Learning Dynamic Polynomial Proofs

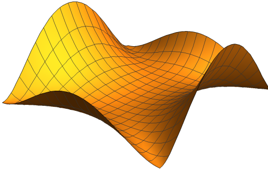
Alhussein Fawzi (DeepMind)
Mateusz Malinowski (DeepMind)
Hamza Fawzi (University of Cambridge)
Omar Fawzi (ENS Lyon)

Proving with polynomials

For a multivariate polynomial p, consider the following task:

Show that $p(x_1, \ldots, x_n) \geq 0$ for all $\boldsymbol{x} \in [0, 1]^n$

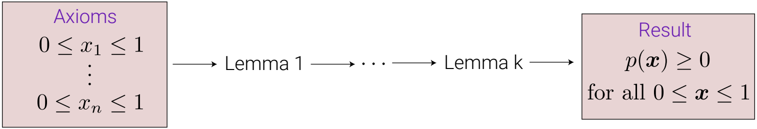
Many hard problems can be formulated as such; e.g., SAT, TSP, stable set, max-cut, etc.



Goal

Train an agent using Reinforcement Learning to prove the non-negativity of polynomials.

Proof system



- **Inference rules**:
    - $h \geq 0 \implies x_i h \geq 0,$
    - $h \geq 0 \implies (1 - x_i)h \geq 0,$
    - $h_i \geq 0 \implies \sum_i \lambda_i h_i \geq 0, \forall \lambda_i \geq 0.$
- **Proof of** $p \geq 0$: corresponds to the **composition** of inference rules, which yields exactly the polynomial p.

Learning dynamic proofs of polynomials

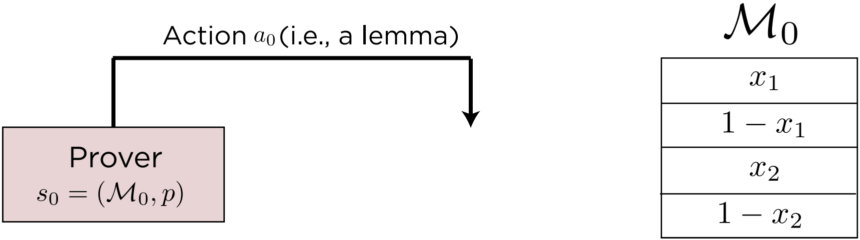
Prover

$s_0 = (\mathcal{M}_0, p)$

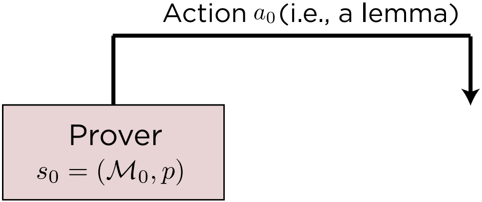
$\mathcal{M}_0$

x_1
$1 - x_1$
x_2
$1 - x_2$

Learning dynamic proofs of polynomials



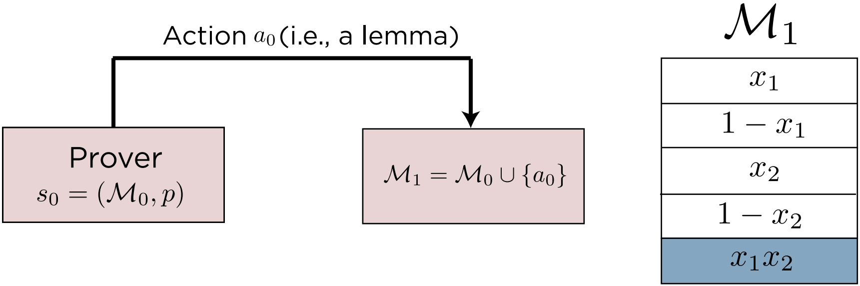
Learning dynamic proofs of polynomials



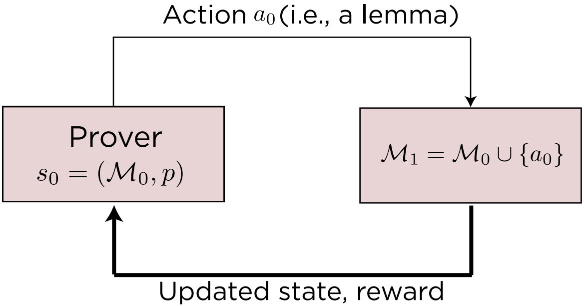
$\mathcal{M}_0$

	x_1
	$1 - x_1$
$x_1 \times$	x_2
	$1 - x_2$

Learning dynamic proofs of polynomials

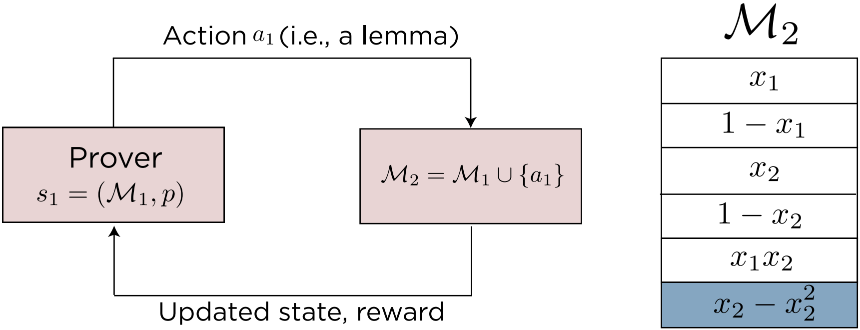


Learning dynamic proofs of polynomials

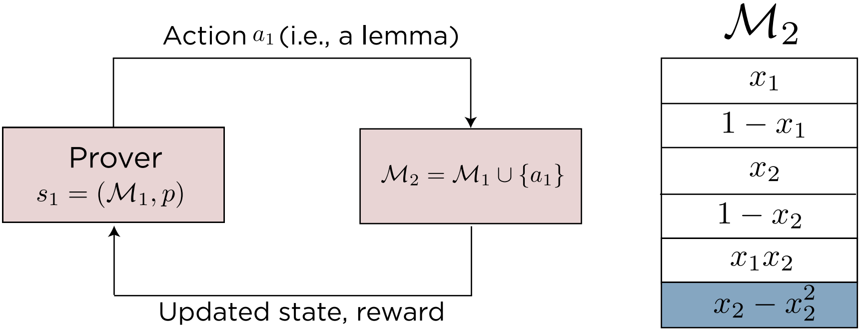


$\mathcal{M}_1$
x_1
$1 - x_1$
x_2
$1 - x_2$
$x_1 x_2$

Learning dynamic proofs of polynomials



Learning dynamic proofs of polynomials



We use DQN to train the prover. Two important ingredients:

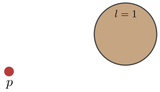
- We use **dense**, **unsupervised** rewards.
- We incorporate important **symmetries** in the Q-network.

Comparison to static approach

Static approach. Inference rules are unrolled for l steps.

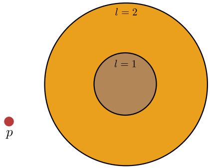
Comparison to static approach

Static approach. Inference rules are unrolled for l steps.



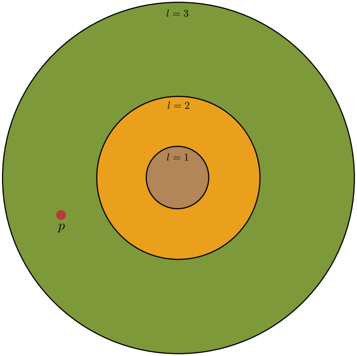
Comparison to static approach

Static approach. Inference rules are unrolled for l steps.



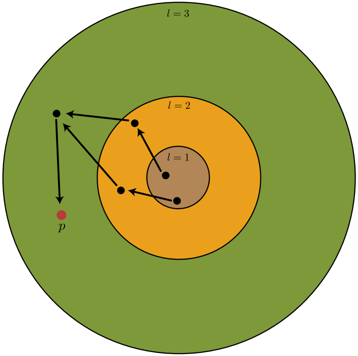
Comparison to static approach

Static approach. Inference rules are unrolled for l steps.



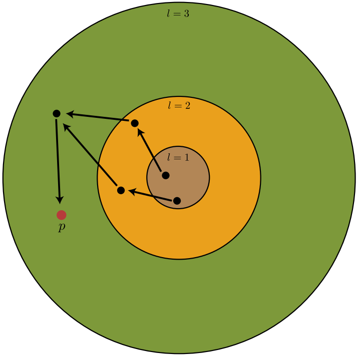
Comparison to static approach

Static approach. Inference rules are unrolled for l steps.



Comparison to static approach

Static approach. Inference rules are unrolled for l steps.



Key result. Reduction of the size of the linear program by several orders of magnitude compared to the static approach.

Come see our poster #120 for more details!