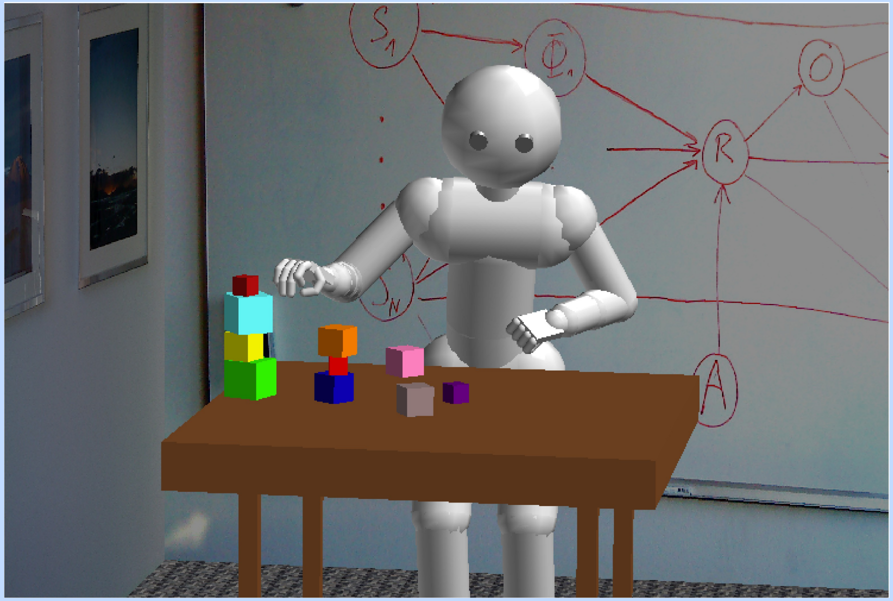
Tobias Lang & Marc Toussaint
Machine Learning and Robotics Group
Technische Universität Berlin

Approximate Inference for Planning in Stochastic Relational Worlds

ICML 2009



Stochastic Relational Worlds

- Simulator example

The Problem

- **Goal**: control an autonomous agent in an unknown environment for varying goals
- **Model-based approach**: learn a world model $P(s' \mid a, s)$ and use this model to plan actions

- Requirements for **world models**:
  - Noise
  - Stochastic action effects
  - Generalize to new situations
  - Learned from experience

$\rightarrow$ We employ noisy probabilistic relational rules.

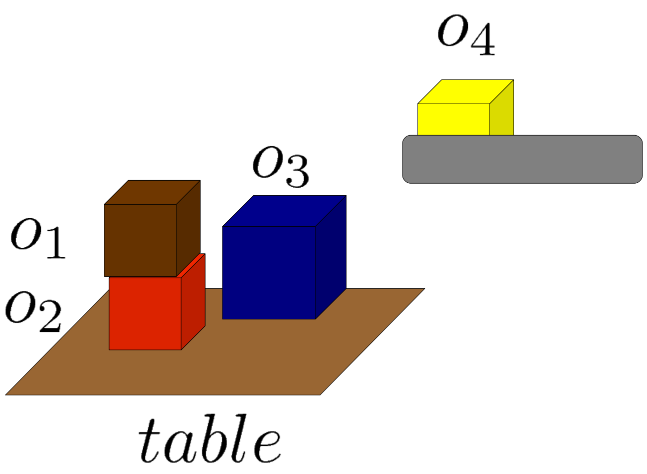
- Requirements for **planning**:
  - Fast
  - Robust
  - Varying goals

★ **Novel planning approach**

Background: Representation

- Symbolic relational representation
- States

$$on(o_1, o_2)$$
$$on(o_2, table)$$
$$on(o_3, table)$$
$$inhand(o_4)$$
$$size(o_3) = big$$



- Actions

$$grab(o_4)$$
$$puton(o_1)$$

Background: Relational rules

- Noisy indeterministic deictic rules (Pasula, Zettlemoyer and Kaelbling, 2007)

deictic reference

factorized

relational

action: $grab(X)$:

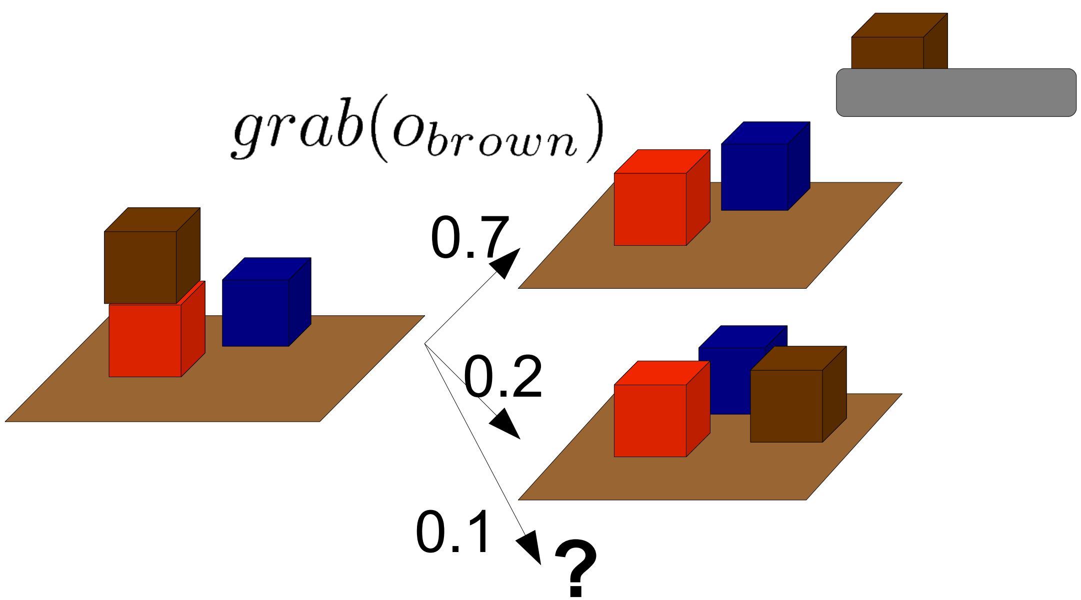
context: $on(X,Y),\ block(Y),\ table(Z)$

outcomes

$$\rightarrow \left\{ \begin{array}{lll} 0.7 & : & inhand(X),\ \neg on(X,Y) \\ 0.2 & : & on(X,Z),\ \neg on(X,Y) \\ 0.1 & : & \text{noise} \end{array} \right\} \text{indeterminism}$$

noise outcome

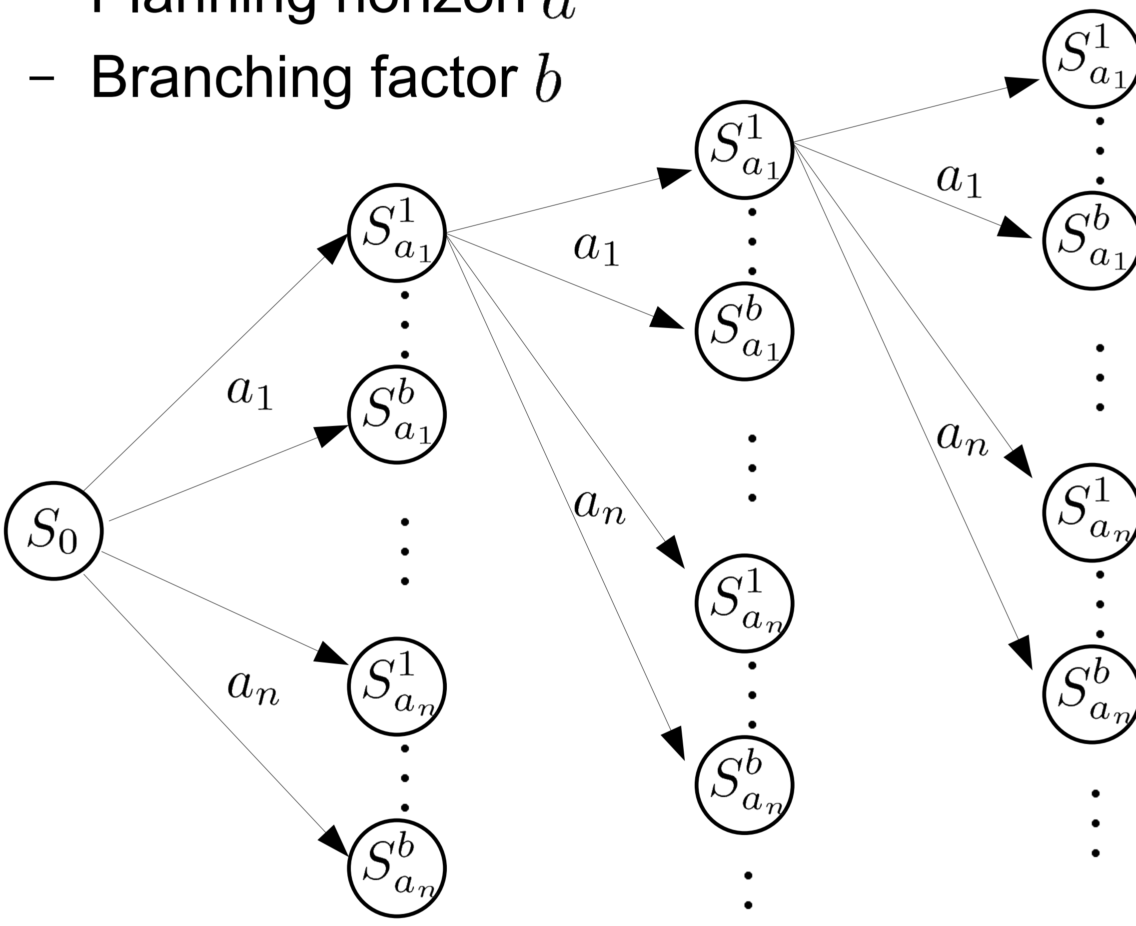
+ effective learning algorithm



no efficient planning method

Background: SST Planning

- Existing method for **planning** with NID rules: **sparse sampling trees** (SST) planning (Kearns et al., 2002)
  - Near optimal, but highly inefficient.
  - Planning horizon d
  - Branching factor b



Leaves at horizon d: $(ba)^d$

$a = 10, d = 4, b = 4 \rightarrow \sim 2.5 Mio.$

Our planning approach

- **PRADA**: **p**robabilistic **r**elational **a**ction-sampling
  in **d**ynamic Bayesian networks
  planning **a**lgorithm
- Plan in relational worlds by means of inference
- We sample action sequences and infer posteriors over hidden state variables.

(1) Convert NID rules to
dynamic Bayesian networks (DBNs)

(2) Approximate **inference** algorithm
to predict effects of action sequences

(3) **Informed sampling** strategy for action sequences

Convert NID rules to DBNs

- For rule-set Γ and set of objects O, ground all rules:

$$grab(X): \quad on(X,Y),\ block(Y),\ table(Z)$$
$$\rightarrow \begin{cases} 0.7 & : & inhand(X),\ \neg on(X,Y) \\ 0.2 & : & on(X,Z),\ \neg on(X,Y) \\ 0.1 & : & \text{noise} \end{cases}$$

$$O = \{o_1, o_2, o_3, \dots\}$$

$$\Gamma(O)_1: \quad X \rightarrow o_1, Y \rightarrow o_2, Z \rightarrow o_3$$
$$grab(o_1): \quad on(o_1,o_2),\ block(o_2),\ table(o_3)$$
$$\rightarrow \begin{cases} 0.7 & : & inhand(o_1),\ \neg on(o_1,o_2) \\ 0.2 & : & on(o_1,o_3),\ \neg on(o_1,o_2) \\ 0.1 & : & \text{noise} \end{cases}$$

Context random variable

State random variable

Outcome random variable

$$\Gamma(O)_2: \quad X \rightarrow o_1, Y \rightarrow o_3, Z \rightarrow o_2$$
$$grab(o_1): \quad on(o_1,o_3),\ block(o_3),\ table(o_2)$$
$$\rightarrow \begin{cases} 0.7 & : & inhand(o_1),\ \neg on(o_1,o_3) \\ 0.2 & : & on(o_1,o_2),\ \neg on(o_1,o_3) \\ 0.1 & : & \text{noise} \end{cases}$$

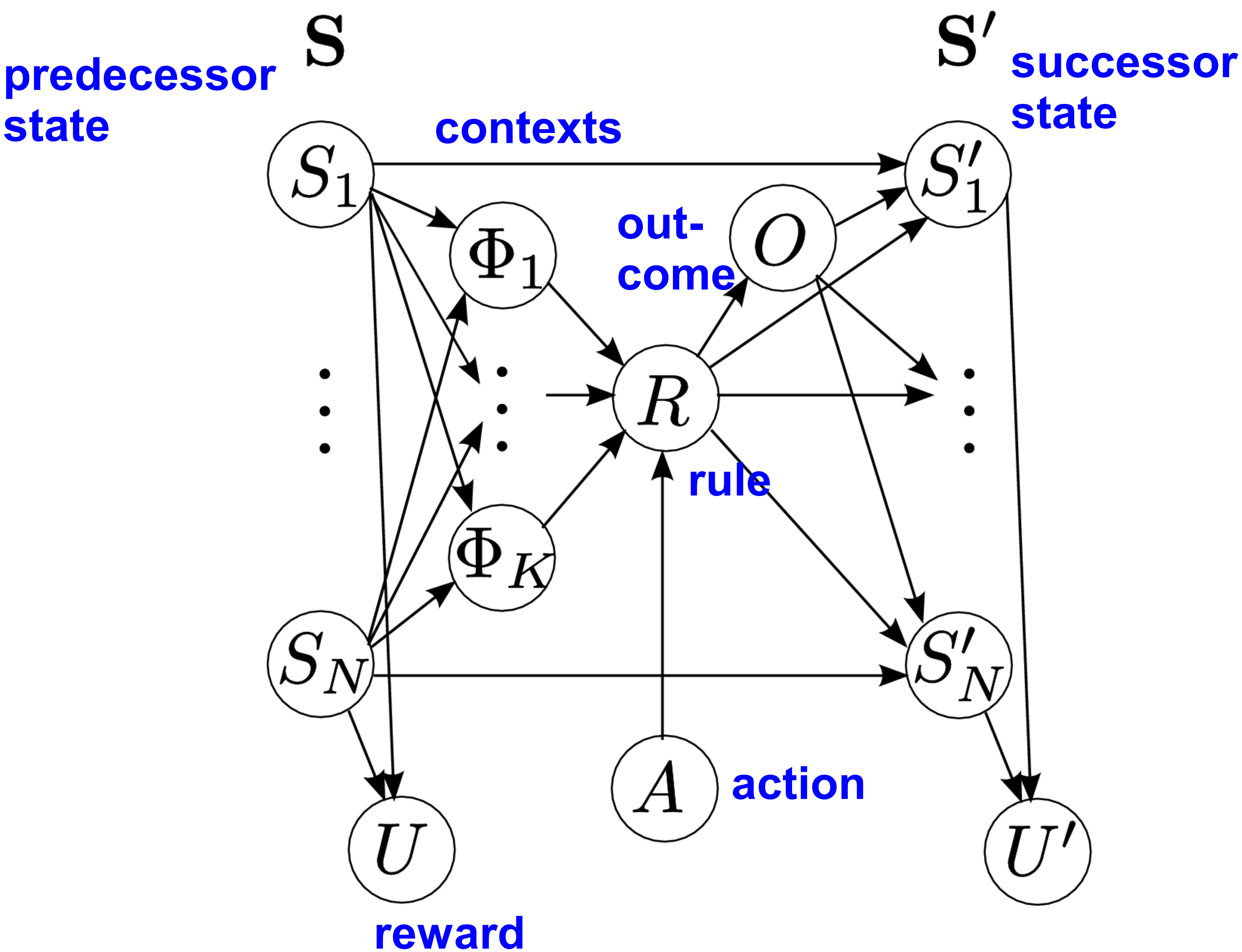
Action random variable

Rule random variable

etc.

Convert NID rules to DBNs

- DBN model for *K* ground rules



Approximate Inference

- **Exact inference** is **intractable** in our graphical model.
- Idea of the **factored frontier** algorithm (Murphy & Weiss, 2001): approximate belief with a product of marginals

$$P(\mathbf{s}^t \,|\, \mathbf{a}^{0:t-1}) \approx \prod_i P(s_i^t \,|\, \mathbf{a}^{0:t-1})$$

- Based on this approximation, we derive a **filter method** to **propagate action effects forward**:

$$P(\mathbf{s}^t \,|\, \mathbf{a}^{0:t-1}) \times a^t \quad \rightarrow \quad P(\mathbf{s}^{t+1} \,|\, \mathbf{a}^{0:t})$$

Approximate Inference

- Let $\alpha(s_i^t) := P(s_i^t \,|\, \mathbf{a}^{0:t-1})$ and $\alpha(\mathbf{s}^t) := P(\mathbf{s}^t \,|\, \mathbf{a}^{0:t-1}) \approx \prod_{i=1}^{N} \alpha(s_i^t)$.
- We calculate:

$$\alpha(s_i^{t+1}) = \sum_{r^t} P(s_i^{t+1} \,|\, r^t, \mathbf{a}^{0:t-1}) \; P(r^t \,|\, \mathbf{a}^{0:t})$$

$$P(s_i^{t+1} \,|\, r^t, \mathbf{a}^{0:t-1}) \approx \sum_{s_i^t} P(s_i^{t+1} \,|\, r^t, s_i^t) \; \alpha(s_i^t)$$

$$\begin{aligned} P(R^t = r \,|\, \mathbf{a}^{0:t}) = I(r \in \Gamma(a^t)) \;\; & P(\Phi_r^t = 1 \,|\, \mathbf{a}^{0:t-1}) \\ \cdot P(& \bigwedge_{r' \in \Gamma(a^t) \setminus \{r\}} \Phi_{r'}^t = 0 \,|\, \Phi_r^t = 1, \mathbf{a}^{0:t-1}) \end{aligned}$$

$$P(U^t = 1 \,|\, \mathbf{a}^{0:t-1}) \approx \prod_{i \in \pi(U^t)} \alpha(S_i^t = \tau_i)$$

Informed action sequence sampling

- Informed **sampling strategy**: sample "sensible" action sequences $\mathbf{a}^{0:T-1}$ with high probability

$$P^t_{sample}(a) \propto \sum_{r \in \Gamma(a)} P(\phi^t_r = 1, \bigwedge_{r' \in \Gamma(a) \setminus \{r\}} \phi^t_{r'} = 0 \,|\, \mathbf{a}^{0:t-1})$$

- Compute **posteriors over rewards** by means of approximate inference

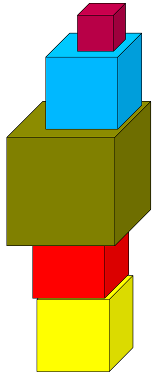
$$Q(\mathbf{a}^{0:T-1}, \mathbf{s}^0) := \sum_{t=1}^{T} \gamma^t P(U^t = 1 \,|\, \mathbf{a}^{0:t-1}, \mathbf{s}^0)$$

- Choose first action of best action sequence $\mathbf{a}^*$

- An extension: **Adaptive PRADA**
  - Can $\mathbf{a}^*$ be further improved by deleting some actions?

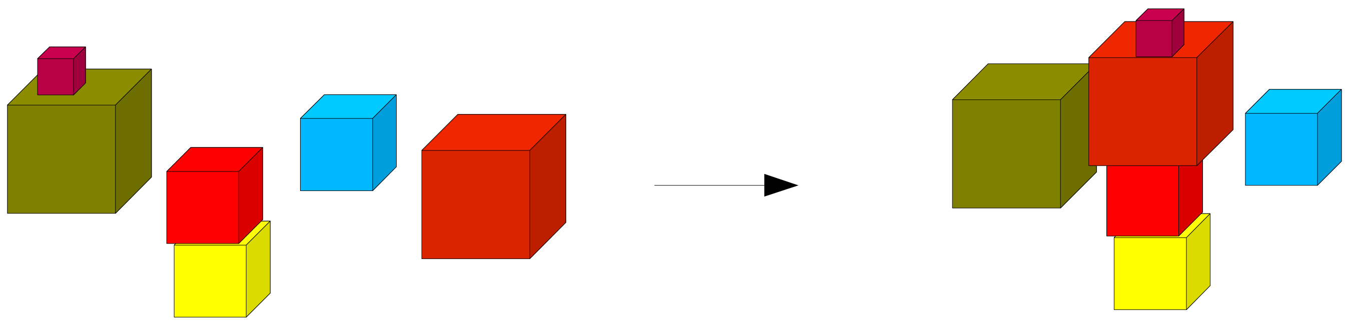
Results

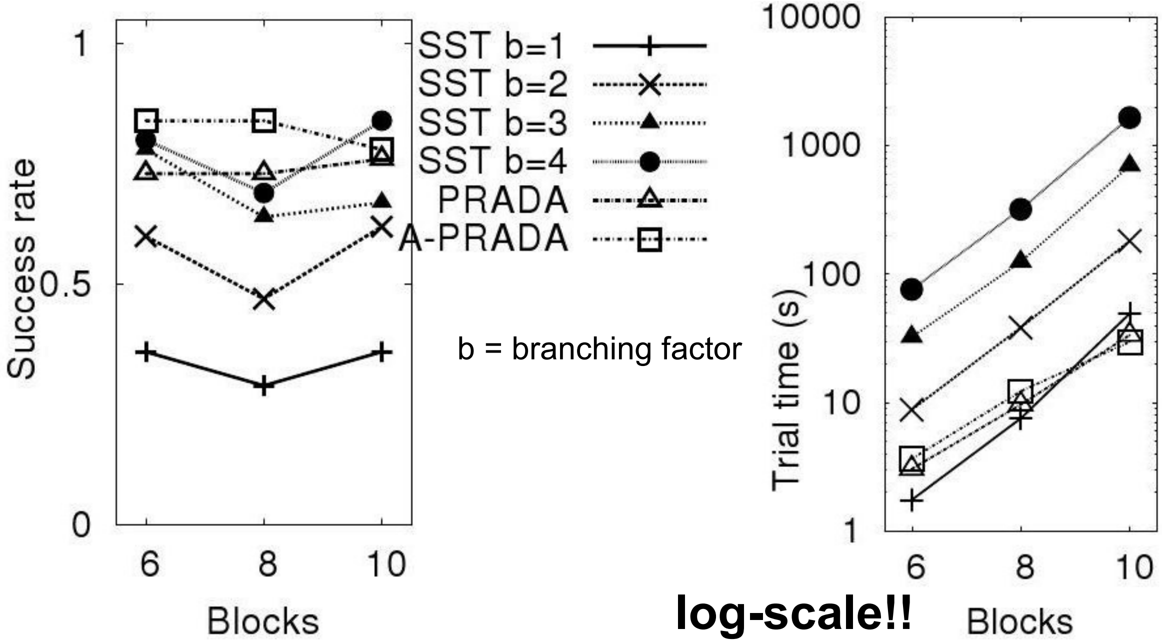
- 3 experiments with different planning goals
- Learn rule-sets in a world of 6 blocks
- Test worlds with **different blocks** and **block numbers**.
  → **Generalization** from training world to test worlds.
- For 10 objects:
  - Number of states $N_S > 2^{160}$
  - Number of actions $N_A = 21$
  - For planning horizon $d = 4$, number of possible action sequences: $N_A^d = 21^4 = 194481$



Results – Three specific blocks

- Build tower with three *specific* blocks.
- Can be achieved with four actions.



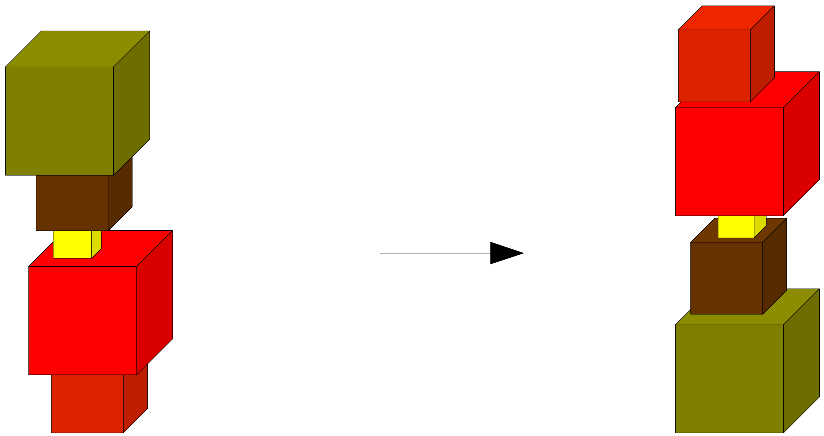


log-scale!!

SST either performs badly (small b) or is extremely slow (large b).

PRADA has **high performance** with **small planning time**!

Results – Reverse tower



Obj.	Planner	Suc.	Trial time (s)	Actions
5+1	SST (b=1)	0.0	-	-
5+1	SST (b=2)	0.0	> 1 day	-
5+1	PRADA	**0.84**	79.9±26.5	12.6±2.9
5+1	A-PRADA	0.78	**66.3±15.6**	10.6±1.4
6+1	PRADA	0.42	**184.9±51.9**	14.6±2.5
6+1	A-PRADA	**0.49**	190.4±49.8	12.8±1.7
7+1	PRADA	0.47	415.9±186.3	18.1±5.1
7+1	A-PRADA	**0.56**	**331.6±118.3**	14.8±1.8

Conclusions

- **Efficient planning** method
  for probabilistic relational rules
  based on **approximate inference**.

- Intelligent agent can now
  - **learn dynamics of complex stochastic world**
  - and **quickly derive appropriate actions** for **varying goals generalizing** to similar, but different worlds.

Thank you for your attention!

More information:
http://cs.tu-berlin.de/~lang/

References

- Pasula, H.M., Zettlemoyer, L.S., & Kaelbling, L.P. (2007): Learning symbolic models of stochastic domains. *Artificial Intelligence Research*, 29.

- Murphy, K.P., & Weiss, Y. (2001): The factored frontier algorithm for approximate inference in DBNs. *UAI Proceedings*.

- Kearns, M.J., Mansour, Y., & Ng., A.Y. (2002): A sparse sampling algorithm for near-optimal planning in large MDPs. *Machine Learning*, 49.