagonally
- goal: visit the origin (0,0) infinitely often
- after every move, random forces repel the robot along x- and y-axis

Robot Example



- robot at arbitrary starting position
- at every step: move one tile diagonally
- goal: visit the origin (0,0) infinitely often
- after every move, random forces repel the robot along x- and y-axis

Robot Example

-3, 3	-2, 3	-1, 3	0, 3	1, 3	2, 3	3, 3
-3, 2	-2, 2	-1, 2	0, 2	1, 2	2, 2	3, 2
-3, 1	-2, 1	-1, 1	0, 1	1, 1	2, 1	3, 1
-3, 0	-2, 0	-1, 0	0, 0	1, 0	2, 0	3, 0
-3, -1	-2, -1	-1, -1	0, -1	1, -1	2, -1	3, -1
-3, -2	-2, -2	-1, -2	0, -2	1, -2	2, -2	3, -2
	-2, -3	-1, -3	0, -3	1, -3	2, -3	3, -3

- robot at arbitrary starting position
- at every step: move one tile diagonally
- goal: visit the origin (0,0) infinitely often
- after every move, random forces repel the robot along x- and y-axis

Probabilistic Programs

$P = (\mathbf{x}, C)$

- $\mathbf{x}$: finite (ordered) set of variables with countable domains
- C: finite set of guarded commands
  - deterministic:

$$g(\mathbf{x}) \mapsto \mathbf{x}' = \mathbf{e}(\mathbf{x})$$

$\rightarrow$ If guard g is fulfilled, assign $\mathbf{x}$ the values of $\mathbf{e}(\mathbf{x})$

  - probabilistic:

$$g(\mathbf{x}) \mapsto \mathbf{x}' = \mathbf{e}_1(\mathbf{x}) \otimes_{=p_1} \ldots \otimes_{=p_k} \mathbf{x}' = \mathbf{e}_{k+1}(\mathbf{x}) \qquad p_i \in [0,1]$$

$\rightarrow$ $\mathbf{x}' = \mathbf{e}_i(\mathbf{x}) \otimes_{=p_i}$: assign $\mathbf{x}$ the values of $\mathbf{e_i}(\mathbf{x})$ with probability p_i
or
assign $\mathbf{x}$ the values of $\mathbf{e}_{k+1}(\mathbf{x})$ with probability $1 - \sum_{i=1}^{k} p_i$

Robot Example: Probabilistic Program

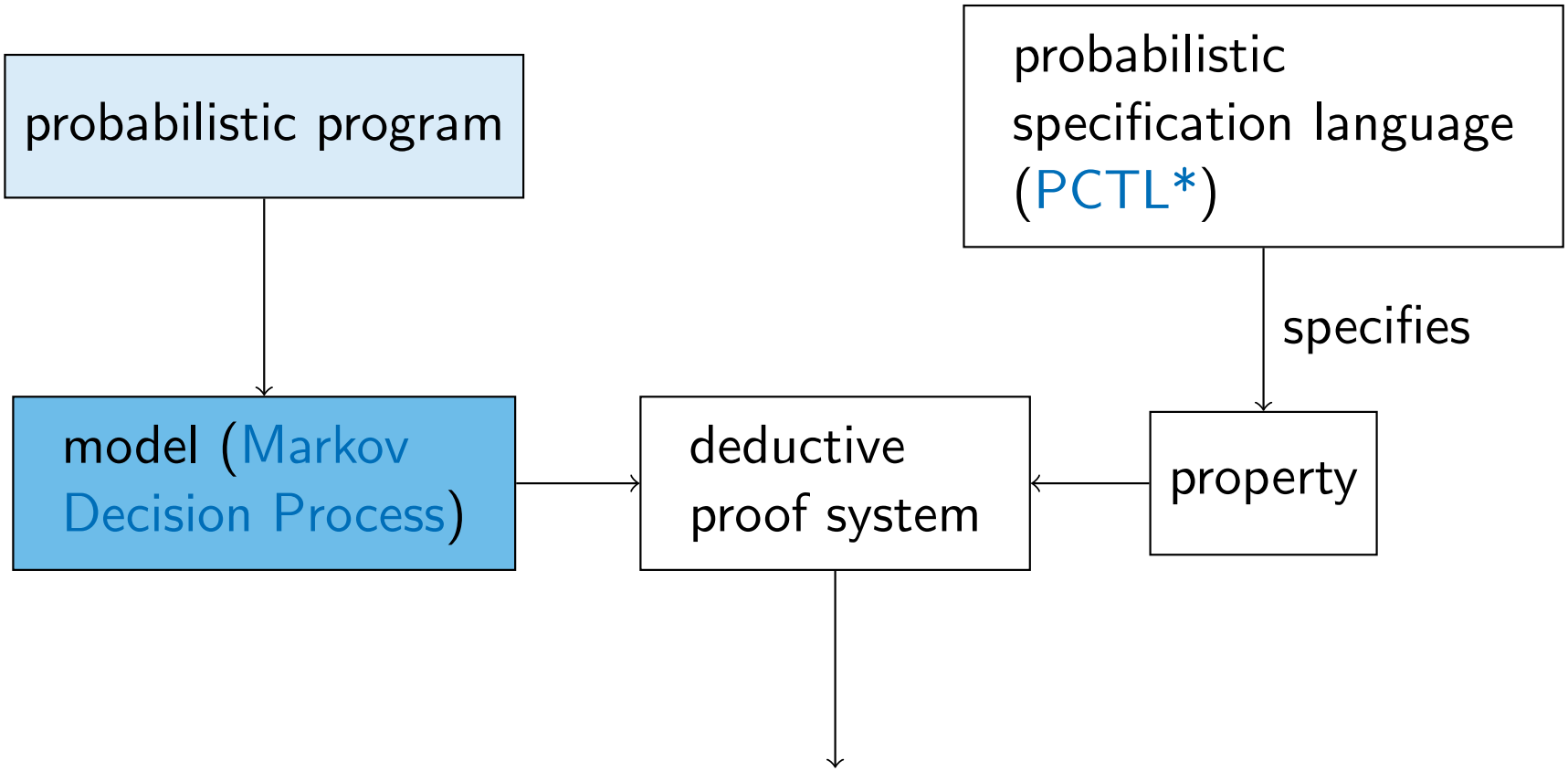
$P = (\mathbf{x}, C)$

- $\mathbf{x} : x \in \mathbb{Z}, y \in \mathbb{Z}, l \in \{0, 1, 2\}$
- C:

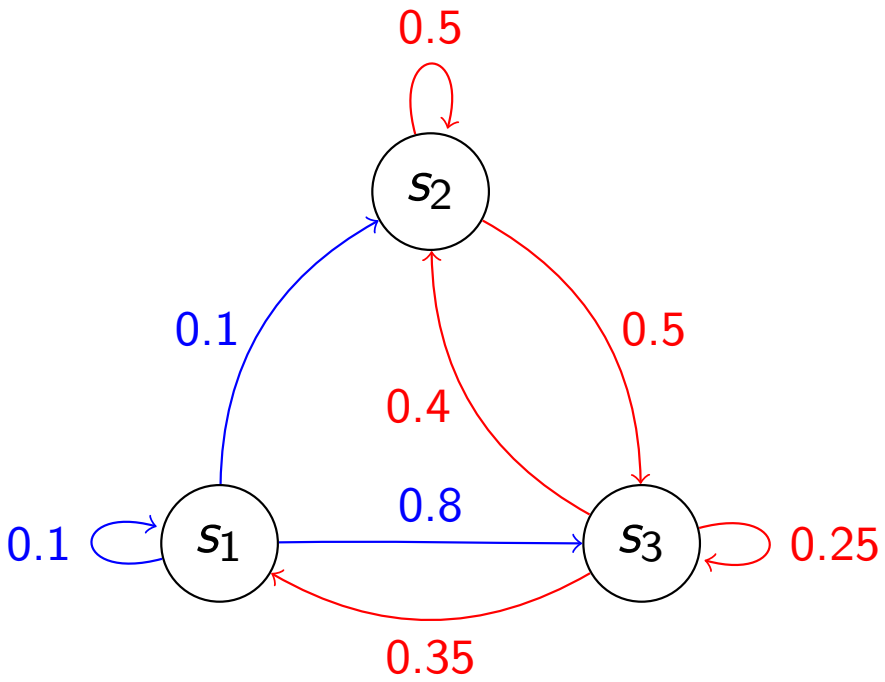
$$c_{NE} : l = 0 \mapsto x' = x + 1 \wedge y' = y + 1 \wedge l' = 1$$
$$c_{SE} : l = 0 \mapsto x' = x + 1 \wedge y' = y - 1 \wedge l' = 1$$
$$c_{NW} : l = 0 \mapsto x' = x - 1 \wedge y' = y + 1 \wedge l' = 1$$
$$c_{SW} : l = 0 \mapsto x' = x - 1 \wedge y' = y - 1 \wedge l' = 1$$

$$c_x : l = 1 \mapsto (x' = x + 9 \cdot \mathrm{sign}(x) \otimes_{=\frac{1}{|x|+1}} x' = x) \wedge y' = y \wedge l' = 2$$
$$c_y : l = 2 \mapsto (y' = y + 9 \cdot \mathrm{sign}(x) \otimes_{=\frac{1}{|y|+1}} y' = y) \wedge x' = x \wedge l' = 0$$

Overview


probabilistic program
probabilistic specification language (PCTL*)
specifies
model (Markov Decision Process)
deductive proof system
property
Does the program satisfy the property?

Markov Chain



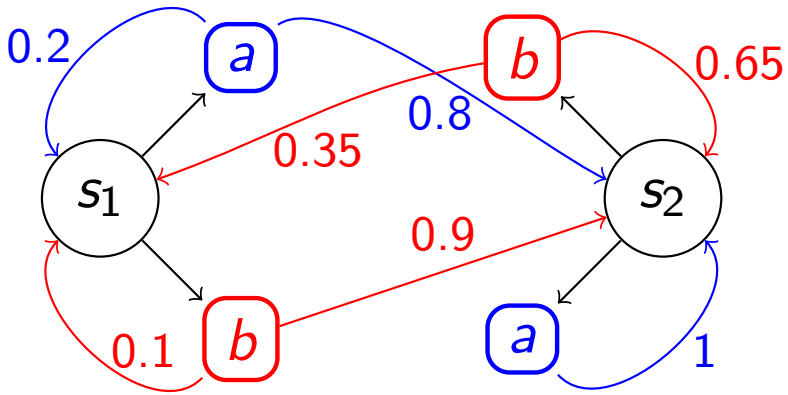
Discrete Time Markov Chain DTMC (S, ρ):

- S is a countable set of states
- $\rho : S \mapsto \mathit{Distr}(S)$ transition function

$\rho(s) =: \mu_s \in \mathit{Distr}(S)$, s.t. $\mu_s(s')$ is the probability that from state s, the transition to state s' is taken

Markov property: the probability of moving to the next state only depends on the current state and not the prior sequence of states

Markov Decision Process



MDP (S, A, ρ):

- S is a countable set of states
- A is a set of actions
- $\rho : S \times A \mapsto Distr(S)$ transition function

$\rho(s, a) =: \mu_{s,a} \in Distr(S)$, s.t. $\mu_{s,a}(s')$ is the probability that from state s, if the action a is chosen, the transition to state s' is taken

added **non-determinism**: choice between actions in every state

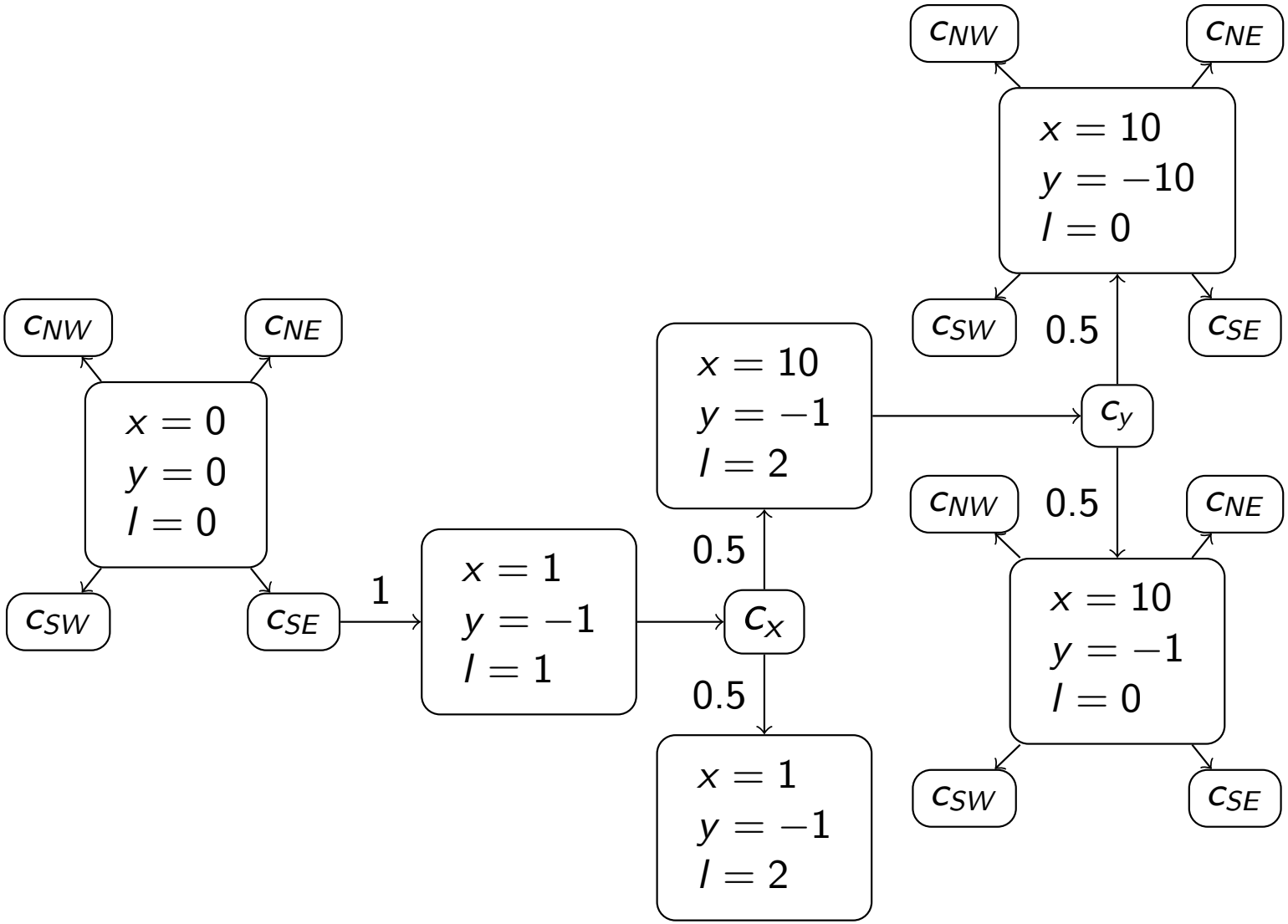
Scheduler

A scheduler is a function $\alpha : S^+ \mapsto \mathit{Distr}(A)$ that maps every non-empty sequence over S to a probability distribution over the action set A.

- deterministic: $\alpha : S^+ \mapsto A$
- memoryless: $\alpha : S \mapsto \mathit{Distr}(A)$

A scheduler for a MDP chooses an action in every state, thereby resolving the non-determinism. The result is a Markov Chain.

Robot Example: Markov Decision Process



infinite state space

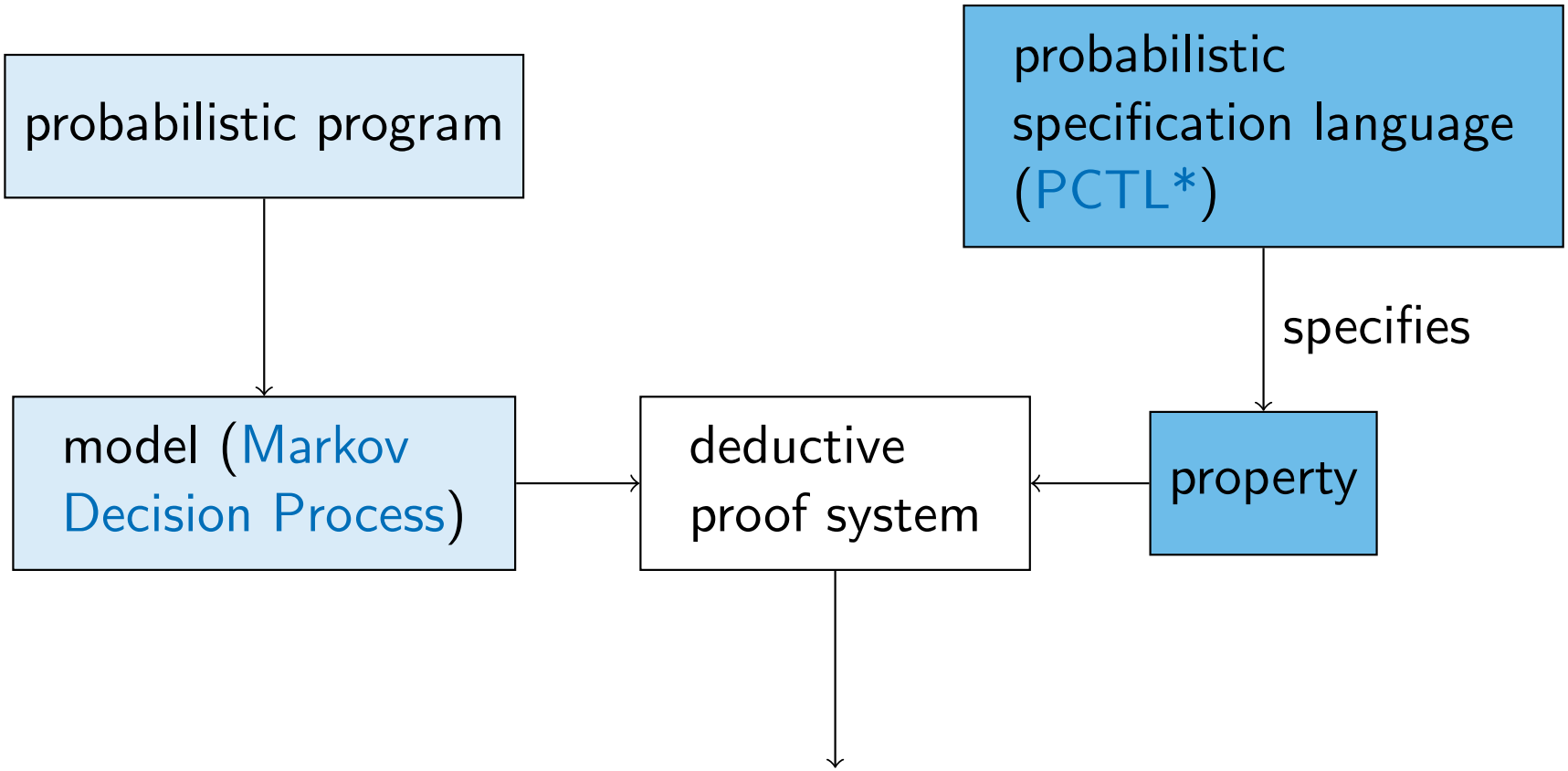
Robot Example: Scheduler

one possible (memoryless, deterministic) scheduler: $\alpha : S \to A$

$$\alpha(x, y, l) = \begin{cases} c_{NE} & l = 0, x < 0, y < 0 \\ c_{SE} & l = 0, x < 0, y \geq 0 \\ c_{NW} & l = 0, x \geq 0, y < 0 \\ c_{SW} & l = 0, x \geq 0, y \geq 0 \\ c_x & l = 1 \\ c_y & l = 2 \end{cases}$$

premise: "always attempt to reduce distance to origin"

Overview

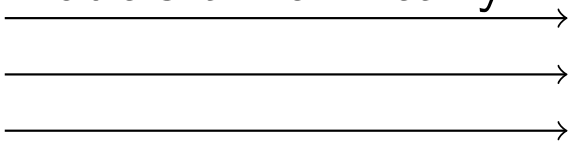

probabilistic program
probabilistic specification language (PCTL*)
specifies
model (Markov Decision Process)
deductive proof system
property
Does the program satisfy the property?

Temporal Logics

use temporal operators to make statements about time paths

Linear Time Logic (LTL)

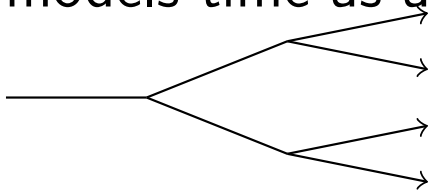
- models time linearly



- always makes statements about all paths

Computation Tree Logic (CTL)

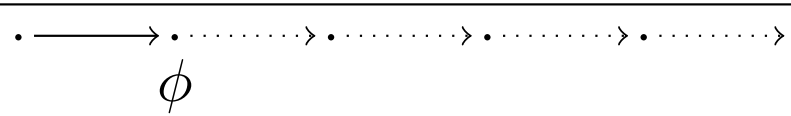
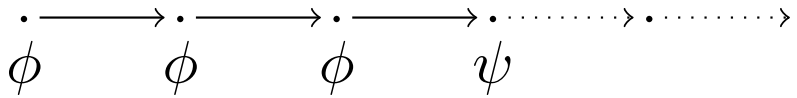
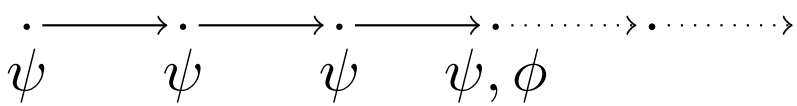
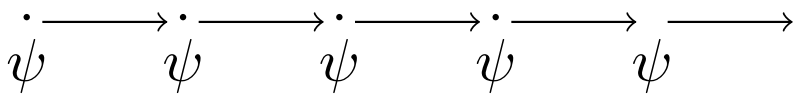
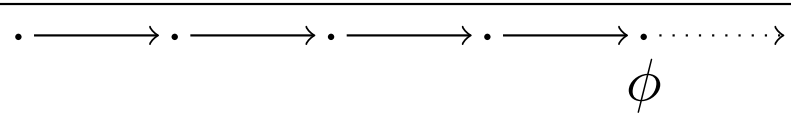
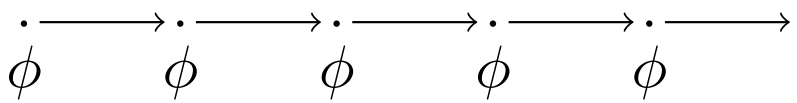
- models time as a tree



- uses quantifiers $\exists, \forall$ to make statements about all or some paths

CTL*: extension of CTL that allows statements about infinite paths

Temporal Operators

operator	derivation	semantics
$\bigcirc\phi$		
$\phi\mathcal{U}\psi$		
$\phi\mathcal{R}\psi$		 
$\Diamond\phi$	$=$ true $\mathcal{U}\phi$	
$\Box\phi$	$=$ false $\mathcal{R}\phi$	

Probabilistic CTL

- AP: set of assertions
- Grammar for state formulas:

$$\Phi := a \mid \neg a \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \mathbb{P}^{\forall}_{\bowtie p}(\varphi) \mid \mathbb{P}^{\exists}_{\bowtie p}(\varphi)$$

$\bowtie \in \{\leq, <, \geq, >\}, p \in \mathbb{R}_{\geq 0}, a \in AP$:

- Path formulas:

$$\varphi := \bigcirc \Phi \mid \Phi_1 \mathcal{U} \Phi_2 \mid \Phi_1 \mathcal{R} \Phi_2$$

Probabilistic CTL

- AP: set of assertions
- Grammar for state formulas:

$$\Phi := a \mid \neg a \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \underbrace{\mathbb{P}^{\forall}_{\bowtie p}(\varphi) \mid \mathbb{P}^{\exists}_{\bowtie p}(\varphi)}_{\text{probabilistic quantifiers}}$$

  $\bowtie \in \{\leq, <, \geq, >\}, p \in \mathbb{R}_{\geq 0}, a \in AP$:

- Path formulas:

$$\varphi := \bigcirc \Phi \mid \Phi_1 \mathcal{U} \Phi_2 \mid \Phi_1 \mathcal{R} \Phi_2$$

Probabilistic Quantifiers

$$P, s \models \mathbb{P}^{\forall}_{\bowtie p}(\varphi) \text{ iff } Prob_{s,a,}(\{\tau \in Paths(M_\alpha, \tau \models \varphi\}) \bowtie p$$
$$\text{for every scheduler } \alpha \text{ inducing a DTMC } M_\alpha$$
$$P, s \models \mathbb{P}^{\exists}_{\bowtie p}(\varphi) \text{ iff } Prob_{s,a,}(\{\tau \in Paths(M_\alpha, \tau \models \varphi\}) \bowtie p$$
$$\text{for some scheduler } \alpha \text{ inducing a DTMC } M_\alpha$$

Example. $\mathbb{P}^{\exists}_{=0.5}(\Diamond a)$: With 50% probability, there exists a path such that somewhere on that path a holds.

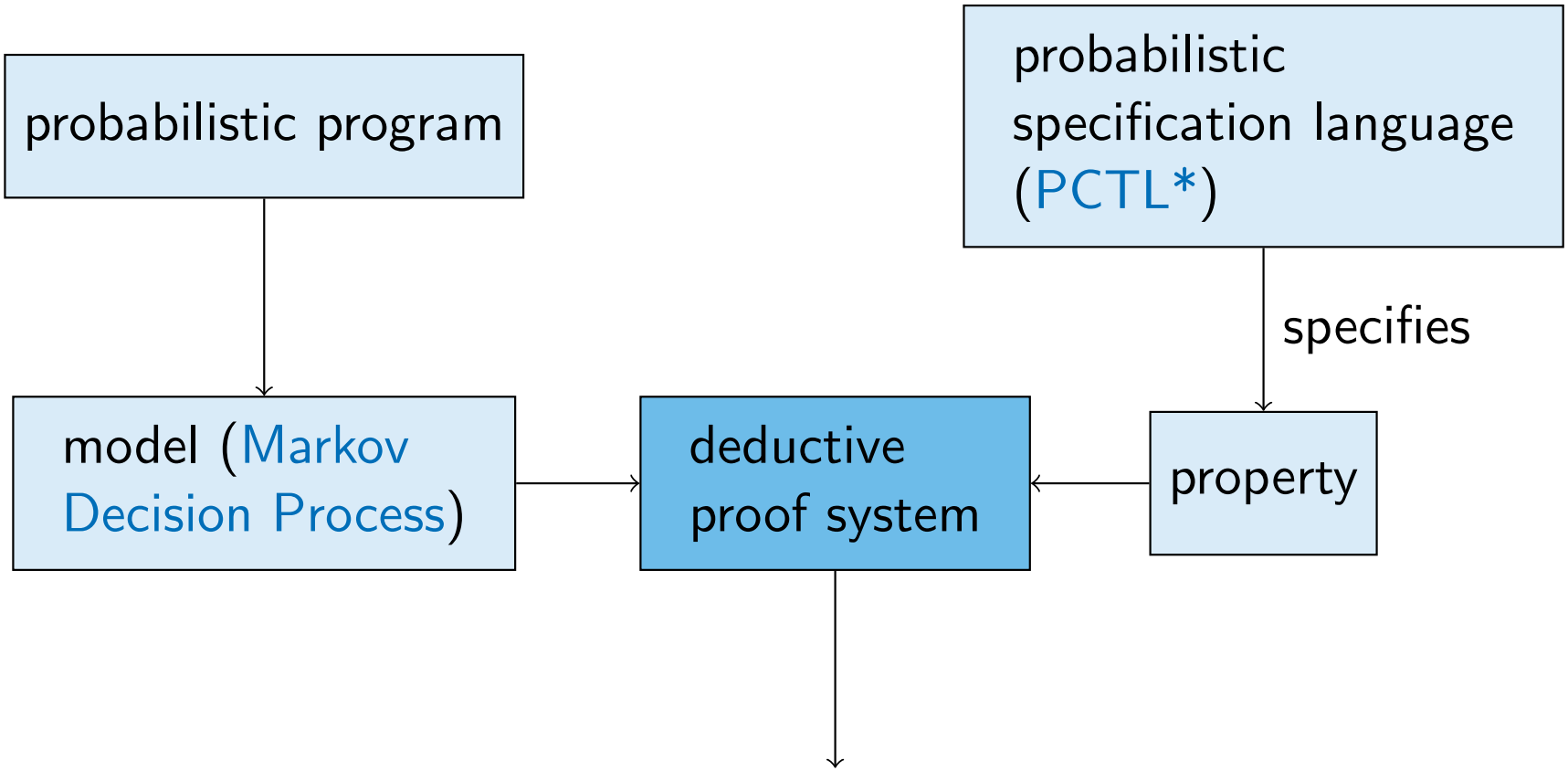
Probabilistic CTL*

PCTL* is a generalization of PCTL which additionally allows ω-regular languages over state formulas as path formulas.

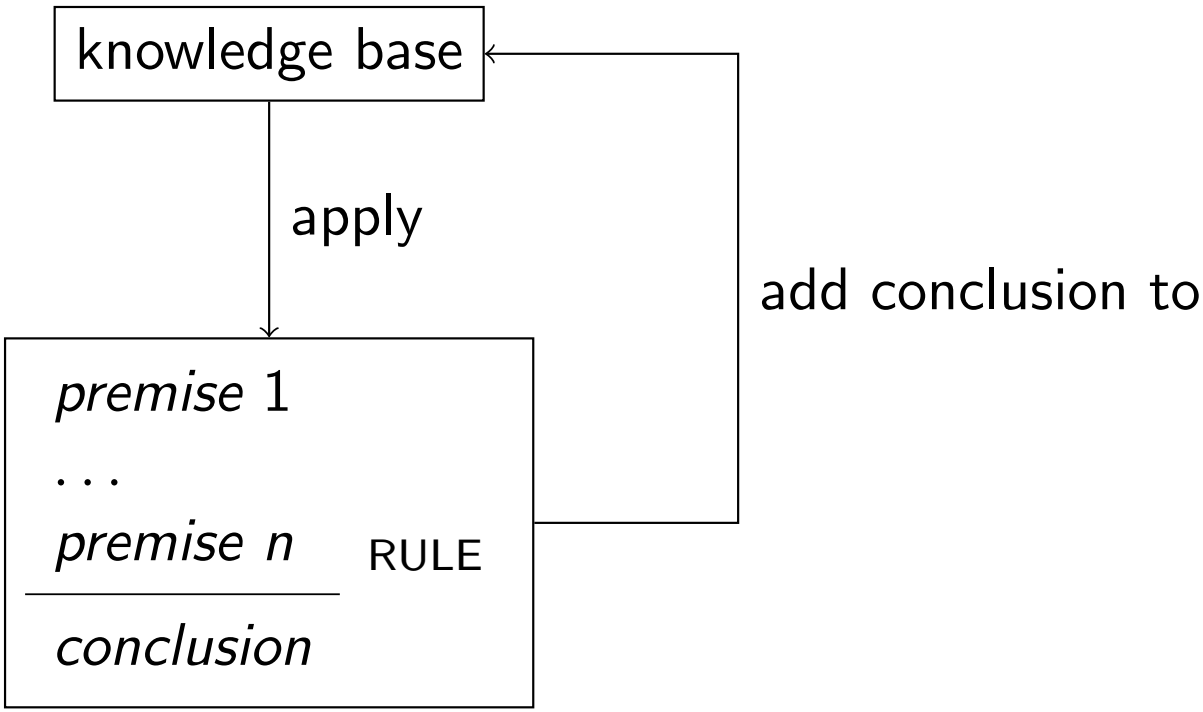
In PCTL/CTL, every temporal operator must be preceded by a quantifier. This restriction is removed in PCTL*/CTL*.

Example. $\Phi = \mathbb{P}^{\exists}_{=1}\Box\Diamond(a)$ is in PCTL*, but not in PCTL.
Φ is satisfied if there exists with probability 1 (almost sure) an infinite path such that a occurs infinitely often on that path.

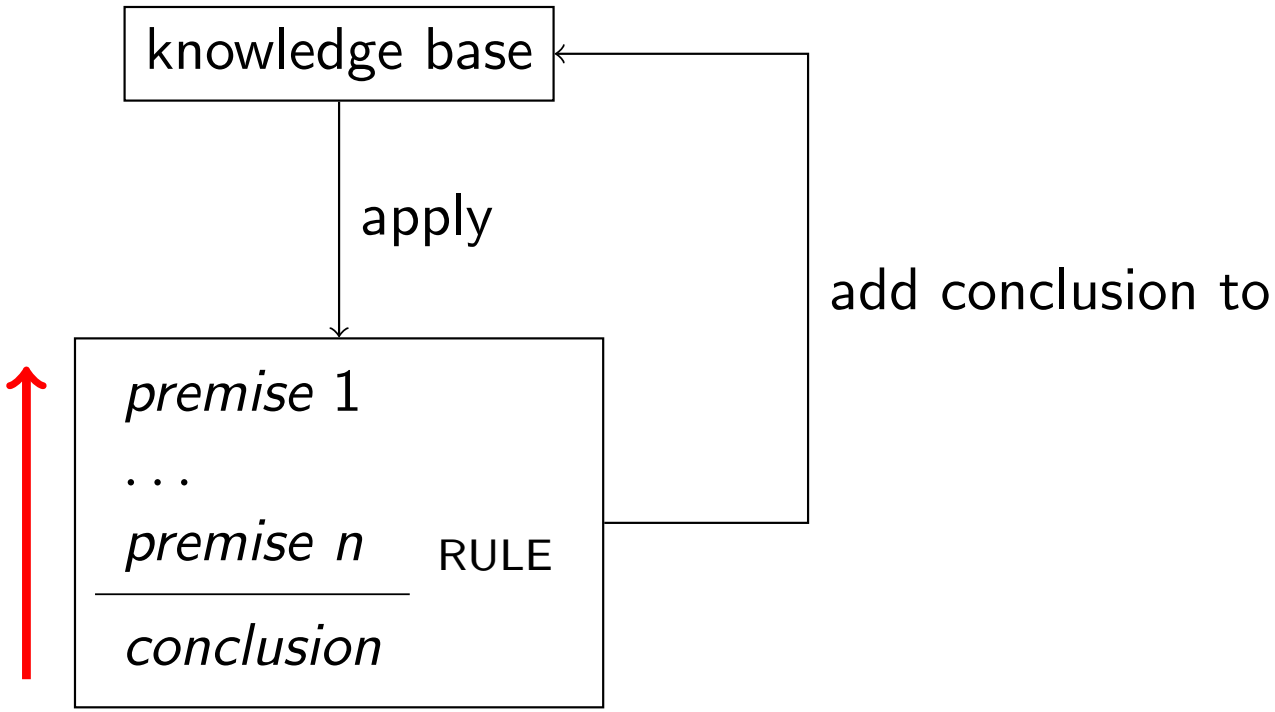
Overview


probabilistic program
probabilistic specification language (PCTL*)
specifies
model (Markov Decision Process)
deductive proof system
property
Does the program satisfy the property?

Deductive Proof System



Deductive Proof System



actual proof of a given property: bottom up

Deductive Proof System For PCTL*

→ Preliminary Rules

- Qualitative PCTL Rules
- Quantitative PCTL Rules
- Extension To PCTL*

Preliminary Rules

$P \vdash \Phi$: "The proof system derives that P satisfies Φ from every state"

$$\frac{\begin{array}{l}\text{assertion } \pi \\ P \vdash \Phi[\Psi/\pi] \\ P \vdash \pi \rightarrow \Psi\end{array}}{P \vdash \Phi} \text{ BASIC-STATE} \qquad \frac{\theta \text{ is a valid assertion}}{P \vdash \theta} \text{ GEN}$$

Basic Approach: Reduce the verification of a formula Φ to the verification of formulas of the form $\pi \rightarrow \Phi$.

Deductive Proof System For PCTL*

- Preliminary Rules
- → Qualitative PCTL Rules
  - qualitative fragment: $\mathbb{P}^{\mathcal{Q}}_{\bowtie p}$ restricted to $p \in \{0, 1\}$
- Quantitative PCTL Rules
- Extension To PCTL*

Lyapunov Ranking Functions

Extension of ranking functions to the probabilistic setting:

Given a well-founded partial order $(\mathbb{R}_{\geq 0}, \succ)$, a Lyapunov ranking function $\delta : S \to \mathbb{R}_{\geq 0}$ defined over a Markov Chain (S, ρ) is a function δ that decreases in expectation on each step:

$$\delta(s) \succ \mathbb{E}(\delta', s) = \sum_{s' \in S} \delta(s')\mu_s(s')$$

Proving Liveness Properties

- Lyapunov ranking functions are a tool for proving liveness properties of the form

$$\mathbb{P}^{\mathcal{Q}}_{\bowtie p}\Diamond\varphi = \mathbb{P}^{\mathcal{Q}}_{\bowtie p}(\mathsf{true}\,\mathcal{U}\,\varphi) \text{ for } \mathcal{Q} \in \{\exists, \forall\}$$

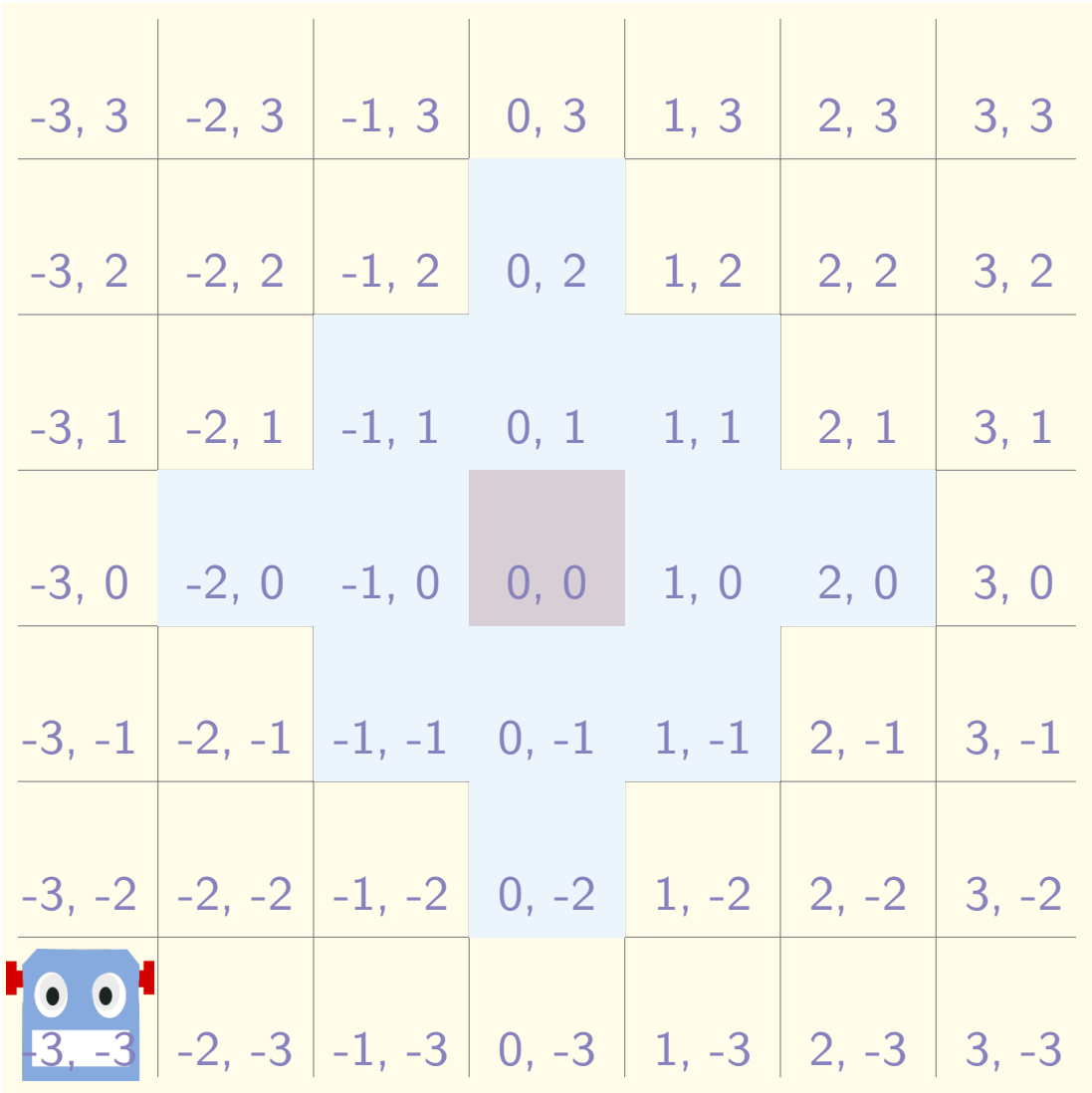
- Idea: find a δ that decreases whenever a transition $s \to s'$ is taken where the state s' does not satisfy the target formula φ
- this method is complete for finite state sets
- incomplete for infinite state sets

Rule $\text{UNTIL}^{\mathcal{Q}}_{=1}$

for proving almost sure liveness properties

$$\frac{\begin{array}{l}\text{assertion } \theta \\ \text{Lyapunov ranking function } \delta \\ P \vdash (\pi \wedge \neg\psi) \to \theta \\ P \vdash (\theta \wedge \neg\psi) \to \varphi \\ P \vdash (\theta \wedge \neg\psi) \to \\ (\mathcal{Q}c \in C : g_c : \theta' \wedge \delta \succ \mathbb{E}(\delta' | s))\end{array}}{P \vdash \pi \to \mathbb{P}^{\mathcal{Q}}_{=1}(\varphi \mathcal{U} \psi)} \; \text{UNTIL}^{\mathcal{Q}}_{=1}$$

Robot Example: Almost Sure Liveness



- Is there a strategy such that the robot eventually enters the area around the origin?

Robot Example: Almost Sure Liveness

$$P \vdash \mathit{true} \to \mathbb{P}^{\exists}_{=1} \Diamond \varphi_{\mathit{close}}, \qquad \varphi_{\mathit{close}} = |x| + |y| \leq 100$$

Robot Example: Almost Sure Liveness

$P \vdash \mathit{true} \rightarrow \mathbb{P}^{\exists}_{=1}\ \mathsf{true}\,\mathcal{U}\varphi_{\mathit{close}}, \quad \varphi_{\mathit{close}} = |x| + |y| \leq 100$

$$\frac{\begin{array}{l}\text{assertion } \theta \\ \text{Lyapunov ranking function } \delta \\ P \vdash (\pi \wedge \neg\psi) \rightarrow \theta \\ P \vdash (\theta \wedge \neg\psi) \rightarrow \varphi \\ P \vdash (\theta \wedge \neg\psi) \rightarrow \\ (\exists c \in C : g_c : \theta' \wedge \delta \succ \mathbb{E}(\delta' | s))\end{array}}{P \vdash \pi \rightarrow \mathbb{P}^{\exists}_{=1}(\varphi \mathcal{U} \psi)}\ \mathsf{UNTIL}^{\exists}_{=1}$$

Robot Example: Almost Sure Liveness

$P \vdash \mathit{true} \rightarrow \mathbb{P}^{\exists}_{=1}\ \mathsf{true}\, \mathcal{U} \varphi_{\mathit{close}}, \quad \varphi_{\mathit{close}} = |x| + |y| \leq 100$

Lyapunov ranking function δ

$P \vdash (\theta \wedge \neg\psi) \rightarrow$

$(\exists c \in C : g_c : \theta' \wedge \delta \succ \mathbb{E}(\delta' | s))$

Robot Example: Almost Sure Liveness

$P \vdash \mathit{true} \rightarrow \mathbb{P}^{\exists}_{=1}\ \mathsf{true}\,\mathcal{U}\varphi_{\mathit{close}}, \quad \varphi_{\mathit{close}} = |x| + |y| \leq 100$

Lyapunov ranking function δ

$P \vdash (\theta \wedge \neg\psi) \rightarrow$

$(\exists c \in C : g_c : \theta' \wedge \delta \succ \mathbb{E}(\delta'|s))$

in a state with $I = 0$, choose the command satisfying $x' = x - \mathit{sign}(x) \wedge y' = y - \mathit{sign}(y)$

Robot Example: Almost Sure Liveness

$P \vdash \mathit{true} \rightarrow \mathbb{P}^{\exists}_{=1}\ \mathsf{true}\,\mathcal{U}\varphi_{\mathit{close}}, \qquad \varphi_{\mathit{close}} = |x| + |y| \leq 100$

Lyapunov ranking function δ

$P \vdash (\theta \wedge \neg\psi) \rightarrow$

$(\exists c \in C : g_c : \theta' \wedge \delta \succ \mathbb{E}(\delta'|s))$

in a state with $l = 0$, choose the command satisfying $x' = x - \mathit{sign}(x) \wedge y' = y - \mathit{sign}(y)$

Robot Example: Almost Sure Liveness

$P \vdash \mathit{true} \rightarrow \mathbb{P}^{\exists}_{=1}\ \mathsf{true}\,\mathcal{U}\varphi_{\mathit{close}}, \quad \varphi_{\mathit{close}} = |x| + |y| \leq 100$

Lyapunov ranking function δ

$P \vdash (\theta \wedge \neg\psi) \rightarrow$

$(\exists c \in C : g_c : \theta' \wedge \delta \succ \mathbb{E}(\delta'|s))$

in a state with $l = 0$, choose the command satisfying $x' = x - \mathit{sign}(x) \wedge y' = y - \mathit{sign}(y)$

$$\delta(x, y, l) = \begin{cases} x^2 + y^2 & \text{if } l = 0 \\ x^2 + y^2 + 120 & \text{if } l = 1 \\ x^2 + y^2 + 60 & \text{if } l = 2 \end{cases}$$

Deductive Proof System For PCTL*

- Preliminary Rules
- Qualitative PCTL Rules
  - qualitative fragment: $\mathbb{P}^{\mathcal{Q}}_{\bowtie p}$ restricted to $p \in \{0, 1\}$

$\rightarrow$ Quantitative PCTL Rules

  - allows full range of probabilities $p \in \mathbb{R}_{\geq 0}$

- Extension To PCTL*

Rule $UNTIL^{\mathcal{Q}}_{\geq p^m}$

$$\frac{\begin{array}{l}\text{assertion } \theta,\ \text{ranking function } \delta \\ P \vdash \pi \wedge \neg\psi \to \theta \\ P \vdash \pi \wedge \neg\psi \to \delta \leq m \\ P \vdash \theta \wedge \neg\psi \to \varphi \\ P \vdash \theta \wedge \neg\psi \to (\mathcal{Q}c \in C : g_c : \\ \qquad (\delta' = \delta - 1 \wedge \theta') \otimes_{\geq p}\ \text{true})\end{array}}{P \vdash \pi \to \mathbb{P}^{\mathcal{Q}}_{\geq p^m}(\varphi \mathcal{U} \psi)}\ \text{UNTIL}^{\mathcal{Q}}_{\geq p^m}$$

the ranking function δ is initially lower bounded by m
decreases each step with probability at least p

Robot Example: Lower Bound

$$P \vdash \varphi_{close} \rightarrow \mathbb{P}^{\exists}_{\geq q}(\Diamond(x = 0 \wedge y = 0)), \quad \varphi_{close} = |x| + |y| \leq 100$$

Robot Example: Lower Bound

$$P \vdash \varphi_{close} \rightarrow \mathbb{P}^{\exists}_{\geq q}\mathsf{true}\, \mathcal{U}(x = 0 \wedge y = 0), \quad \varphi_{close} = |x| + |y| \leq 100$$

Robot Example: Lower Bound

$$P \vdash \varphi_{close} \rightarrow \mathbb{P}^{\exists}_{\geq q}\text{true}\, \mathcal{U}(x = 0 \wedge y = 0), \quad \varphi_{close} = |x| + |y| \leq 100$$

$$\frac{\begin{array}{l}\text{assertion } \theta, \text{ ranking function } \delta \\ P \vdash \pi \wedge \neg\psi \rightarrow \theta \\ P \vdash \pi \wedge \neg\psi \rightarrow \delta \leq m \\ P \vdash \theta \wedge \neg\psi \rightarrow \varphi \\ P \vdash \theta \wedge \neg\psi \rightarrow (\exists c \in C : g_c : \\ \qquad (\delta' = \delta - 1 \wedge \theta') \otimes_{\geq p} \text{ true})\end{array}}{P \vdash \pi \rightarrow \mathbb{P}^{\exists}_{\geq p^m}(\varphi \mathcal{U} \psi)} \text{ UNTIL}^{\exists}_{\geq p^m}$$

Robot Example: Lower Bound

$$P \vdash \varphi_{close} \rightarrow \mathbb{P}^{\exists}_{\geq q}\text{true}\, \mathcal{U}(x = 0 \wedge y = 0), \quad \varphi_{close} = |x| + |y| \leq 100$$

$$\frac{\begin{array}{l}\text{assertion } \theta, \text{ ranking function } \delta \\ P \vdash \pi \wedge \neg\psi \rightarrow \theta \\ P \vdash \pi \wedge \neg\psi \rightarrow \delta \leq m \\ P \vdash \theta \wedge \neg\psi \rightarrow \varphi \\ P \vdash \theta \wedge \neg\psi \rightarrow (\exists c \in C : g_c : \\ \qquad (\delta' = \delta - 1 \wedge \theta') \otimes_{\geq p} \text{ true})\end{array}}{P \vdash \pi \rightarrow \mathbb{P}^{\exists}_{\geq p^m}(\varphi \mathcal{U} \psi)} \text{ UNTIL}^{\exists}_{\geq p^m}$$

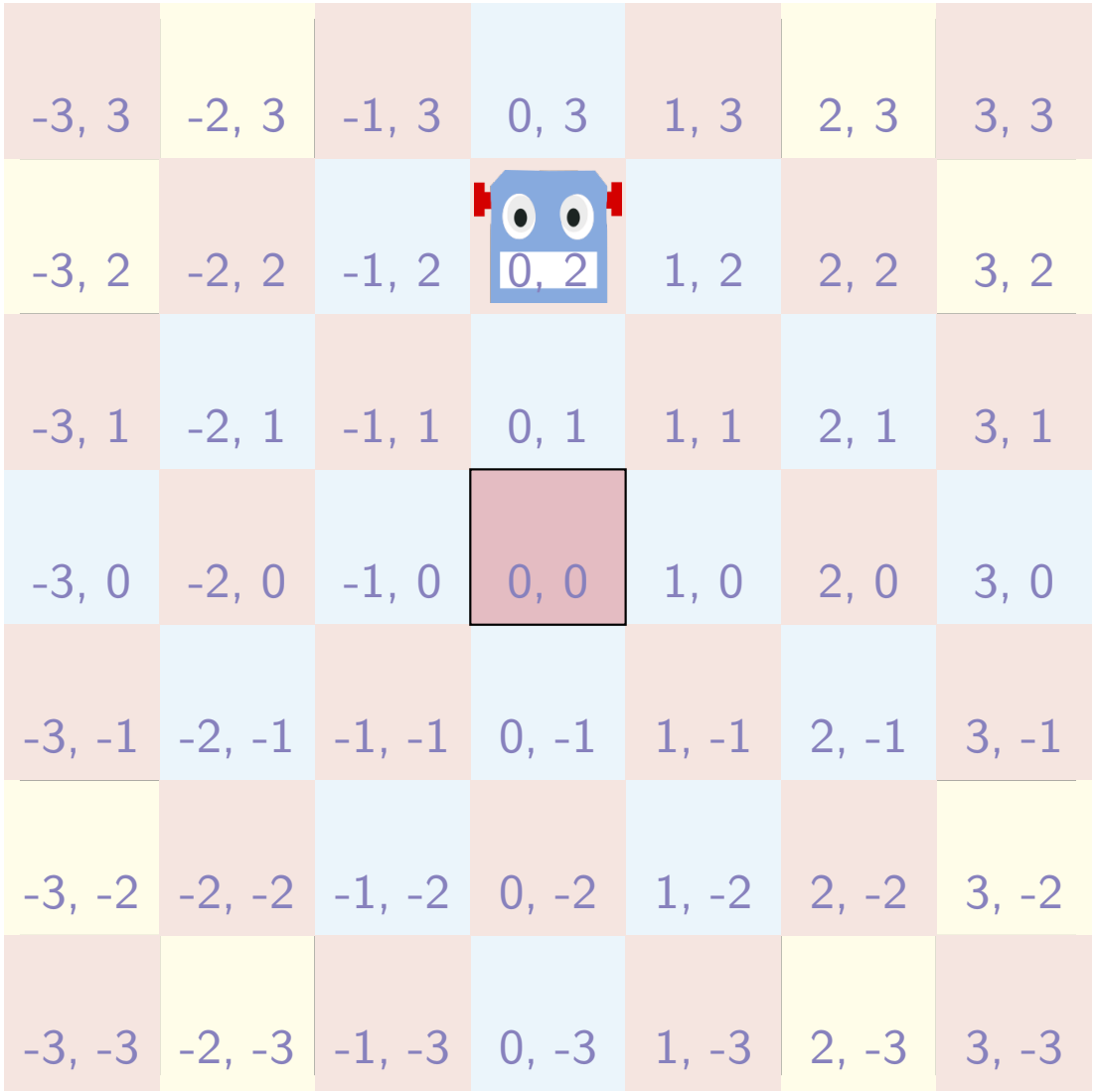
Robot Example: Lower Bound

$$P \vdash \varphi_{close} \rightarrow \mathbb{P}^{\exists}_{\geq q}\text{true}\ \mathcal{U}(x = 0 \wedge y = 0), \quad \varphi_{close} = |x| + |y| \leq 100$$

$$\frac{\begin{array}{l}\text{assertion } \theta,\ \text{ranking function } \delta \\ P \vdash \pi \wedge \neg\psi \rightarrow \theta \\ P \vdash \pi \wedge \neg\psi \rightarrow \delta \leq m \\ P \vdash \theta \wedge \neg\psi \rightarrow \varphi \\ P \vdash \theta \wedge \neg\psi \rightarrow (\exists c \in C : g_c : \\ \qquad (\delta' = \delta - 1 \wedge \theta') \otimes_{\geq p}\ \text{true})\end{array}}{P \vdash \pi \rightarrow \mathbb{P}^{\exists}_{\geq p^m}(\varphi \mathcal{U} \psi)}\ \text{UNTIL}^{\exists}_{\geq p^m}$$

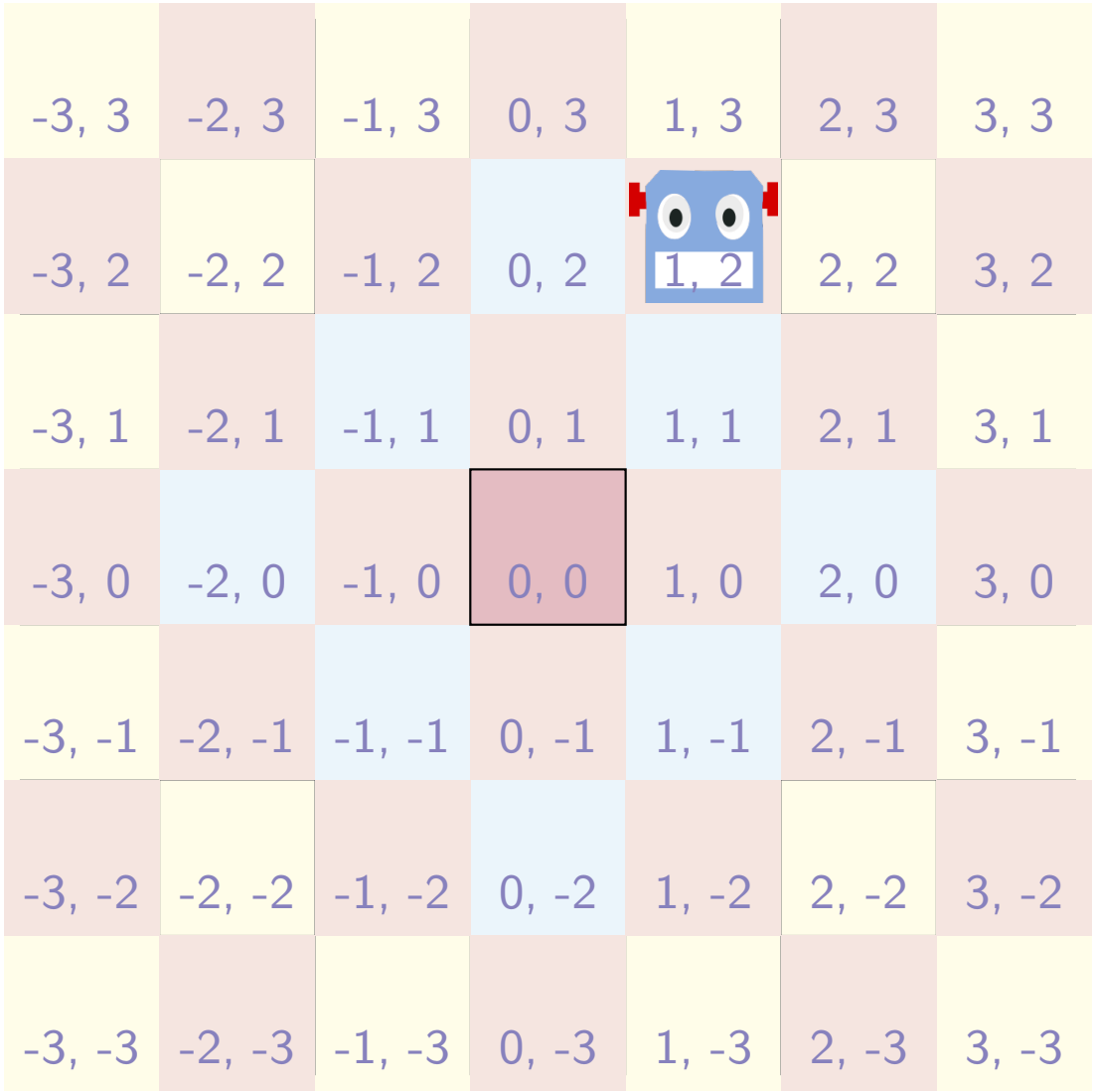
$\rightarrow$ need to construct a δ which satisfies the premises

Robot Example: Lower Bound



- Case 1:
  $x \equiv y \mod 2$

Robot Example: Lower Bound



- Case 2:
  $x \not\equiv y \mod 2$

Robot Example: Lower Bound

$$P \vdash \varphi_{close} \rightarrow \mathbb{P}^{\exists}_{\geq q}\text{true}\ \mathcal{U}(x = 0 \wedge y = 0), \quad \varphi_{close} = |x| + |y| \leq 100$$

$$\delta(l, x, y) = \begin{cases} \max(|x|, |y|) & \text{if } x \equiv y \mod 2 \text{ (Case 1)} \\ \max(|x|, |y|) + 5 & \text{if } x \not\equiv y \mod 2 \text{ (Case 2)} \end{cases}$$

- bounded $\delta(l, x, y) \leq 105 =: m$
- repel probability: $\frac{1}{|x|+1}, \frac{1}{|y|+1} \geq \frac{1}{101}$

Case 1: use appropriate command and no repel $(p_1 = (1 - \frac{1}{101})^2)$

Case 2: need repel along at least one axis to decrease δ (reach Case 1) $(p_2 = (1 - \frac{1}{101}) \cdot \frac{1}{101})$

- $p_1, p_2 \geq \frac{1}{101^2} := p$

Deductive Proof System For PCTL*

- Preliminary Rules
- Qualitative PCTL Rules
- Quantitative PCTL Rules

→ Extension To PCTL*

Extension To PCTL*

PCTL* is a generalization of PCTL which additionally allows ω-regular languages over state formulas as path formulas.

- until now: rules only consider formulas of the form $\mathbb{P}^{\mathcal{Q}}_{\bowtie p}\varphi$ where φ is a path formula which contains only one temporal operator
- in PCTL*: multiple temporal operators may appear in sequence
- Example: $\Phi = \mathbb{P}^{\exists}_{=1}\Box\Diamond(a)$

! $\Box\Diamond(a)$ is still an LTL formula

$\rightarrow$ LTL formula can be converted to deterministic ω-automata (in particular: Streett Automata)

Encoded Streett Acceptance Condition

- Street Automaton $\mathcal{A} = (Q, \Sigma, \rho, q_0, \{(E_i, F_i)_{i=1}^k\})$

Encoded Streett Acceptance Condition

- Street Automaton $\mathcal{A} = (Q, \Sigma, \rho, q_0, \{(E_i, F_i)_{i=1}^k\})$
- An (inifinite) run η on $\mathcal{A}$ is considered **accepting** if it holds that if $\mathit{Inf}(\eta) \cap E_i \neq \emptyset$, then also $\mathit{Inf}(\eta) \cap F_i \neq \emptyset$.
- $\mathit{Inf}(\eta)$:= set of states occurring infinitely often in η

Encoded Streett Acceptance Condition

- Street Automaton $\mathcal{A} = (Q, \Sigma, \rho, q_0, \{(E_i, F_i)_{i=1}^k\})$
- An (inifinite) run η on $\mathcal{A}$ is considered **accepting** if it holds that if $\mathit{Inf}(\eta) \cap E_i \neq \emptyset$, then also $\mathit{Inf}(\eta) \cap F_i \neq \emptyset$.
- $\mathit{Inf}(\eta)$:= set of states occurring infinitely often in η
- Encoded as a PCTL* formula (recurrence property):

$$\bigwedge_{i=1}^{k} (\Box\Diamond\varphi_i \rightarrow \Box\Diamond\psi_i)$$

for assertion φ_i and ψ_i that encode the sets E_i and F_i.

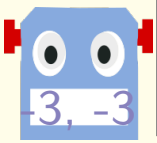
Rule $REC^{\exists}_{=1}$

for resolving (almost sure) existential recurrence properties

$$\frac{\begin{array}{l}\text{assertions } \theta, \overline{\theta} \\ \text{constant } p > 0 \\ P \vdash \pi \to \overline{\theta} \\ P \vdash \overline{\theta} \to \mathbb{P}^{\exists}_{=1}(\Diamond\theta) \\ P \vdash \theta \to \mathbb{P}^{\forall}_{=1}(\Box\theta) \\ \text{for all } i = 1, \ldots, m: \\ P \vdash \theta \wedge \varphi^i \to \mathbb{P}^{\exists}_{\geq p}(\Diamond\psi^i)\end{array}}{P \vdash \pi \to \mathbb{P}^{\exists}_{=1}\big(\bigwedge_{i=1}^{m}(\Box\Diamond\varphi^i \to \Box\Diamond\psi^i)\big)} \; \text{REC}^{\exists}_{=1}$$

Robot Example: Infinite Path

-3, 3	-2, 3	-1, 3	0, 3	1, 3	2, 3	3, 3
-3, 2	-2, 2	-1, 2	0, 2	1, 2	2, 2	3, 2
-3, 1	-2, 1	-1, 1	0, 1	1, 1	2, 1	3, 1
-3, 0	-2, 0	-1, 0	0, 0	1, 0	2, 0	3, 0
-3, -1	-2, -1	-1, -1	0, -1	1, -1	2, -1	3, -1
-3, -2	-2, -2	-1, -2	0, -2	1, -2	2, -2	3, -2
-3, -3	-2, -3	-1, -3	0, -3	1, -3	2, -3	3, -3



- original goal: visit the origin (0,0) infinitely often
- can finally be encoded in PCTL* as a recurrence property

Robot Example: Infinite Path

PCTL* property $P \models \text{ true } \rightarrow \mathbb{P}^{\exists}_{=1}(\Box\Diamond(x = 0 \wedge y = 0))$

from previous examples:

$$P \vdash \text{true } \rightarrow \mathbb{P}^{\exists}_{=1}(\Diamond\varphi_{close})$$
$$P \vdash \varphi_{close} \rightarrow \mathbb{P}^{\exists}_{\geq p}(\Diamond(x = 0 \wedge y = 0))$$

Robot Example: Infinite Path

PCTL* property $P \models \text{ true } \rightarrow \mathbb{P}^{\exists}_{=1}(\Box\Diamond(x = 0 \wedge y = 0))$

from previous examples:

$$P \vdash \text{true } \rightarrow \mathbb{P}^{\exists}_{=1}(\Diamond\varphi_{close})$$
$$P \vdash \varphi_{close} \rightarrow \mathbb{P}^{\exists}_{\geq p}(\Diamond(x = 0 \wedge y = 0))$$
$$P \vdash \text{true } \rightarrow \mathbb{P}^{\exists}_{\geq p}(\Diamond(x = 0 \wedge y = 0))$$

Robot Example: Infinite Path

prove: $P \models \text{ true } \rightarrow \mathbb{P}^{\exists}_{=1}(\text{ true } \rightarrow \Box\Diamond(x = 0 \wedge y = 0))$

known: $P \vdash \text{true } \rightarrow \mathbb{P}^{\exists}_{\geq p}(\Diamond(x = 0 \wedge y = 0))$

assertions $\theta, \overline{\theta}$
constant $p > 0$
$P \vdash \pi \rightarrow \overline{\theta}$
$P \vdash \overline{\theta} \rightarrow \mathbb{P}^{\exists}_{=1}(\Diamond\theta)$
$P \vdash \theta \rightarrow \mathbb{P}^{\forall}_{=1}(\Box\theta)$
for all $i = 1, \ldots, m$:
$P \vdash \theta \wedge \varphi^i \rightarrow \mathbb{P}^{\exists}_{\geq p}(\Diamond\psi^i)$

$$\frac{\text{premises}}{P \vdash \pi \rightarrow \mathbb{P}^{\exists}_{=1}\big(\bigwedge_{i=1}^{m}(\Box\Diamond\varphi^i \rightarrow \Box\Diamond\psi^i)\big)}\ \text{REC}^{\exists}_{=1}$$

Robot Example: Infinite Path

prove: $P \models \text{ true } \rightarrow \mathbb{P}^{\exists}_{=1}(\text{ true } \rightarrow \Box\Diamond(x = 0 \wedge y = 0))$

known: $P \vdash \text{ true } \rightarrow \mathbb{P}^{\exists}_{\geq p}(\Diamond(x = 0 \wedge y = 0))$

assertions $\theta, \overline{\theta}$
constant $p > 0$
$P \vdash \pi \rightarrow \overline{\theta}$
$P \vdash \overline{\theta} \rightarrow \mathbb{P}^{\exists}_{=1}(\Diamond\theta)$
$P \vdash \theta \rightarrow \mathbb{P}^{\forall}_{=1}(\Box\theta)$
for all $i = 1, \ldots, m$:
$P \vdash \theta \wedge \varphi^i \rightarrow \mathbb{P}^{\exists}_{\geq p}(\Diamond\psi^i)$

$$\frac{}{P \vdash \pi \rightarrow \mathbb{P}^{\exists}_{=1}\left(\bigwedge_{i=1}^{m}(\Box\Diamond\varphi^i \rightarrow \Box\Diamond\psi^i)\right)} \; \text{REC}^{\exists}_{=1}$$

Robot Example: Infinite Path

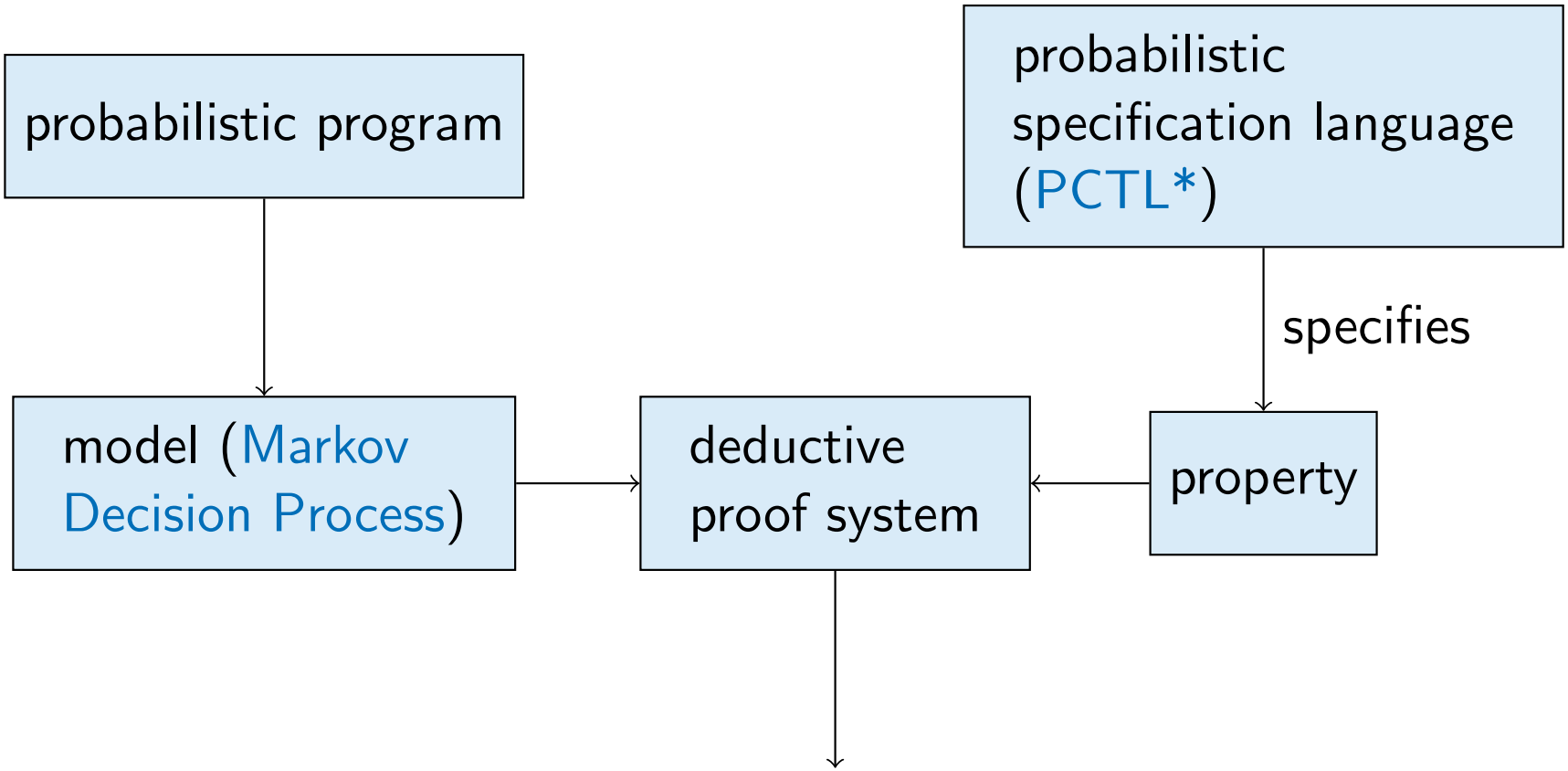
prove: $P \models \text{true} \rightarrow \mathbb{P}^{\exists}_{=1}(\text{true} \rightarrow \Box\Diamond(x = 0 \wedge y = 0))$

known: $P \vdash \text{true} \rightarrow \mathbb{P}^{\exists}_{\geq p}(\Diamond(x = 0 \wedge y = 0))$

assertions $\theta, \overline{\theta}$
constant $p > 0$
$P \vdash \pi \rightarrow \overline{\theta}$
$P \vdash \overline{\theta} \rightarrow \mathbb{P}^{\exists}_{=1}(\Diamond\theta)$
$P \vdash \theta \rightarrow \mathbb{P}^{\forall}_{=1}(\Box\theta)$
for all $i = 1, \ldots, m$:
$P \vdash \theta \wedge \varphi^i \rightarrow \mathbb{P}^{\exists}_{\geq p}(\Diamond\psi^i)$

$$\frac{}{P \vdash \pi \rightarrow \mathbb{P}^{\exists}_{=1}\big(\bigwedge_{i=1}^{m}(\Box\Diamond\varphi^i \rightarrow \Box\Diamond\psi^i)\big)} \; \text{REC}^{\exists}_{=1}$$

Summary


probabilistic program
probabilistic specification language (PCTL*)
specifies
model (Markov Decision Process)
deductive proof system
property
Does the program satisfy the property?

Summary

- Preliminary Rules
- Qualitative PCTL Rules
  - Lyapunov Ranking Functions (liveness properties)
- Quantitative PCTL Rules
- Extension To PCTL*
  - ω-regular languages
  - Streett acceptance condition encoded as recurrence property

Discussion

- sound proof system for PCTL*
- completeness: Lyapunov ranking functions only complete in finite state spaces
- automation:
  - how to generate ranking functions automatically?

References I