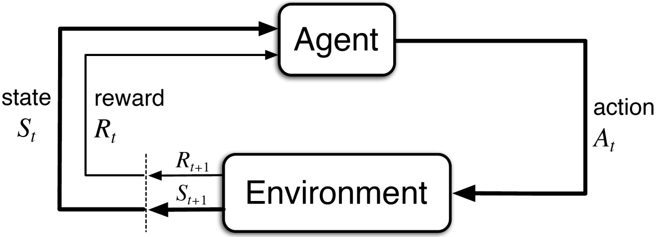
Regular Reinforcement Learning

Taylor Dohmen, Mateo Perez, Fabio Somenzi, & Ashutosh Trivedi

January 16, 2025

University of Colorado Boulder

Reinforcement Learning - A Brief Review



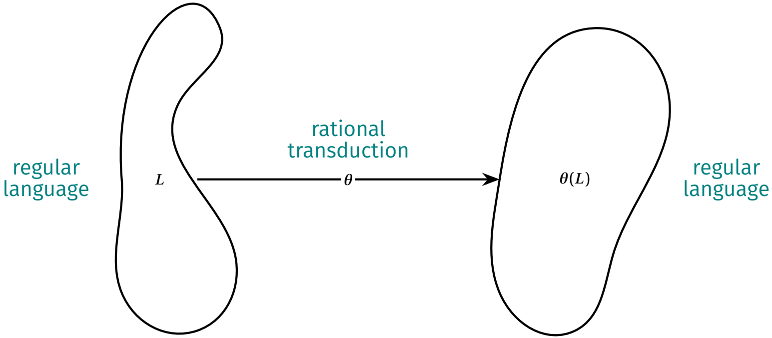
The generic RL feedback loop.*

*Figure credit to Sutton and Barto (Reinforcement learning - an introduction).

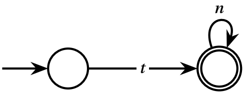
Inspiration — Regular Model Checking (RMC)

Infinite (non-stochastic) systems are modeled such that

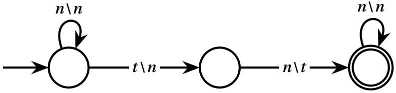
- states are expressed with regular languages,
- state transitions are expressed as rational transductions.



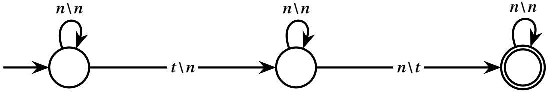
An RMC Example — Token Passing



automaton for initial language $I = t\,n^*$



transducer T passes token 1 index right

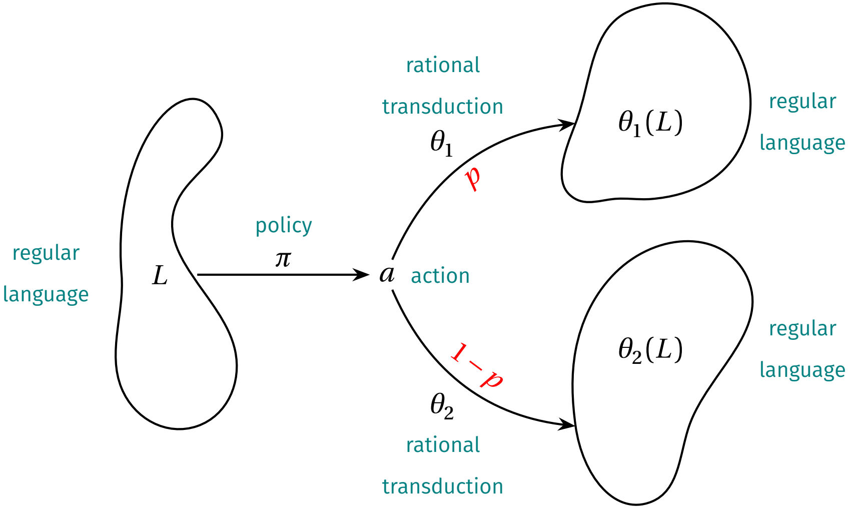


transducer T^+ moves the token rightwards by an arbitrary number of positions

$$\boxed{T^+(I) = n^*\,t\,n^*}$$

Reachable Language

Regular Markov Decision Processes (RMDPs)

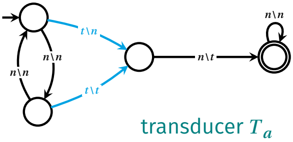


RMDP Example — A Variation on Token Passing

3 actions are available to the agent:

- (a) Each odd-index with a token passes it right.
  Each even-index with a token passes it right & generates itself a new token.
- (b) Each even-index with a token passes it right.
  Each odd-index with a token passes it right & generates itself a new token.
- (c) Mimics action (a) with probability p and mimics action (b) with probability $1-p$

$$\text{reward at } L = \begin{cases} 0 & \text{if } L \subseteq n^* t n^* \\ -1 & \text{otherwise} \end{cases}$$



Values of Arbitrary RMDPs

Theorem : General Undecidability

Whether a given RMDP satisfies any fixed non-trivial[†] property is undecidable.

Corollary : Non-Computable Values

Given an arbitrary RMDP, optimal values are not computable with respect to any fixed objective/payoff function.

[†]in the sense of Rice's theorem

Discounted Values of Computable RMDPs

An RMDP is called computable if the probabilities and rewards associated to each transition are computable.

Theorem : Approximability of Discounted Values

For any discount factor $\lambda \in [0, 1)$ and any tolerance $\epsilon > 0$, it is possible to compute an ϵ-approximation of the λ-discounted value from any state of a computable RMDP.

Theorem : PAC-Learnability of Discounted Values

For any discount factor $\lambda \in [0, 1)$, the λ-discounted value from any state of a computable RMDP is PAC-learnable.

Regular Reinforcement Learning (RRL) in Finitary RMDPs

An RMDP is finitary if either

- it has finitely many states, or
- it has finitely many distinct classes of reward-equivalent states.

Two states (languages) L_1 and L_2 are reward-equivalent, $L_1 \sim L_2$, iff

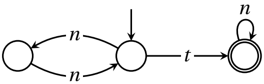
(1) rewards are equal at L_1 and L_2, and

(2) $\theta(L_1) \sim \theta(L_2)$ for every transduction θ.

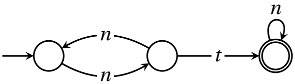
Q-learning for Finitary RMDPs

$$Q_{n+1}\big([L_n]_\sim, a_n\big) \;:=\; (1-\alpha_n)Q_n\big([L_n]_\sim, a_n\big) + \alpha_n\Big(r_n + \lambda \max_{a \in A} Q_n\big([L_{n+1}]_\sim, a\big)\Big)$$

Automata for reward-equivalence classes from the token passing RMDP.



recognizes $\bigcup_{k \in \mathbb{N}} n^{2k} t n^*$

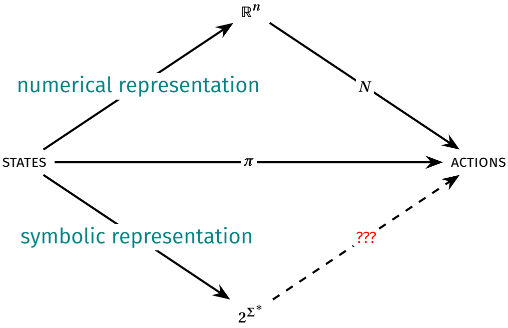


recognizes $\bigcup_{k \in \mathbb{N}} n^{2k+1} t n^*$

Traditionally, a policy π is approximated by a neural network N.

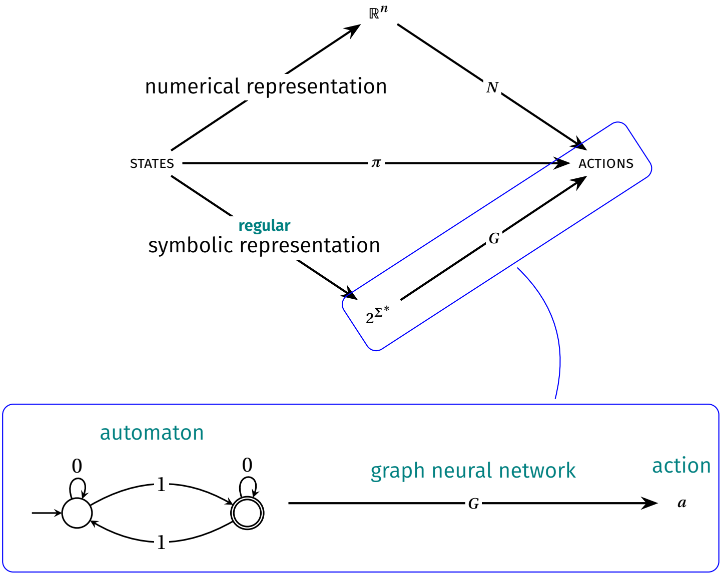
$$\begin{tikzcd}
 & \mathbb{R}^n \arrow[dr, "N"] & \\
\textsc{States} \arrow[ur, "\text{numerical representation}"] \arrow[rr, "\pi"] & & \textsc{Actions}
\end{tikzcd}$$

Symbolic Deep Reinforcement Learning

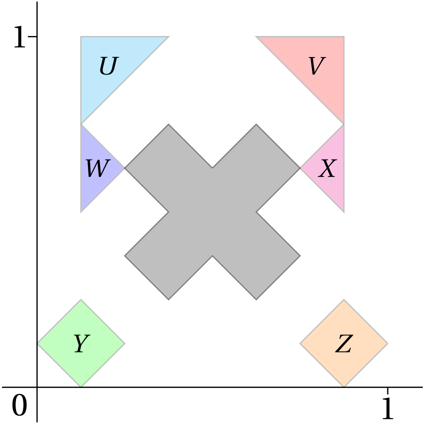


Can the deep RL approach be adapted around symbolic representation of states as formal languages?

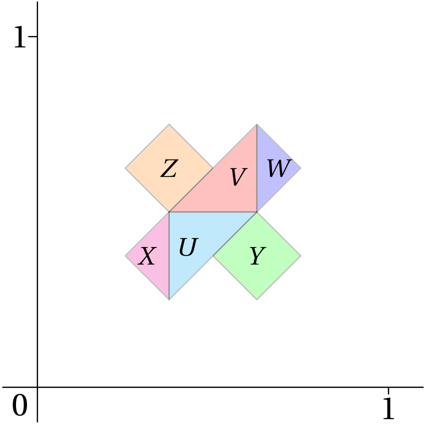
Deep Regular Reinforcement Learning



Deep Regular Reinforcement Learning — Modified Tangrams



initial configuration

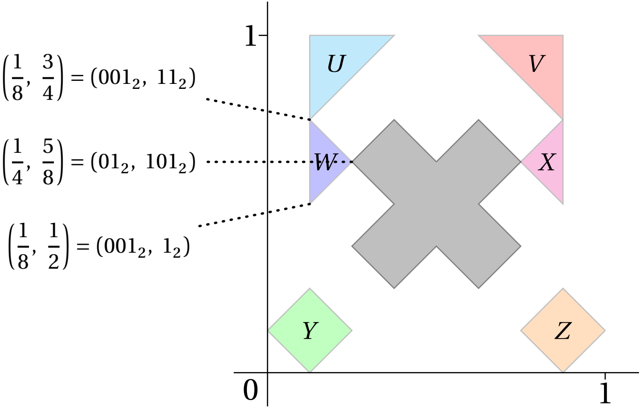


solution

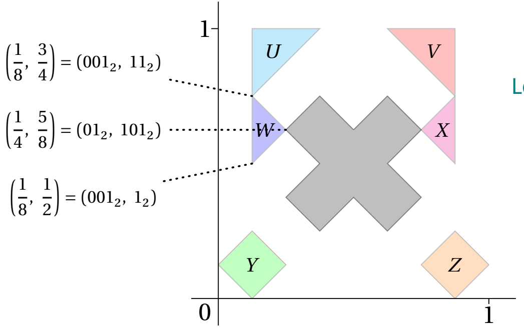
Deep Regular Reinforcement Learning — Modified Tangrams

For any $b \in \mathbb{N}$, each $w \in \{0, \dots, b-1\}^*$ encodes an element of $[0, 1]$ as

$$w_b = \sum_{k=1}^{|w|} \frac{w(k)}{b^k}.$$

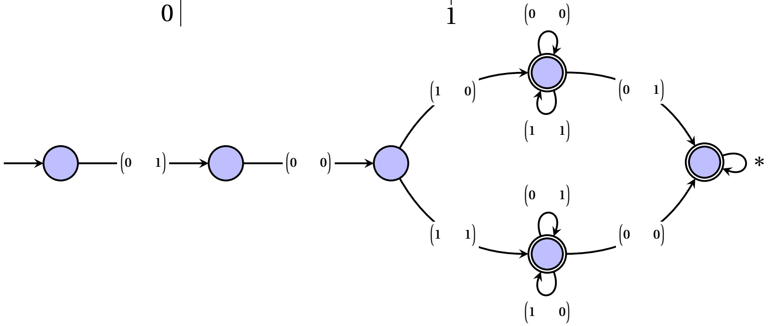


Deep Regular Reinforcement Learning — Modified Tangrams

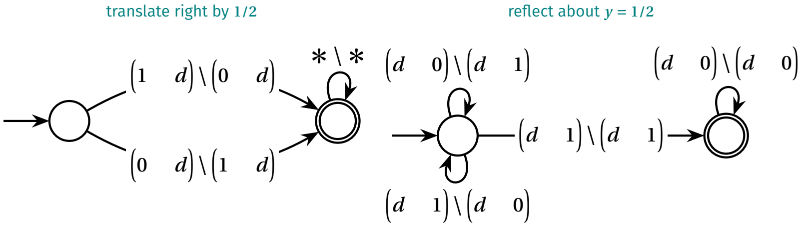


Let $w = x \otimes y$ iff $w(k) = (x(k), y(k))$.

The language
$\left\{x \otimes y \in \left(\{0,1\}^2\right)^* : (x_2, y_2) \in W\right\}$
is regular.



Deep Regular Reinforcement Learning — Modified Tangrams

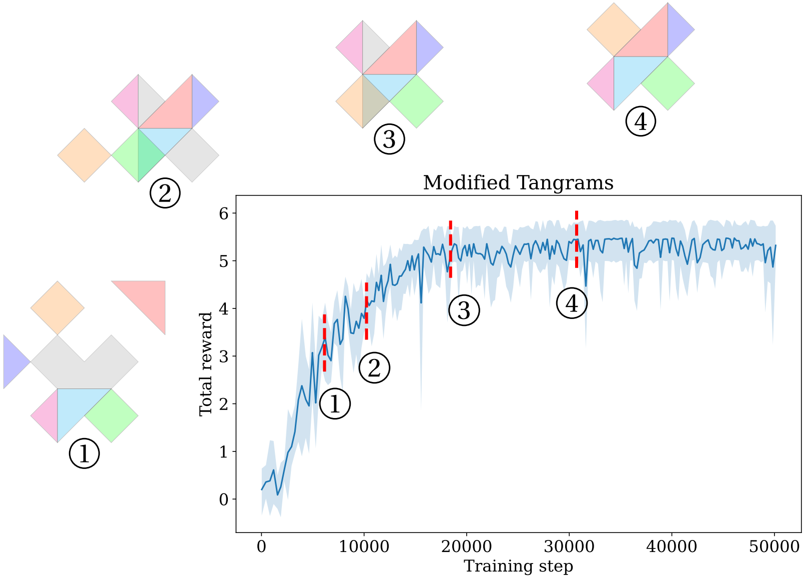


Transducers implementing rigid transformations on the unit square.

> Can an RL agent effectively solve tangram-style puzzles modeled as RMDPs?

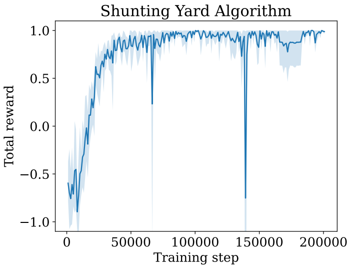
Digits d are arbitrary but must match on each side of \. $*$ represents arbitrary pairs of digits.

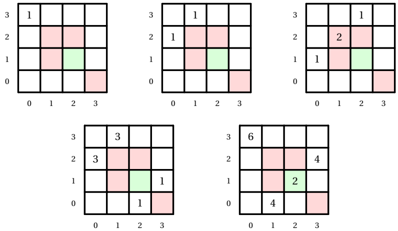
Deep Regular Reinforcement Learning — Modified Tangrams

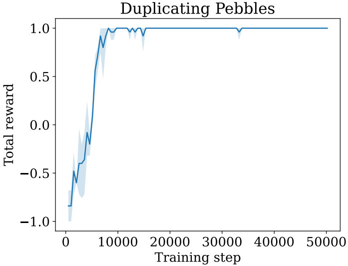


Further Experimental Studies

State	Action
7-1*3##	Move
-1*3##7	Push
1*3#-#7	Move
*3#-#71	Push
3#-*#71	Move
#-*#713	Pop
#-#713*	Pop
##713*-	







Questions?

Sutton, Richard S. and Andrew G. Barto. **Reinforcement learning - an introduction.** Adaptive computation and machine learning. MIT Press, 1998. ISBN: 978-0-262-19398-6. URL: https://www.worldcat.org/oclc/37293240.