Strategy Representation by Decision Trees with Linear Classifiers

Pranav Ashok[1], Tomáš Brázdil[2], Krishnendu Chatterjee[3], Jan Křetínský[1], Christoph H. Lampert[3], and Viktor Toman[3]

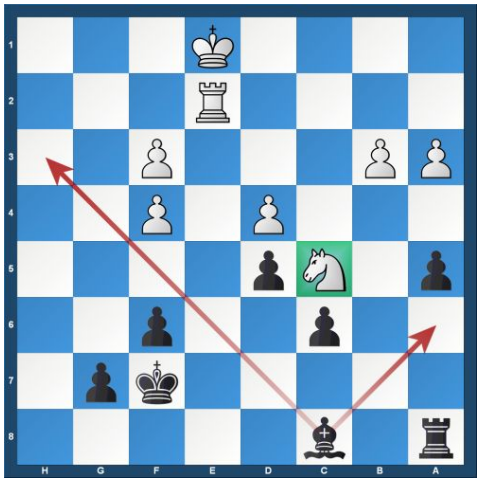
[1]Technical University of Munich, Munich, Germany
[2]Masaryk University, Brno, Czech Republic
[3]IST Austria, Klosterneuburg, Austria

Strategy

System:

S: state space
A: action space

Positional Strategy: resolve choices
σ: **S** □ **A** or
σ: **S** x **A** □ **{tt, ff}**



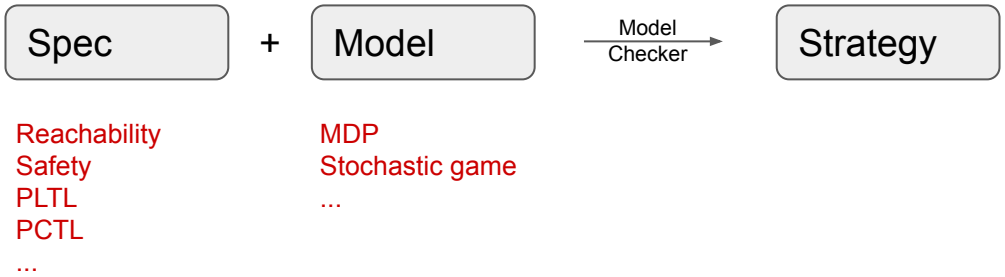
Focus: Synthesis of controllers

Synthesising controllers

Model Checking Approach

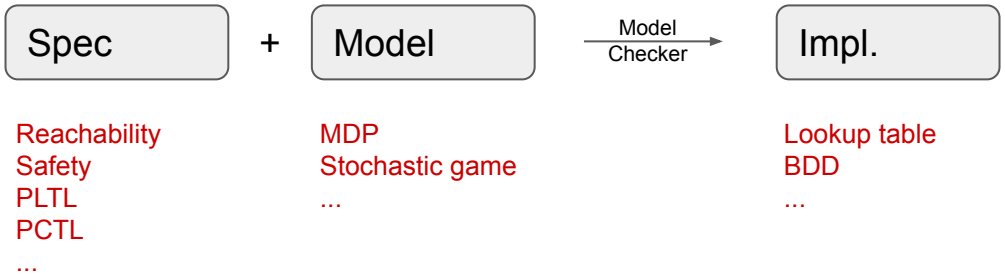
Synthesising controllers

Model Checking Approach



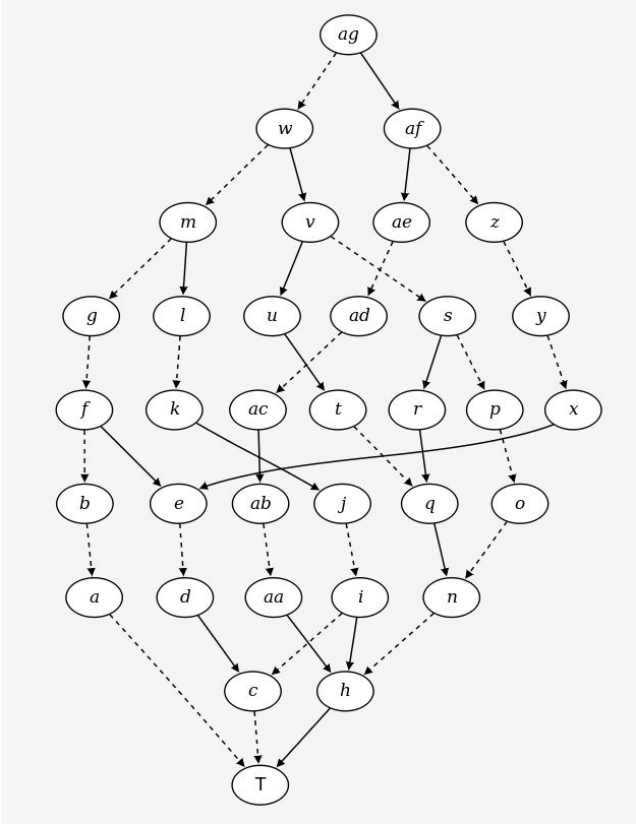
Synthesising controllers

Model Checking Approach



$(s, \sigma(s))$	$(s, a \neq \sigma(s))$
YES	NO
(0,1,0,1,0,0,0,0)	(0,1,0,1,1,0,1,0)
(0,0,0,0,1,0,1,0)	(0,0,0,0,1,0,1,1)
(0,0,0,0,0,0,0,0)	(0,1,0,1,0,0,1,0)
(0,0,0,0,0,0,0,1)	(1,0,0,0,1,0,1,1)
(1,0,0,0,0,0,0,1)	(0,1,0,1,1,0,1,1)
(1,0,0,0,1,0,1,0)	(0,1,1,0,1,0,1,0)
(0,1,0,1,1,1,0,1)	(0,0,0,0,1,1,1,0)
(0,0,1,0,1,0,1,1)	(0,1,0,1,0,0,0,1)
(1,1,0,0,1,0,1,1)	(0,1,1,1,0,0,0,1)
(0,1,1,1,0,1,0,1)	(0,0,1,0,0,0,0,1)
...	...

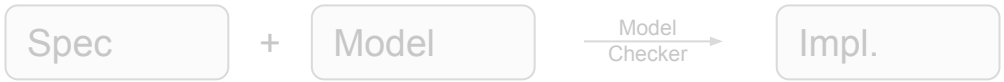
Lookup table



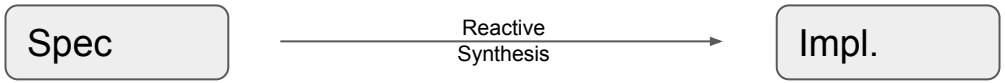
Binary Decision Diagram (BDD)

Synthesising controllers

Model Checking Approach

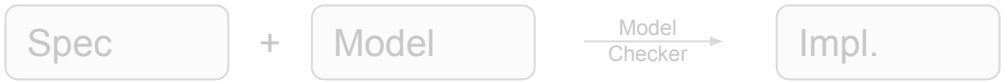


Reactive Synthesis Approach

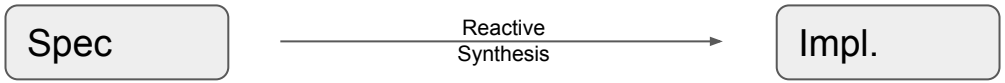


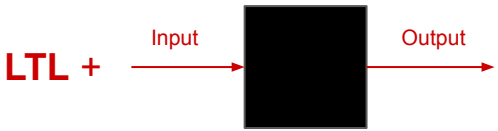
Synthesising controllers

Model Checking Approach



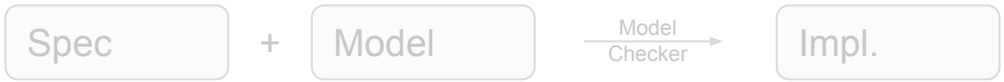
Reactive Synthesis Approach



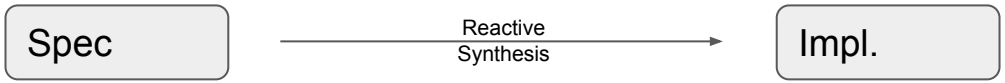


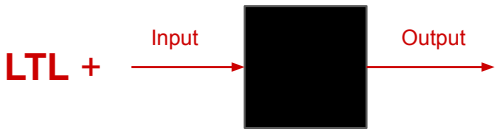
Synthesising controllers

Model Checking Approach



Reactive Synthesis Approach





Lookup table
BDD
AIGER circuit
...

Pitfalls of current implementations

Large

Uninterpretable

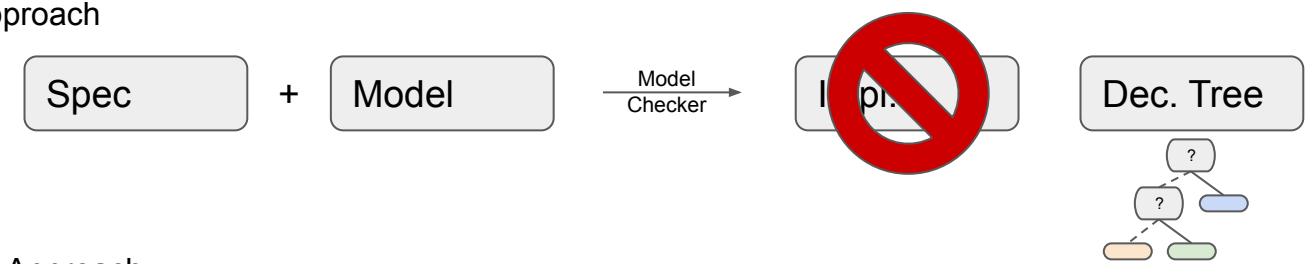
Difficult to implement

Difficult to debug

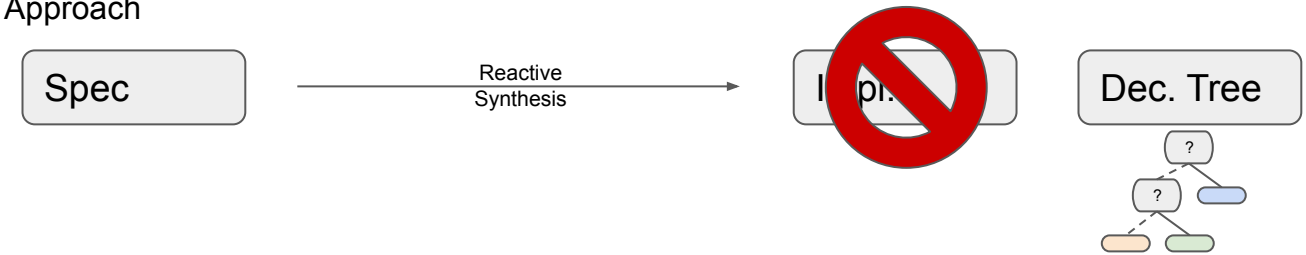
Mistrust in methods

Synthesising controllers

Model Checking Approach



Reactive Synthesis Approach



Decision trees

(from Machine Learning)

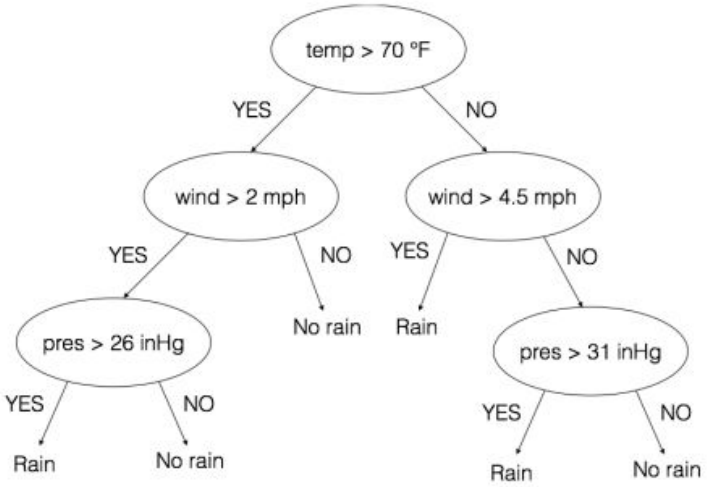
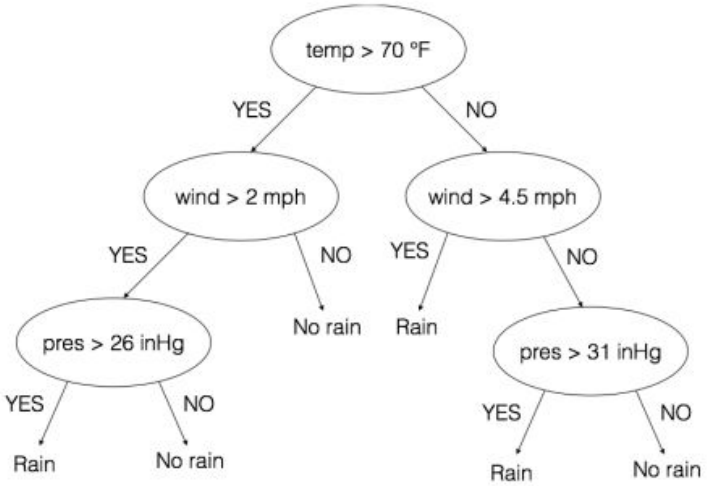


Image courtesy: https://prateekvjoshi.com/2016/03/22/how-are-decision-trees-constructed-in-machine-learning/

Decision trees

(from Machine Learning)




```
temp = 100 F
wind = 5 mph
pres = 32 inHg
```

Image courtesy: https://prateekvjoshi.com/2016/03/22/how-are-decision-trees-constructed-in-machine-learning/

Decision trees in formal methods

Garg et. al. POPL'16

- Program invariants from implication counterexamples

Neider et. al. (ACM TOCL, May 2018)

- Piece-Wise Functions against logical specifications

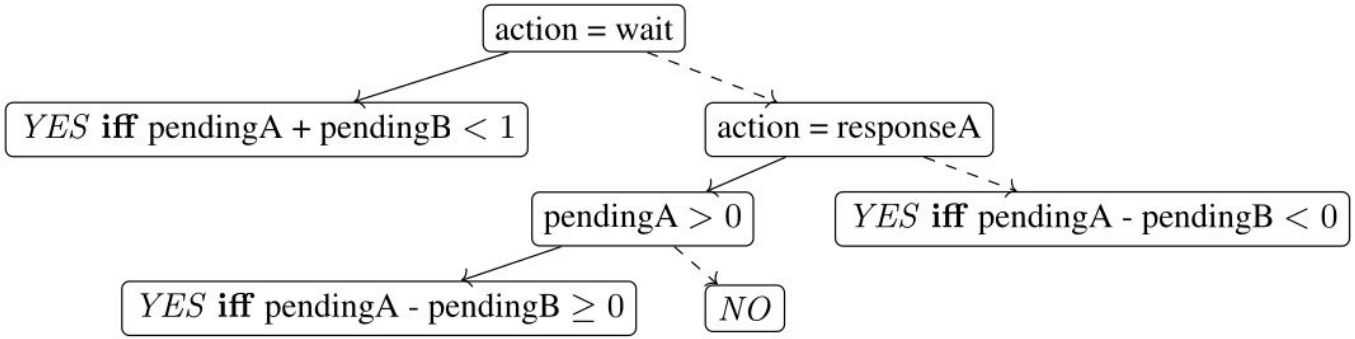
Brazdil et. al. (CAV 2015)

- Counter-example representation using dec. trees

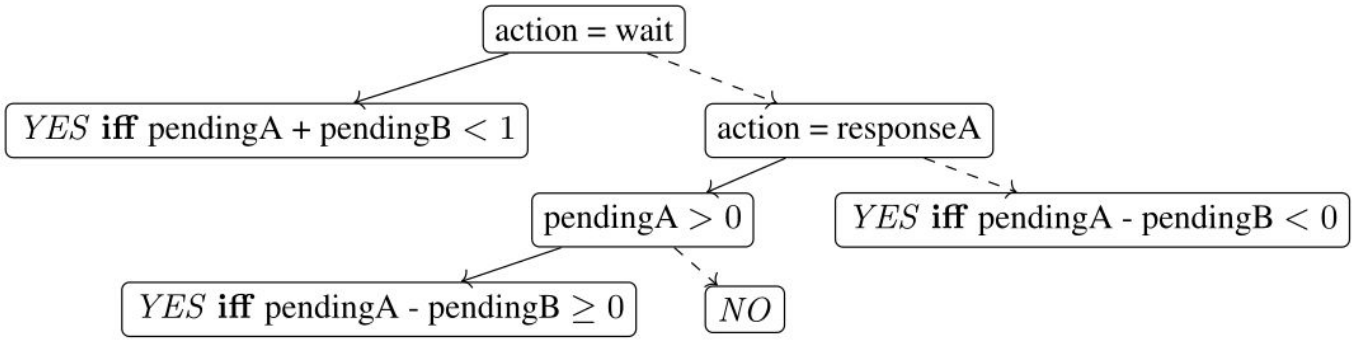
Brazdil et. al. (TACAS 2018)

- Dec. trees vs BDDs in reactive synthesis

Contribution: DT with linear classifiers



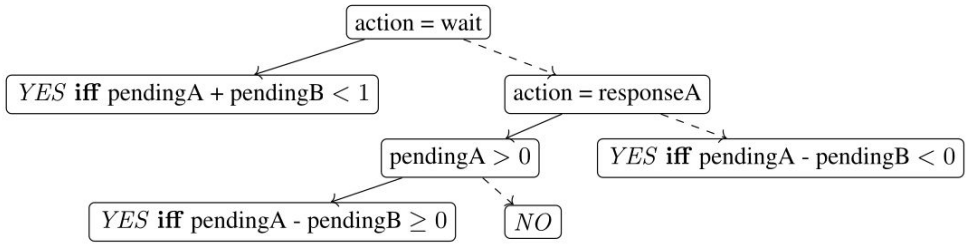
Contribution: DT with linear classifiers



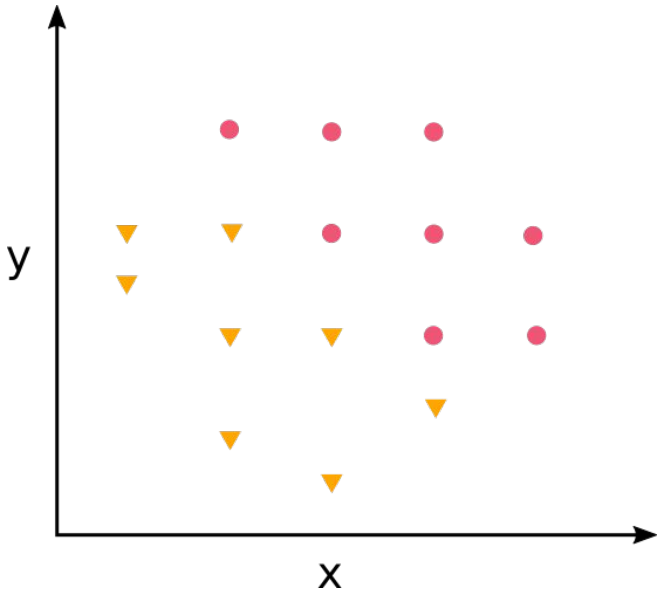
pendingA = 10
pendingB = 5
action = responseA?

No such algorithm from ML?

- Want exact data representation
- No algorithm with LC + exact data reprst.
- Focus in ML on generalizing to data
- Not ideal for strategies

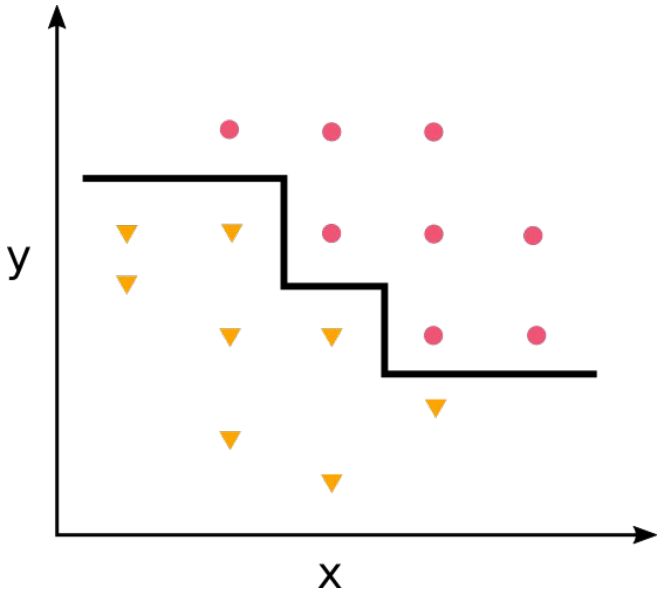


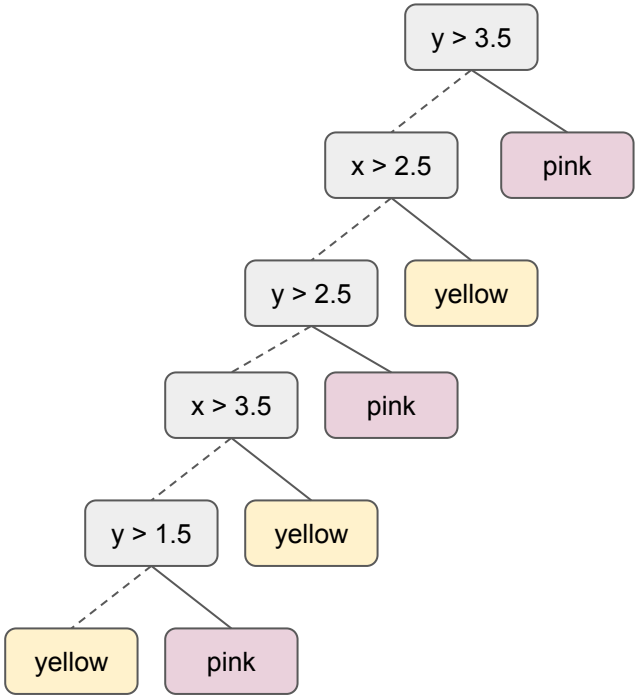
Why linear classifiers?



- Each point - state
- Two actions: circle/triangle

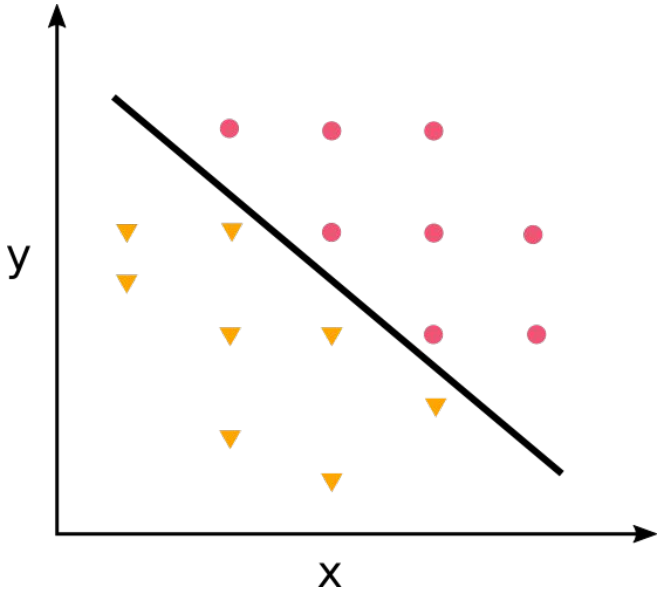
Why linear classifiers?



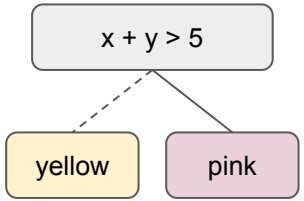


Standard DT: Only axis-parallel predicates

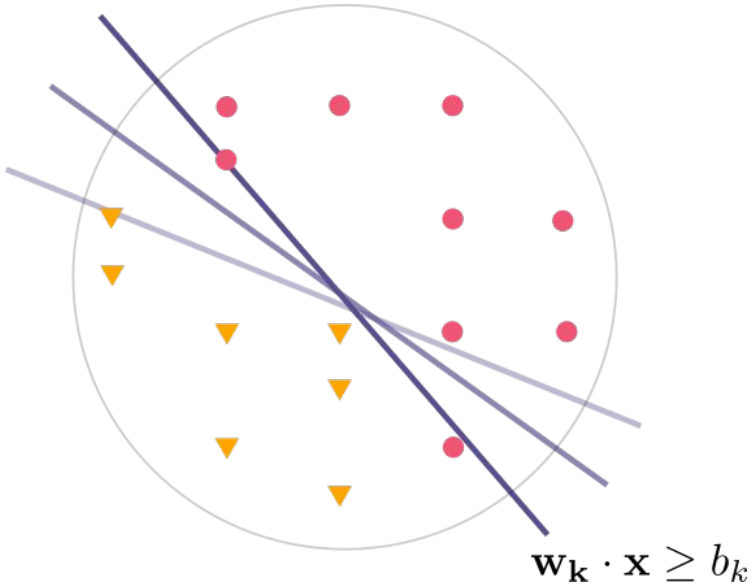
Why linear classifiers?



Any linear predicate

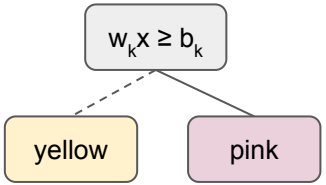


How does our algorithm work?

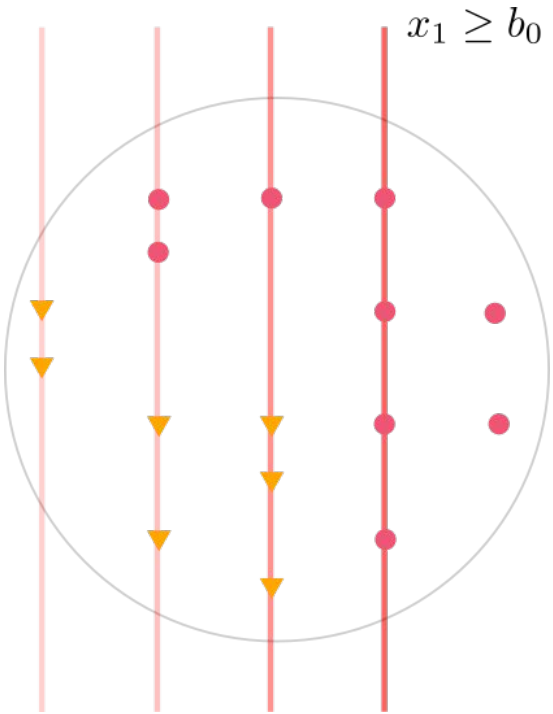


Step 1: Check if dataset is linearly separable

(If yes, we are done)



How does our algorithm work?



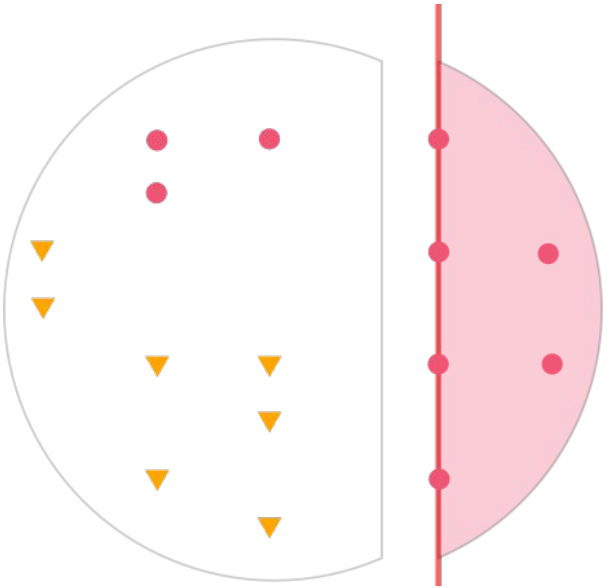
Step 2: Try out each axis-parallel split…

$x_1 \geq 1$ $x_1 \geq 2$ $x_1 \geq 3$ ….. $x_1 \geq b_0$

……

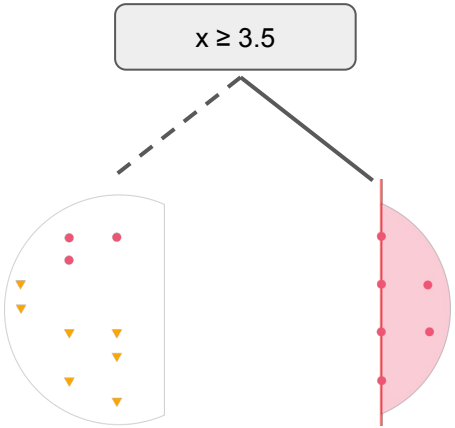
$x_2 \geq 1$ $x_2 \geq 2$ $x_2 \geq 3$ …..

How does our algorithm work?

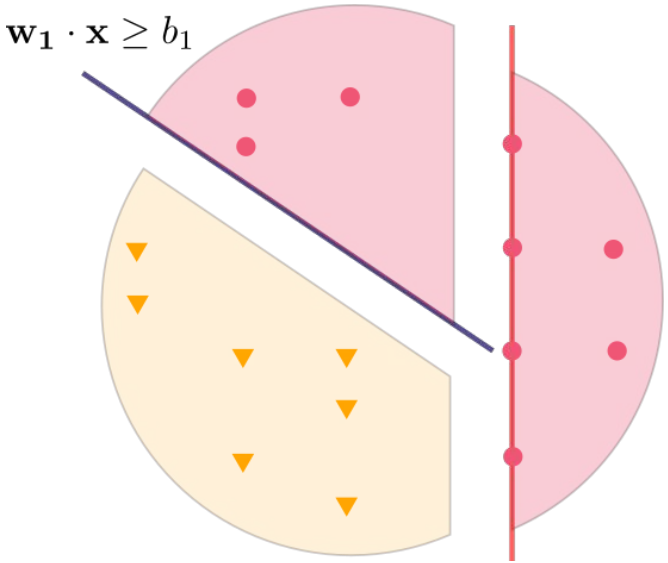


Step 2: Try out each axis-parallel split…

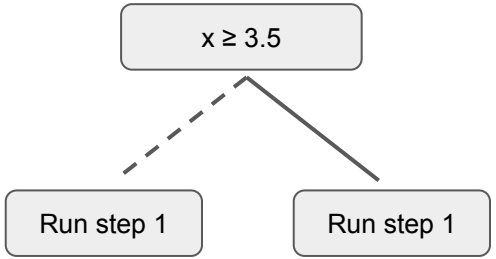
until the best split is found!



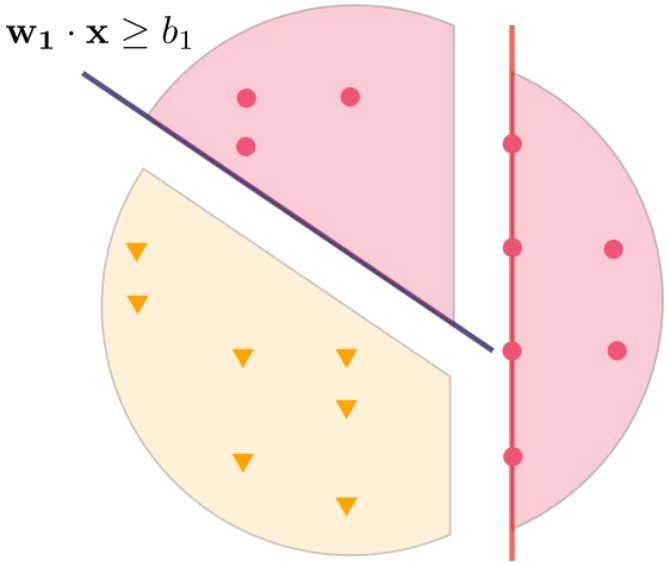
How does our algorithm work?



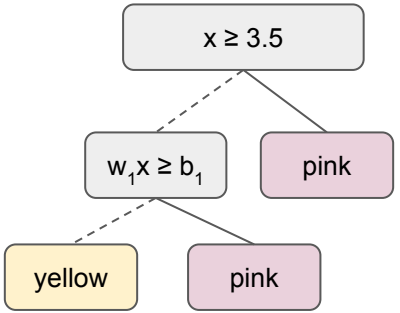
Step 3: for each child, continue from step 1.



How does our algorithm work?

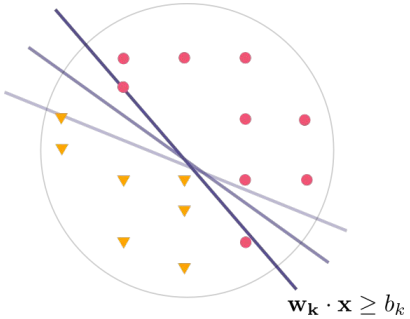


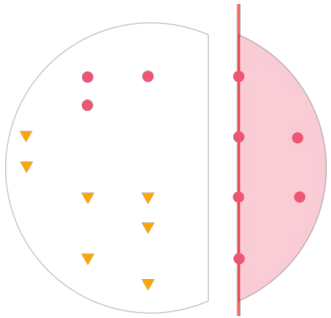
Step 3: for each child, continue from step 1.

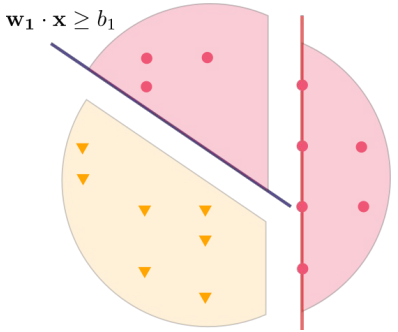


Summary of algorithm

1. Check if linearly separable (if yes, we are done)
2. Find find best axis-parallel split
3. For each children, go to step 1



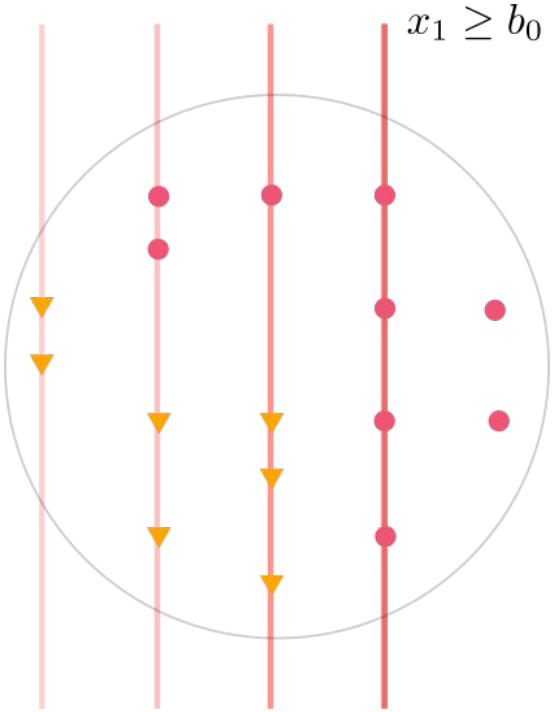




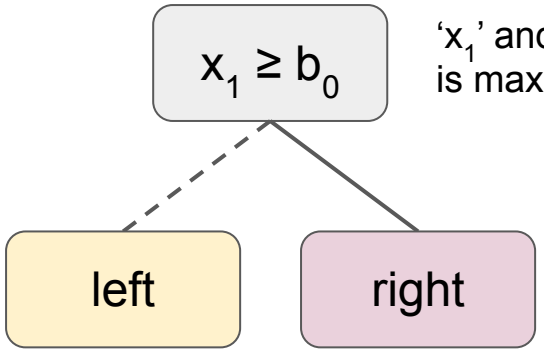
Finding best split

Common measures used in ML

- Information gain
- Gini index



Finding best split: Information Gain



'x_1' and 'b_0' such that information gain is maximized

Information gain = entropy(**parent**) - (entropy(**left**) + entropy(**right**))

Finding best split: something better?

Area under receiver operator characteristics (AUROC)

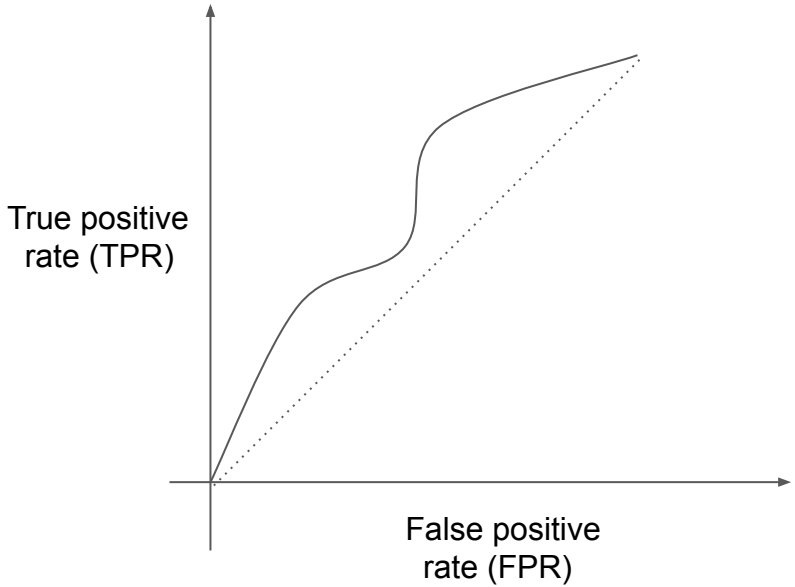
- Devised during WWII
- Now used in machine learning

Finding best split: AUROC

Area under ROC

Given: dataset and family of classifiers w**x** ≥ b

```
for each b:
     TPR = % actual +ve identified as +ve
     FPR = % actual -ve identified as -ve
     plot (TPR, FPR)
```



Finding best split: AUROC

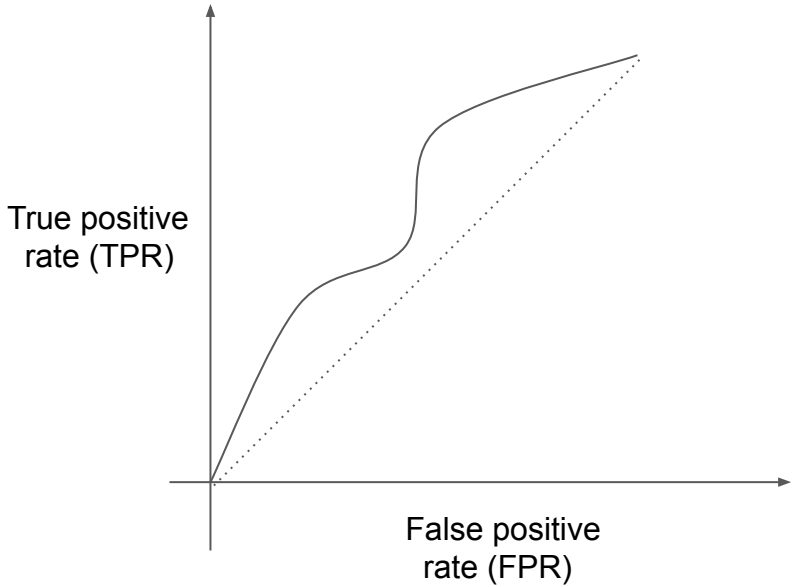
Area under ROC

Given: dataset and family of classifiers w**x** ≥ b

Intuition

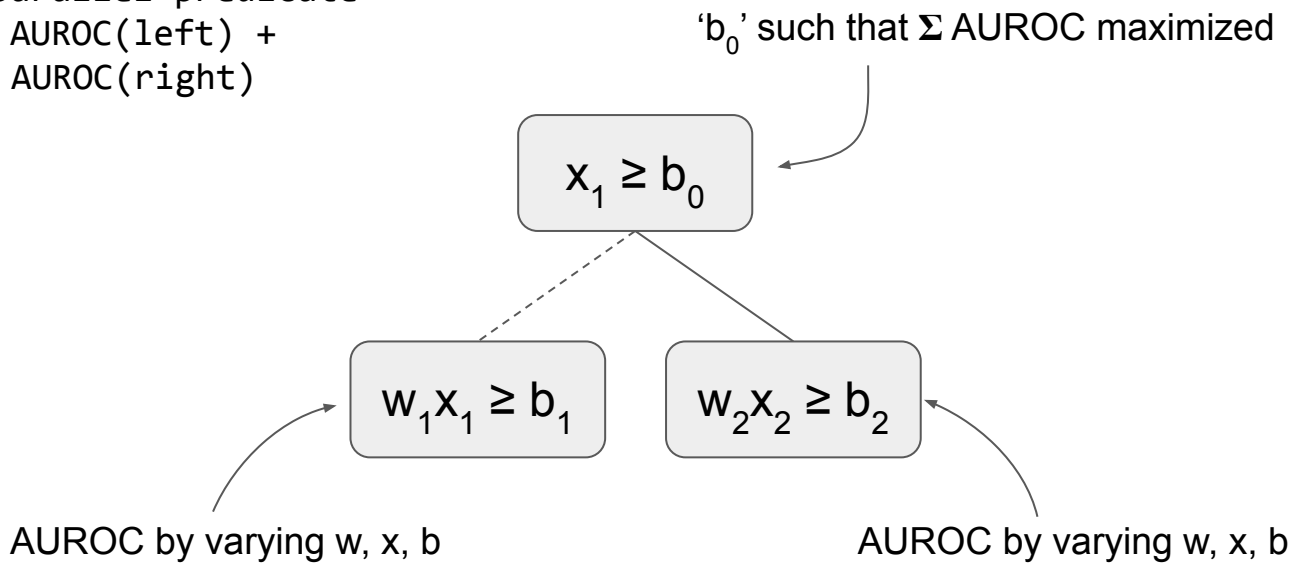
Point above 45 deg line - good
Point below 45 deg line - bad

Area under ROC curve = 1 (perfect)
Area under ROC curve = 0.5 (average)
Area under ROC curve = 0 (predicts exact opposite!)



Finding best split: AUROC

```
for each axis-parallel predicate
    goodness = AUROC(left) +
               AUROC(right)
```



Experiments

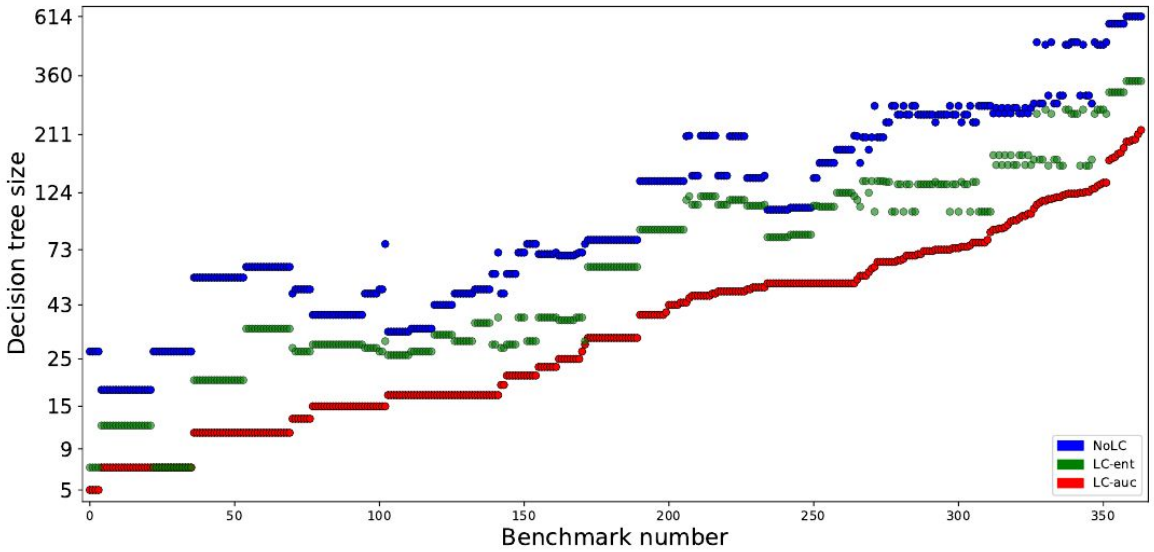
Reactive synthesis

1. Scheduling of washing cycles (SYNTCOMP 2015)
2. LTL synthesis
3. LTL synthesis with randomized environment

Model checking

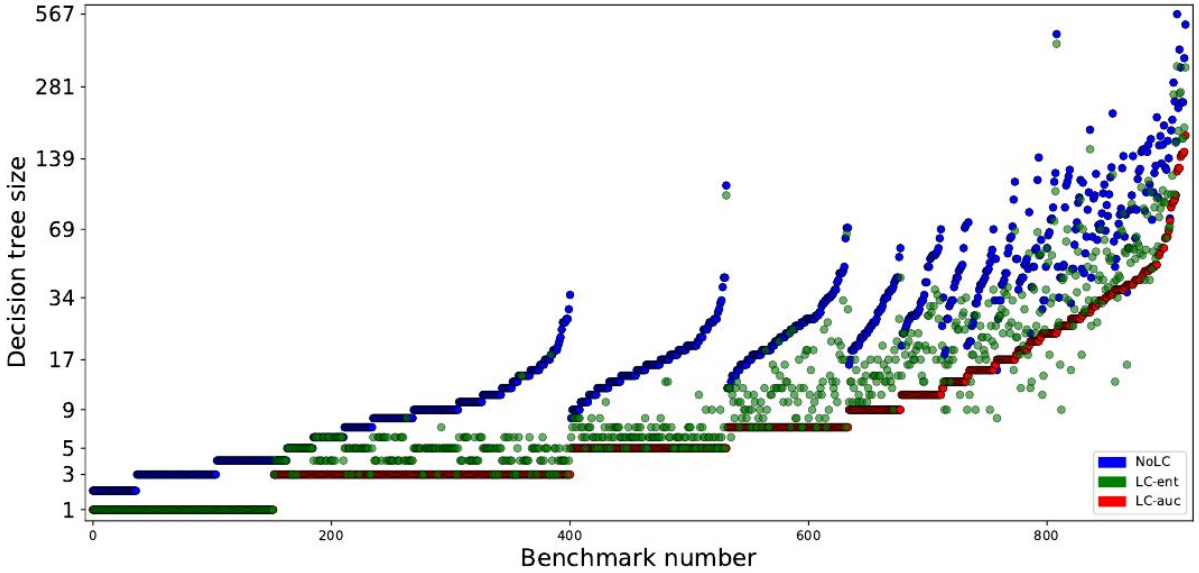
4. PCTL on MDPs

Experiments: Washing Cycle (SYNTCOMP)



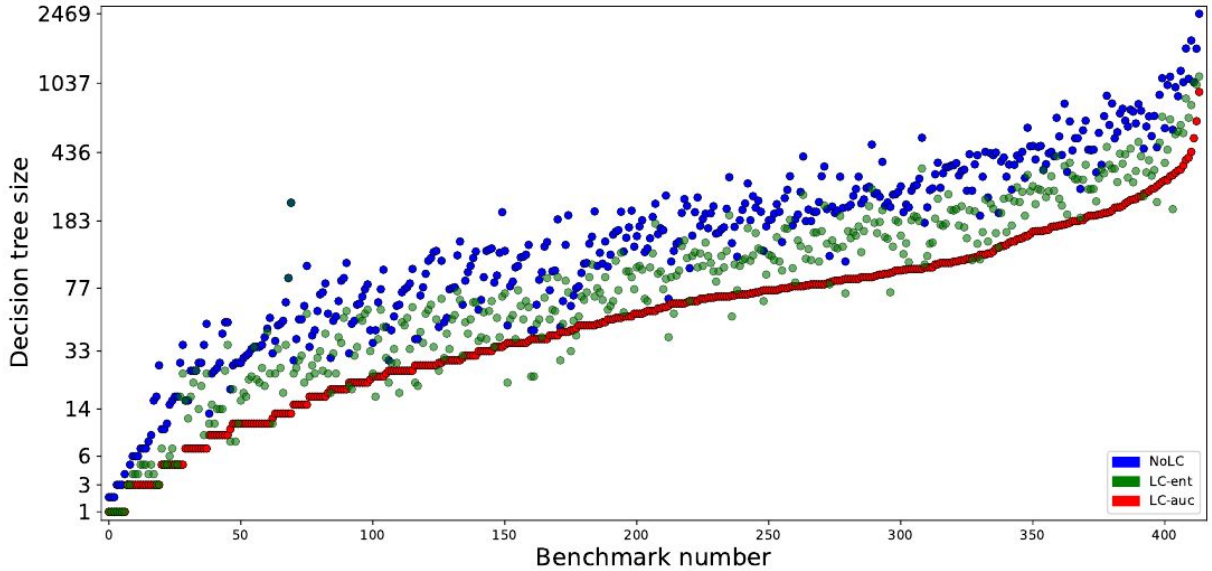
Mean
LC-entropy is 59% of NoLC
LC-auc is 33% of NoLC

Experiments: LTL synthesis



Mean
LC-ent is 51% of NoLC
LC-auc is 36% of NoLC

Experiments: LTL synthesis with randomized env.



Mean
LC-ent is 58% of NoLC
LC-auc is 38% of NoLC

Experiments: MDP

Model	# Train	# Dim	NoLC	LC+Entropy	LC+AUROC
coin-5	451204	13	2572	1626	**566**
csma-2-4	79580	13	54	**41**	59
firewire-5	90389	12	102	85	**72**
leader-4	38016	22	152	92	**45**
mer-30	1408932	19	1373	1332	**126**
wlan-2	275140	14	288	**206**	353
zeroconf	268326	24	413	**330**	376

Summary

- Representing strategies using DT with linear classifiers
- New splitting measure (AUROC)

Future work

- Representing AIGER circuits as DT
- Linear predicates in all nodes?
- More complex predicates?