A Pattern Matching Approach to Map Cognitive Domain Ontologies to the IBM TrueNorth Processor

CCAA 2017

Nayim Rahman[1], Tanvir Atahary[1], Tarek Taha[1], and Scott A. Douglass[2]

[1]Electrical and Computer Engineering Department, University of Dayton

[2]711 HPW/RHAC, Cognitive Models and Agents Branch, Air Force Research Laboratory

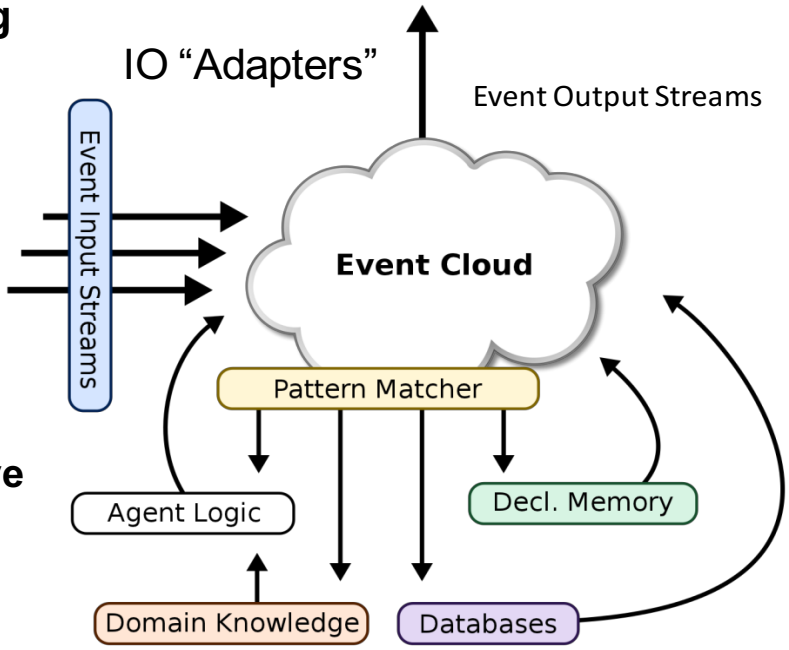
UNIVERSITY of DAYTON

Overview

- Computerized agent-based decision aids
- Represent knowledge in databases
- Mine this knowledge to find solutions to specific problems
- Use computer clusters to accelerate this mining

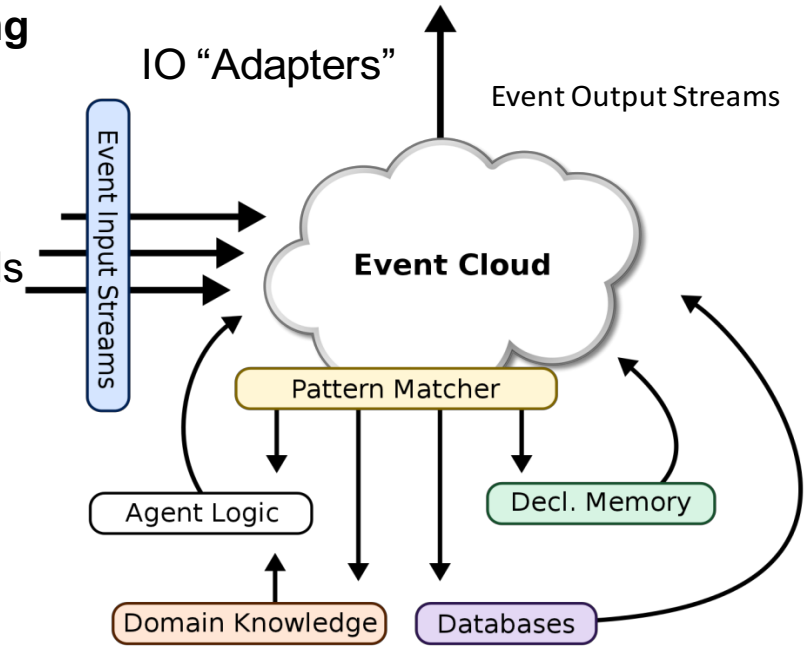
Cognitively Enhanced Complex Event Processing

- **Cognitively Enhanced Complex Event Processing (CECEP) Architecture:**
  - Models and agents are specified in research modeling language (RML)
  - RML has been developed using the Generic Modeling Environment (GME)
- **Consists of the following central net-centric components:**
  - soaDM: an associative memory application that allows RML models and agents to store and retrieve **declarative knowledge.**
  - soaCDO: a knowledge representation and mining application that allows RML models and agents to store and exploit **domain knowledge**.
  - Esper: a complex event processing framework that allows RML models and agents to base actions on context assessment and **procedural knowledge**. Procedural knowledge is represented in RML behavior models and processed using pattern matching and event abstraction capabilities provided by Esper.



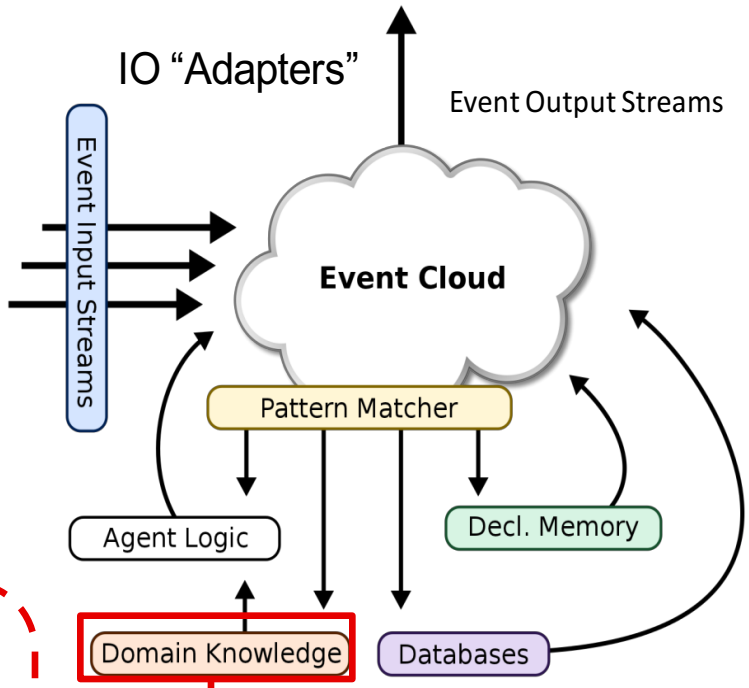
Cognitively Enhanced Complex Event Processing

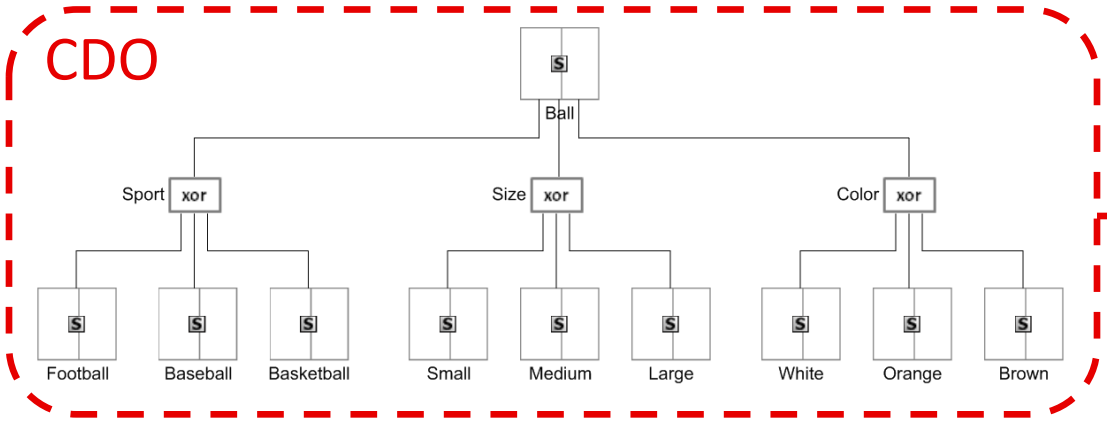
- **Cognitively Enhanced Complex Event Processing (CECEP) Architecture:**
  - Declarative Knowledge: specified as events and relations
  - Procedural Knowledge: specified as behavior models
  - Domain Knowledge: specified in CDOs and processed by constraint-satisfaction framework
  - Adapters : includes a number of IO "Adapters" or event input and output streams, allow models and agents specified in RML to be integrated into software-based instructional systems



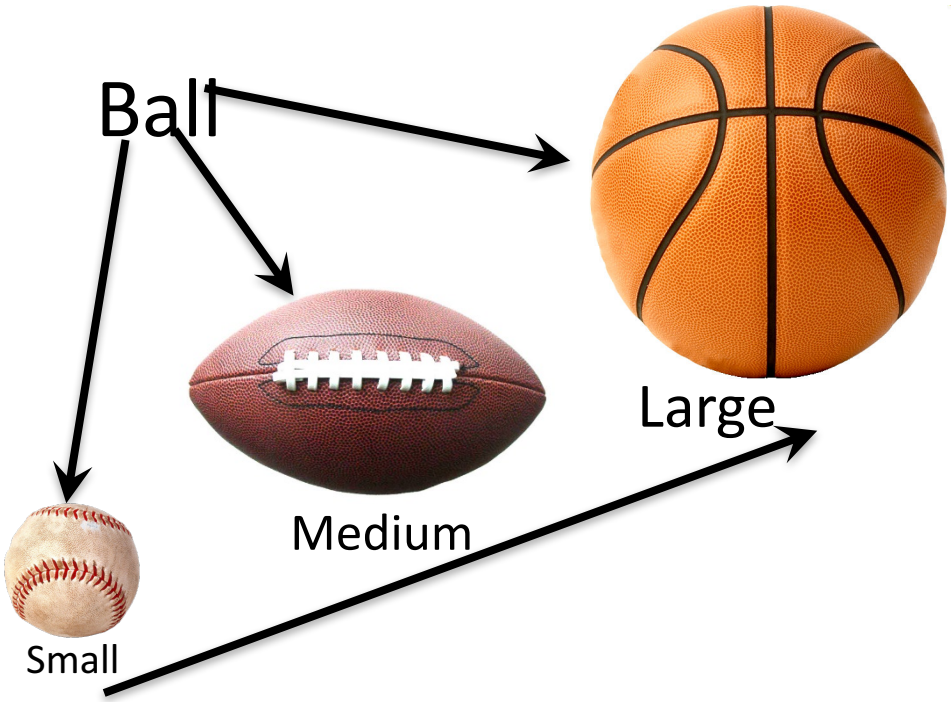
Cognitive Domain Ontology

Domain knowledge represented as a data structure called "Cognitive Domain Ontology" (CDO).





Example CDO



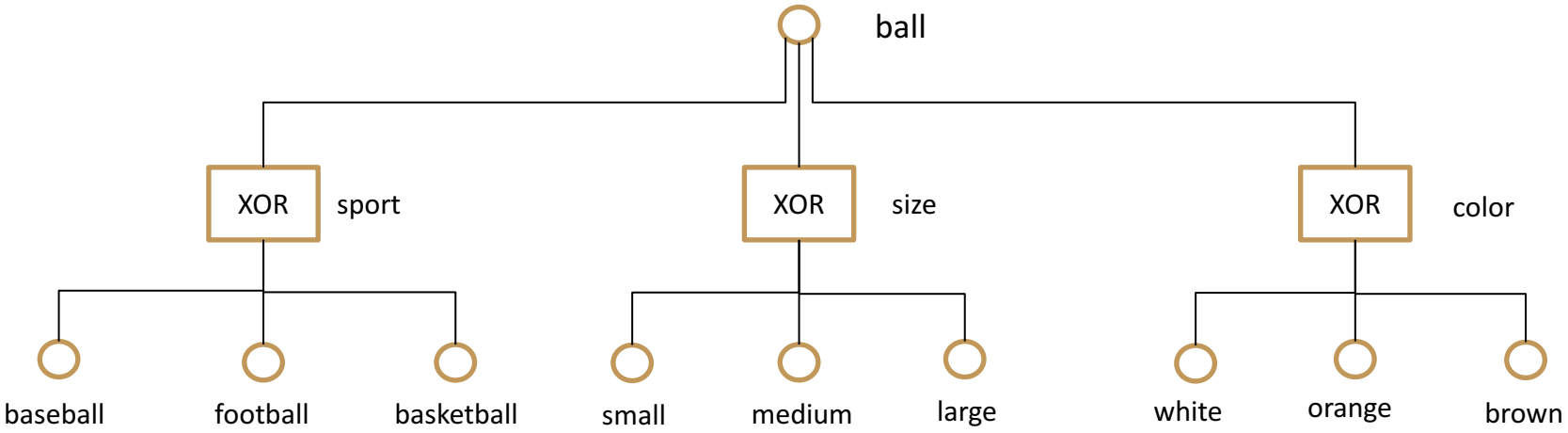
Sports

Size Constraints:
- Small
- *Not Medium*
- *Not Large*

Size Constraints:
- Medium
- *Not Small*
- *Not Large*

Size Constraints:
- Large
- *Not Small*
- *Not Medium*

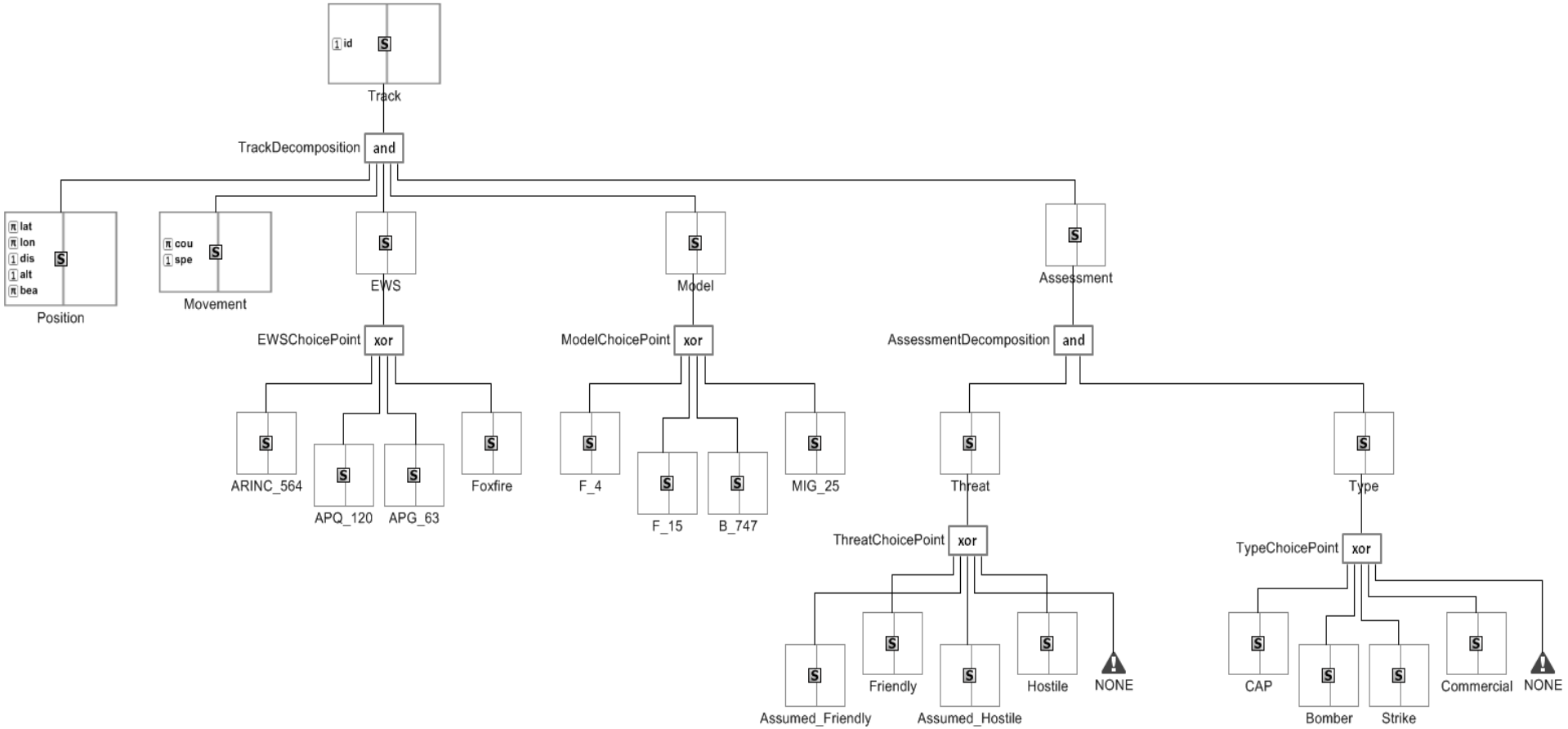
CDOs Model Domains and Have Constraints



Constraints:

If (sport ==baseball) then (size =small) and (color =white)
If (sport ==football) then (size =medium) and (color=brown)
If (sport ==basketball) then (size =large) and (color=orange)

Track CDO


id
S
Track
TrackDecomposition
and
lat
lon
dis
alt
bea
Position
cou
spe
Movement
EWS
Model
Assessment
EWSChoicePoint
xor
ModelChoicePoint
xor
AssessmentDecomposition
and
ARINC_564
APQ_120
APG_63
Foxfire
F_4
F_15
B_747
MIG_25
Threat
Type
ThreatChoicePoint
xor
TypeChoicePoint
xor
Assumed_Friendly
Friendly
Assumed_Hostile
Hostile
NONE
CAP
Bomber
Strike
Commercial
NONE

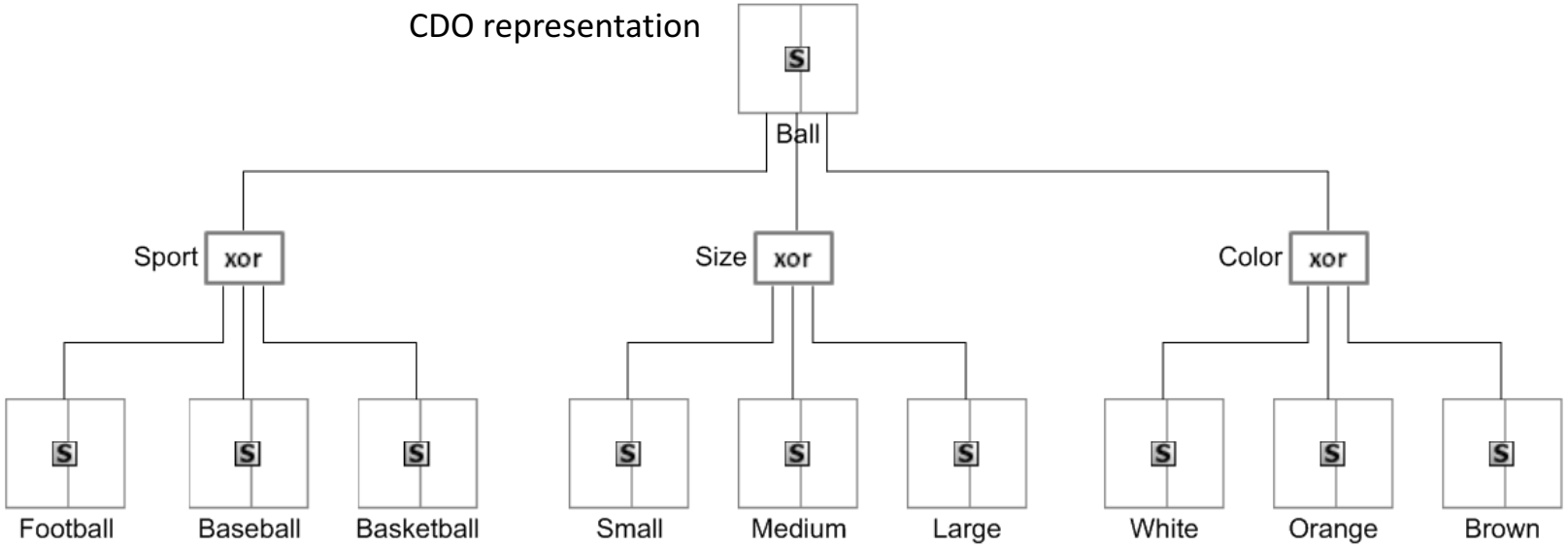
UNIVERSITY of DAYTON

Track CDO Constraints

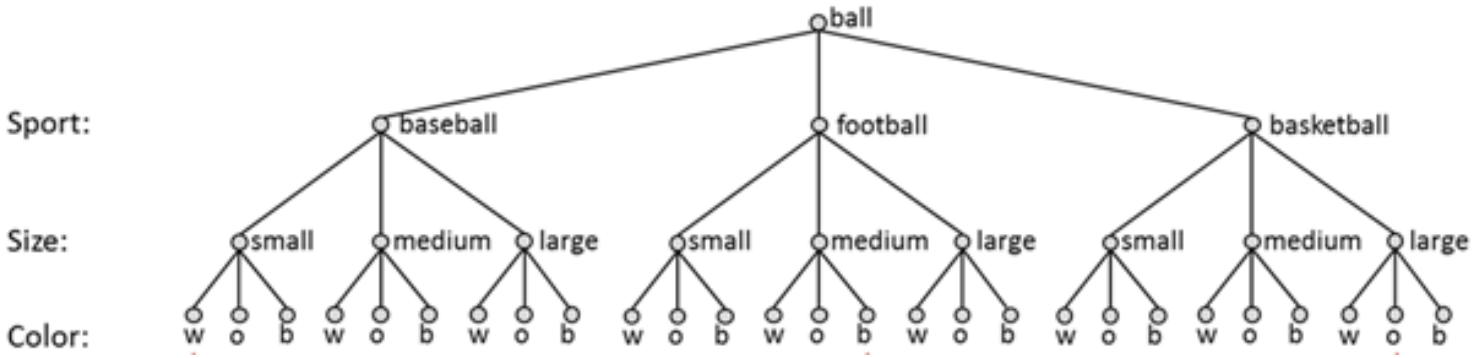
Name	Specification
C1	iff ews_choices is arinc_564 then model_choices is b_747
C2	iff ews_choices is apq_120 then model_choices is f_4
C3	iff ews_choices is apg_63 then model_choices is f_15
C4	iff ews_choices is foxfire then model_choices is mig_25
C5	if model_choices is f_4 then threat_choices is assumed_hostile hostile assumed_friendly or friendly type_choices is strike
C6	if model_choices is f_15 then threat_choices is assumed_friendly or friendly threat_choices is not (assumed_hostile or hostile) type_choices is strike
C7	if model_choices is b_747 then threat_choices is friendly or assumed_friendly threat_choices is not (assumed_hostile or hostile) type_choices is commercial
C8	if model_choices is mig_25 then threat_choices is assumed_hostile or hostile threat_choices is not (assumed_friendly or friendly) type_choices is strike
C9	if speed is between 350 and 550, altitude is between 25000 and 36000 then model_choices is b_747 threat_choices is friendly type_choices is commercial

CDO as a CSP Tree

CDO representation



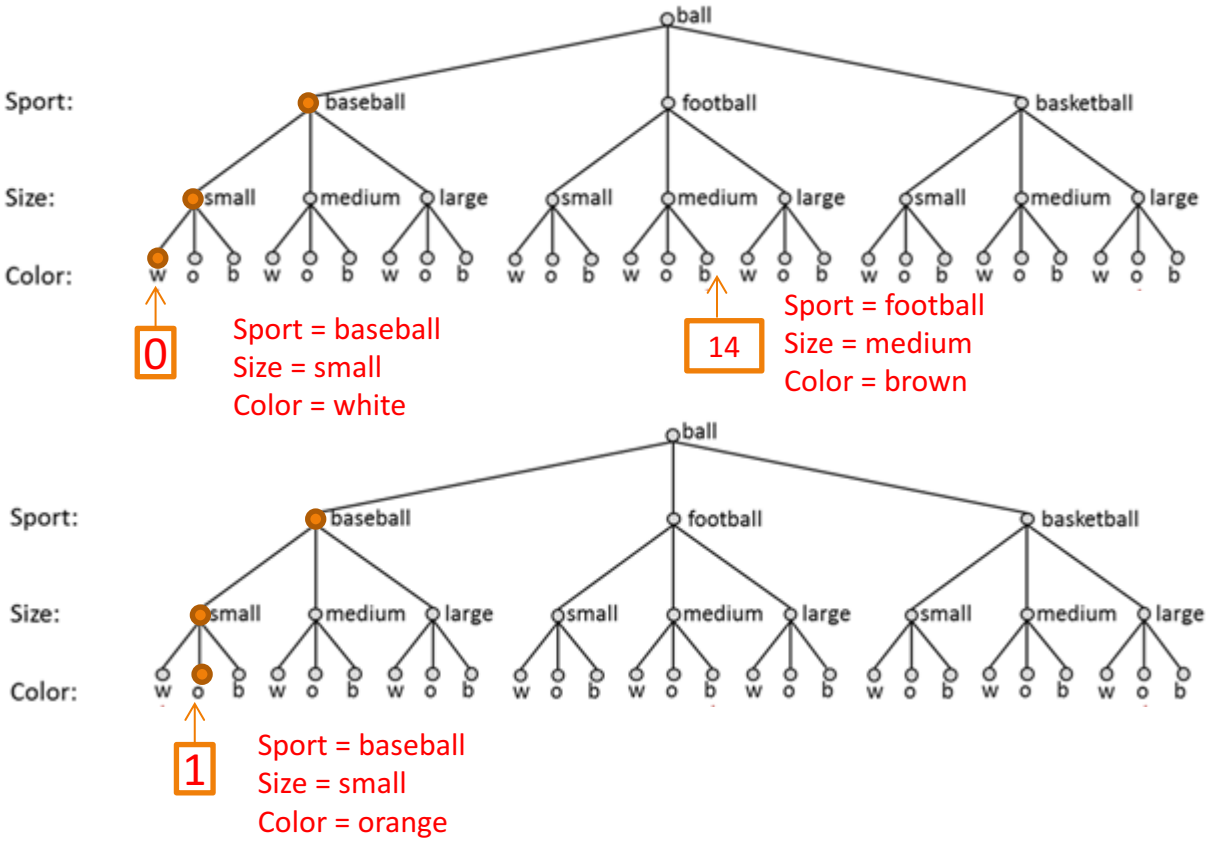
CSP tree representation



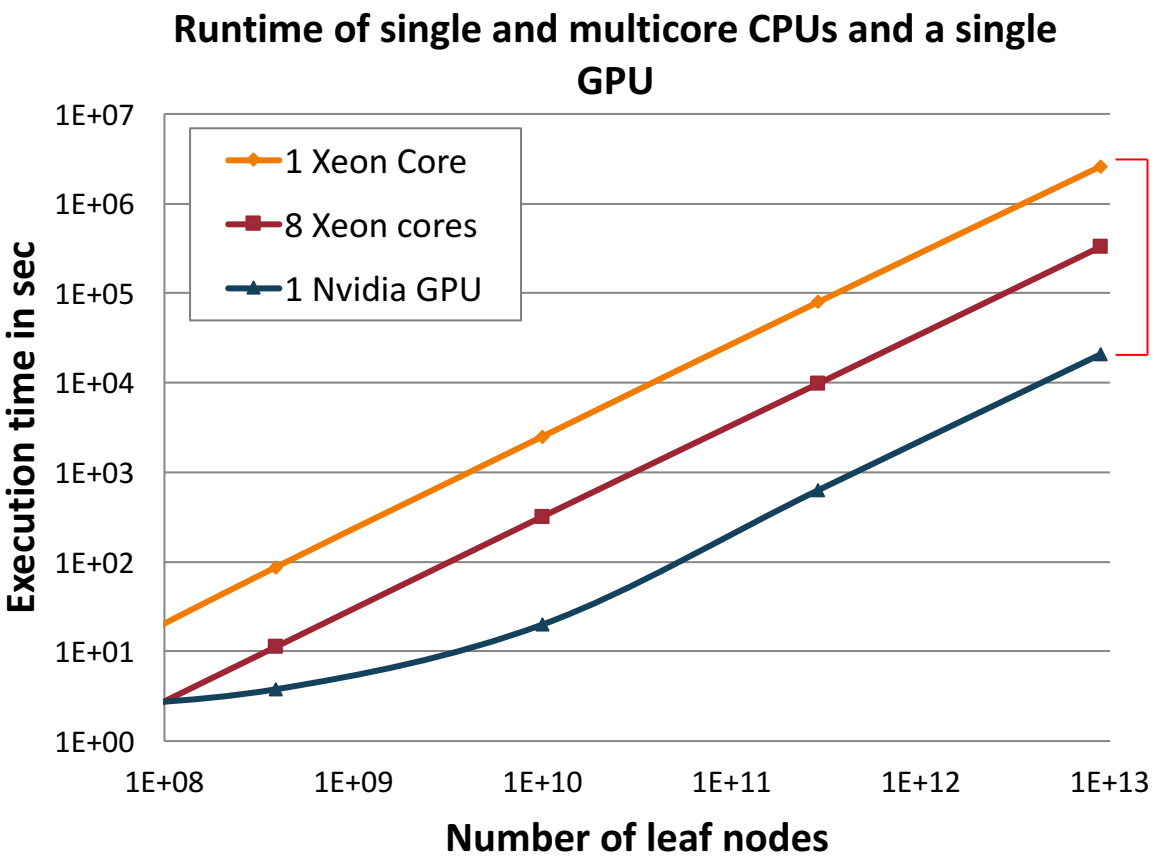
HPC Approach 1: Exhaustive Search

- Exhaustive depth first search
- Highly parallel

Constraints:
If (sport ==baseball) then (size =small) and (color =white)
If (sport ==football) then (size =medium) and (color=brown)
If (sport ==basketball) then (size =large) and (color=orange)

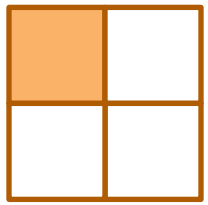


Results



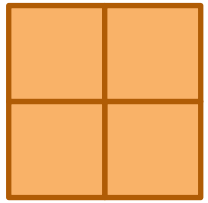
100x

Speedup relative to 1 Xeon core:



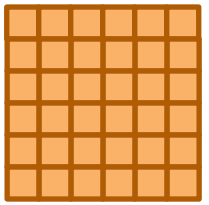
Intel Xeon (1 core)

4



Intel Xeon (4 cores)

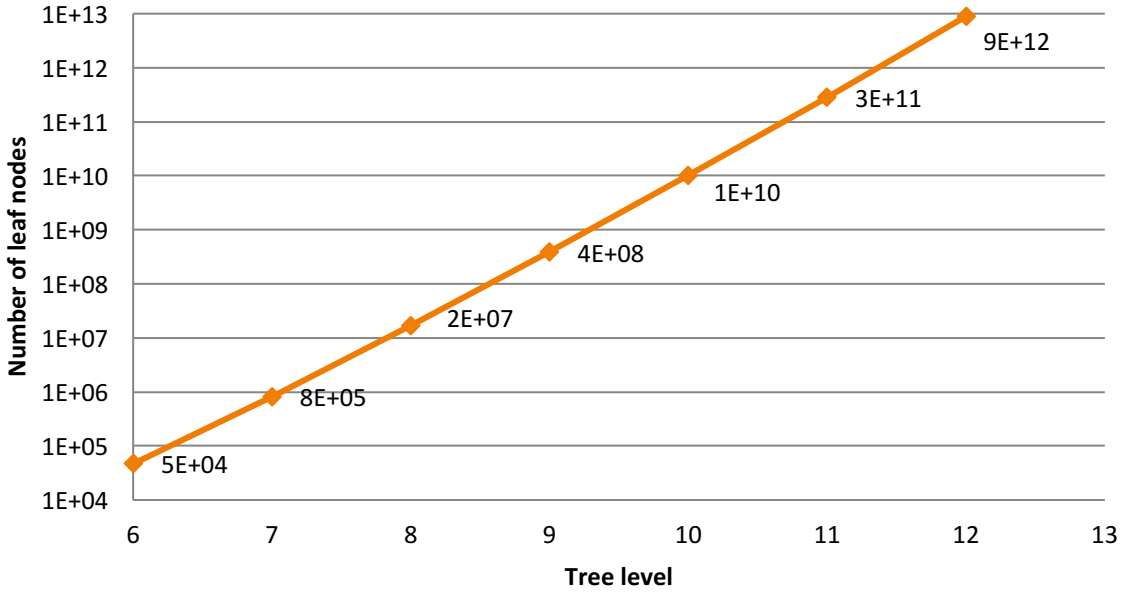
100



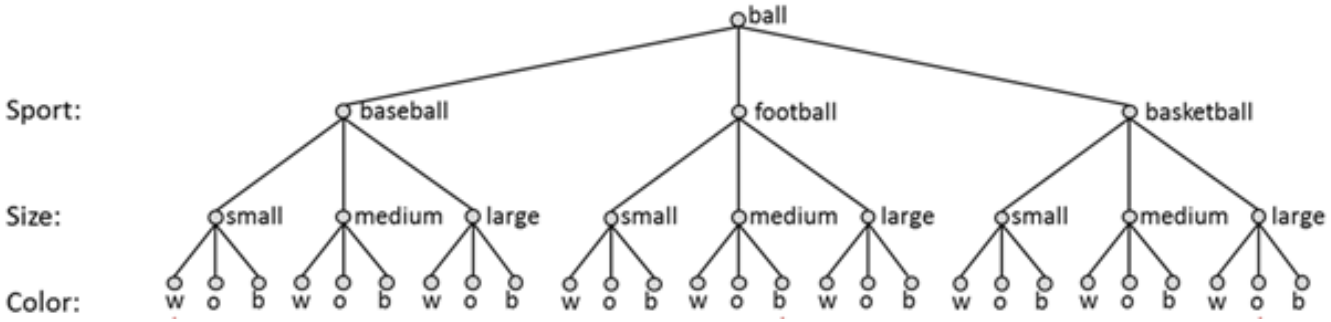
NVIDIA GPU (448 cores)

CDO Scaling

Data based on synthetic CDOs with *n* branches having *n* entities each.



Example tree from a CDO with 3 levels:



UNIVERSITY of DAYTON

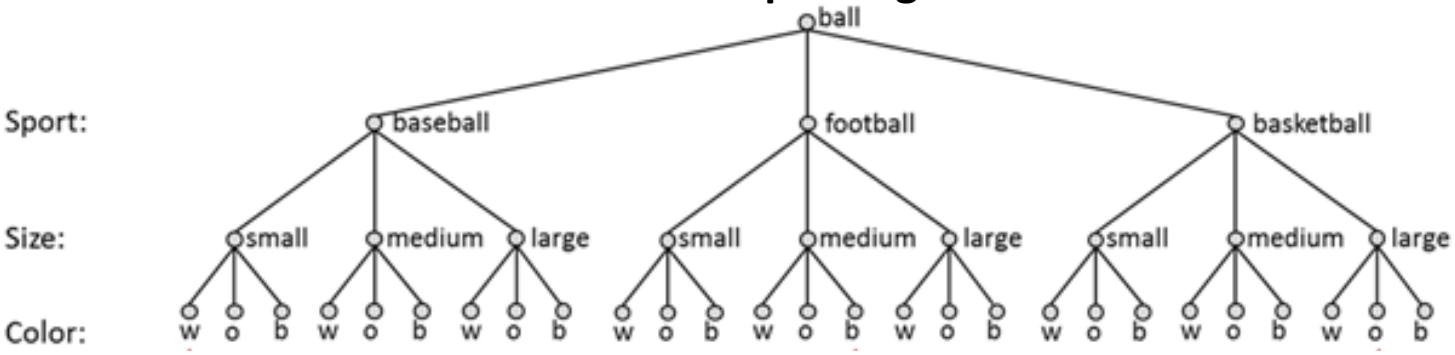
CDO Run Times: Exhaustive Search

Tree Level	CPU runtime	1 GPU run time
10	42 m	20 s
11	22 h	10 m 22s
12	30 days*	5.75 h
13	3 years *	9 days*
14	75 years*	1 years*
15	2250 years*	42 years*
16	675 centuries*	1898 years*

*estimated run times

HPC Approach 2: Informed Search

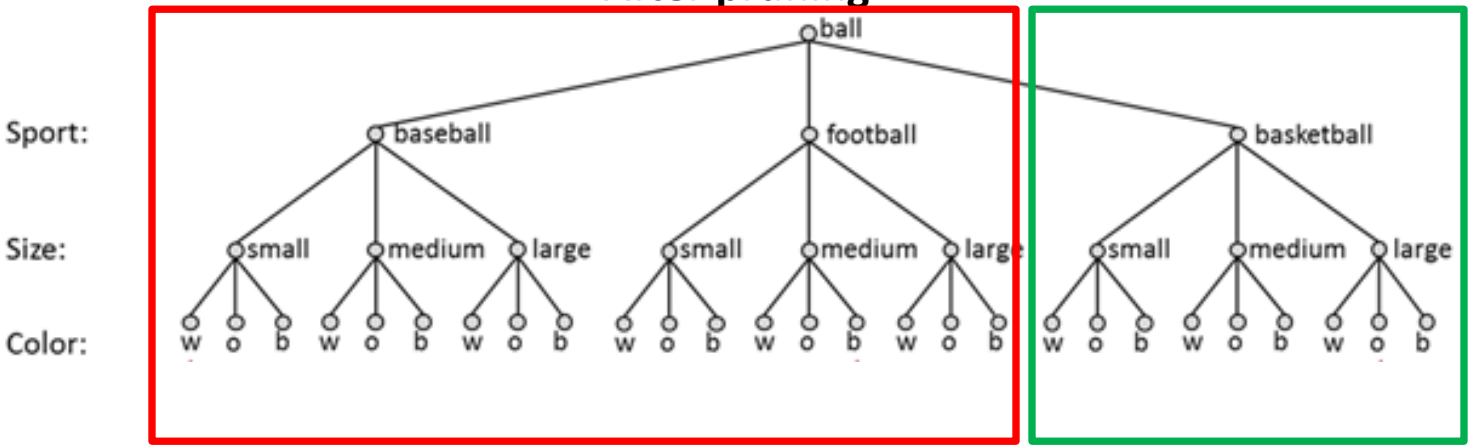
Before pruning



Constraints:

If (sport == basketball) then (size ⇐ large) and (color ⇐ orange)

After pruning

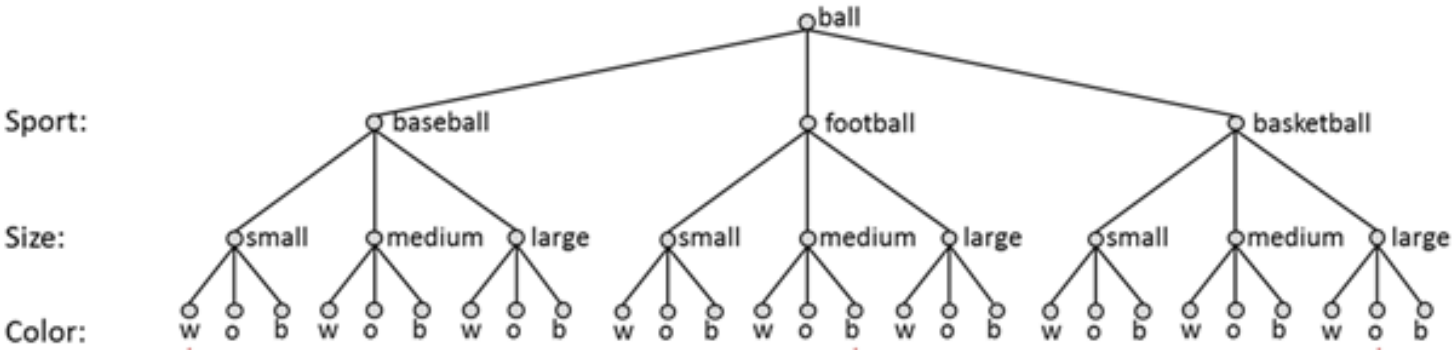


No solution for that particular constraint
--Pruned 18 out of 27 child node

Possible solution space
--Only 9 child node

HPC Approach 2: Informed Search

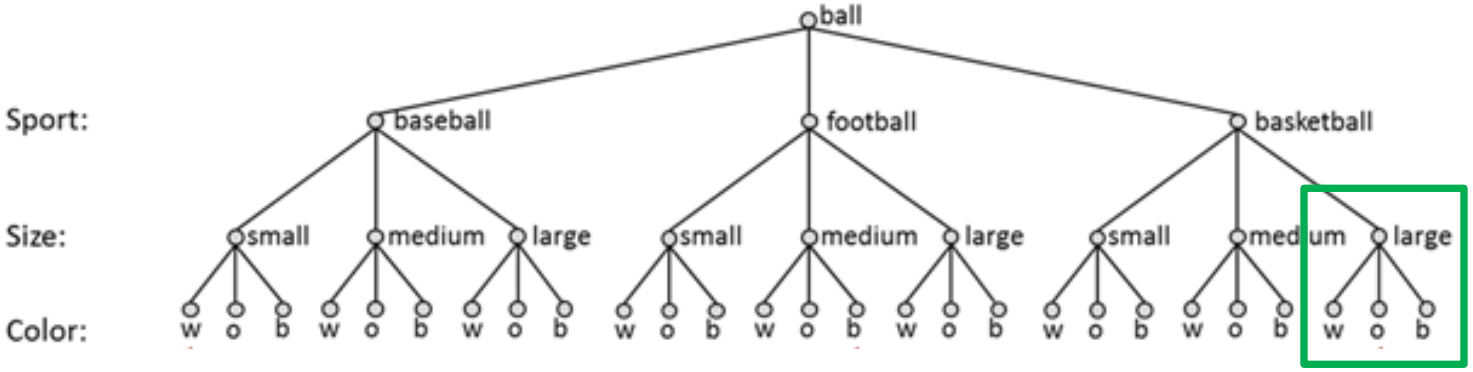
Before pruning



Constraints:

If (sport == basketball) then (size ⇐ large) and (color ⇐ orange)

After pruning



No solution for that particular constraint
--Pruned 24 out of 27 child node

Possible solution space
--Only 3 child node

UNIVERSITY of DAYTON

Path based Forward Checking Algorithm

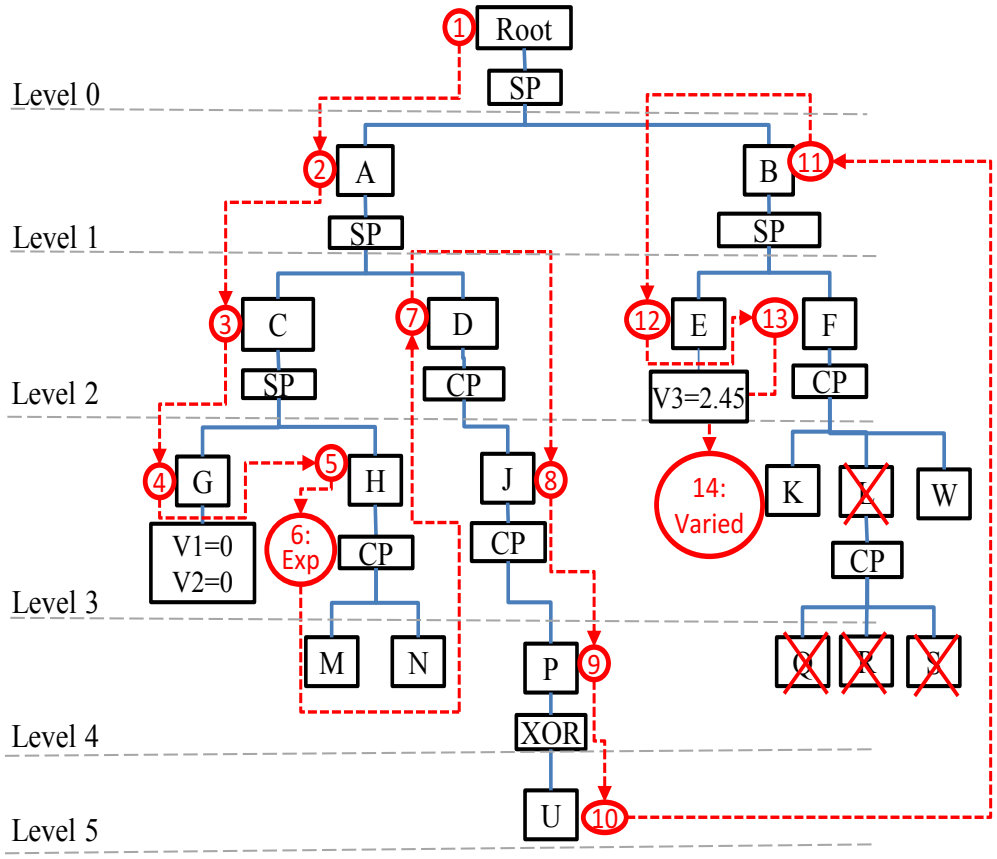
This algorithm has three steps :

- ***Pre-processing step :*** **simplifies constraints which in turn removes any interdependencies be-tween branches.**

- ***Path generation step*** **: prunes the search space based on constraints and make a path from root to leaf node.**

- ***Solution generation step*** **: lists all possible solutions in a compact manner based on the path generated in step 2.**

Path Generation



Path:

Root-A-C-G-H-Exp-D-J-P-U-B-E-V3=2.45-F-varied

Solution based on the path:

Root is SubPart event and it's corresponding choice is below
- A is SubPart event and it's corresponding choice is below
- C is SubPart event and it's corresponding choice is below
- G is 'Variable' event and values associated with it is below
  - V1 is 0 V2 is 0
- H is a ChoicePoint event and it is expanded
  - *H can be M or H can be N*
- D is a ChoicePoint event and it is J
- J is a ChoicePoint event and it is P
- P is a ChoicePoint event and it is U

Number of solution found from branch 1 = 2

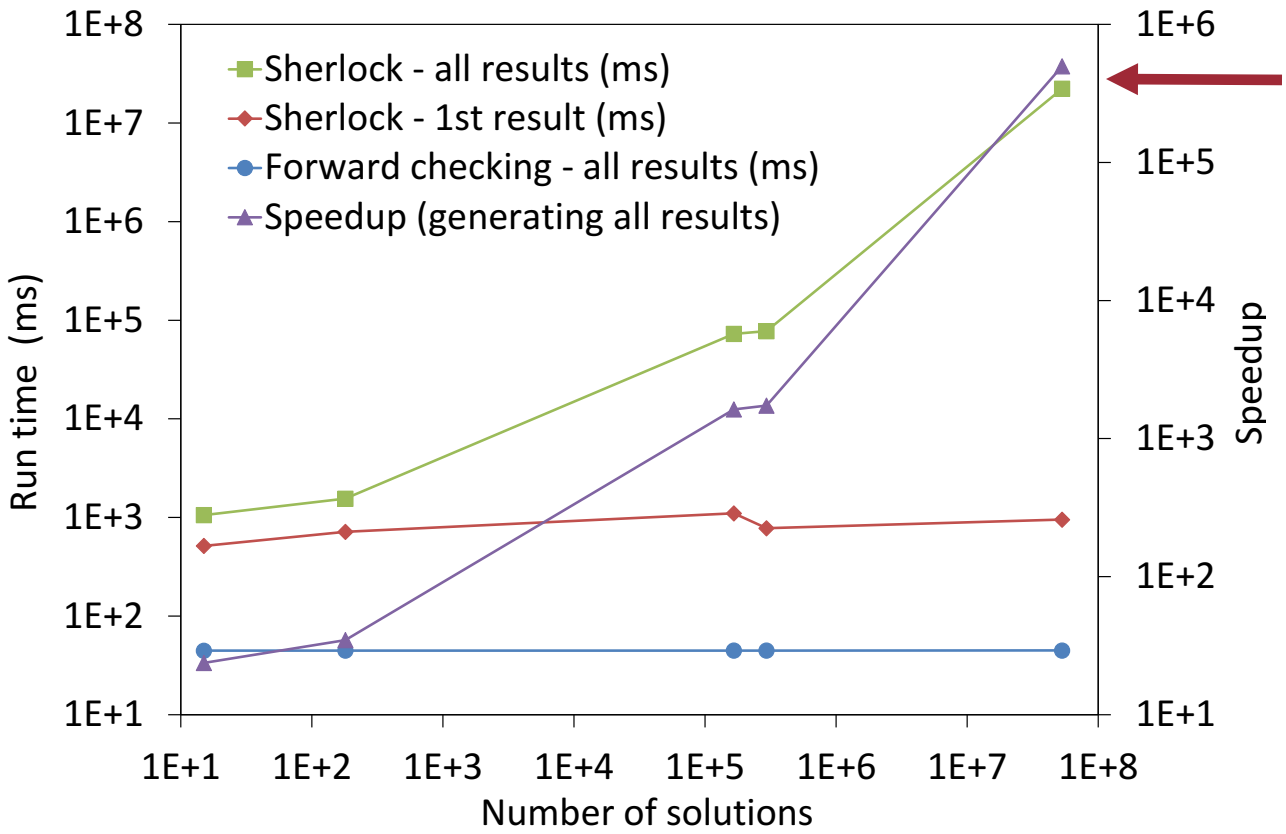
- B is SubPart event and it's corresponding choice is below
- E is 'Variable' event and values associated with it is below
  - V3 is 2.45
- F is a ChoicePoint event and it is varied
  - *F can be K or F can be W*

Number of solution found from branch 2 = 2

Total possible number of solution= 2×2=4

Performance (1 Xeon core)

Scar CDO

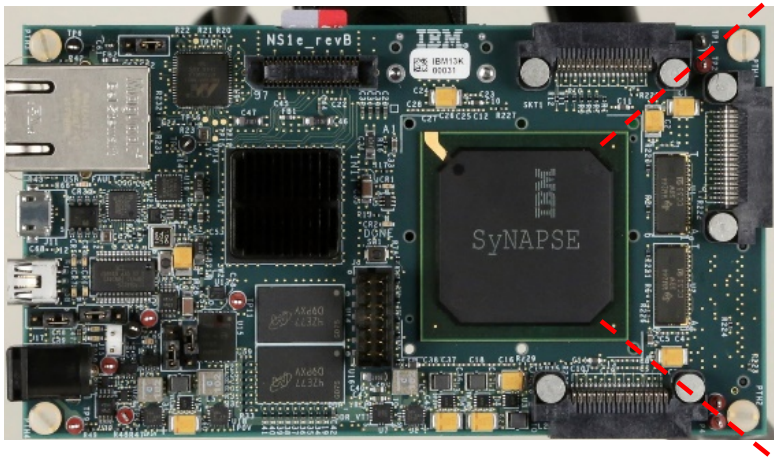


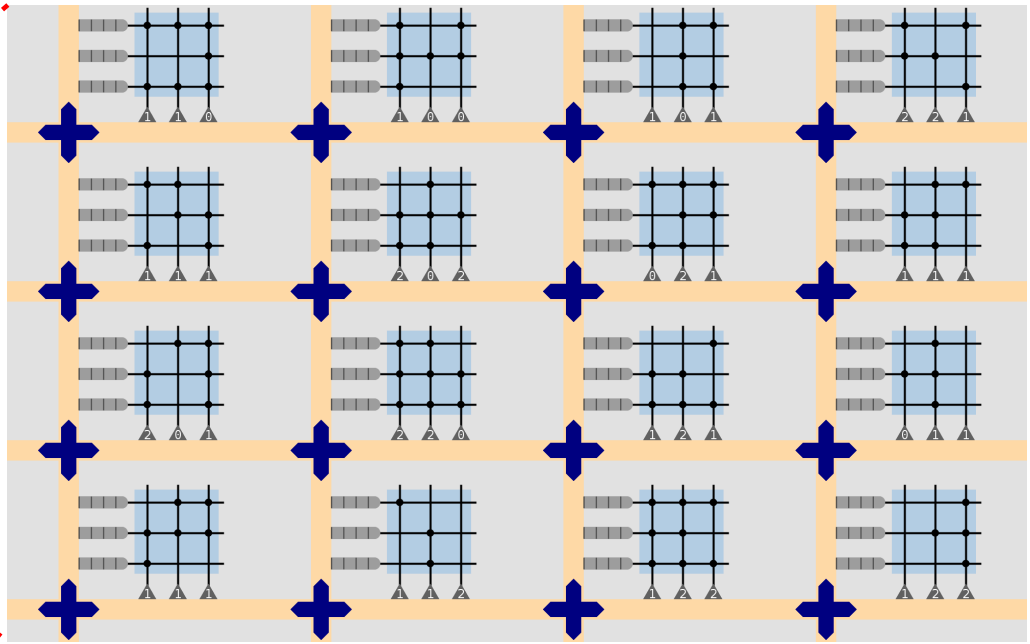
~1 Million times speedup.
Requires high power computing resources.

UNIVERSITY of DAYTON

IBM TrueNorth

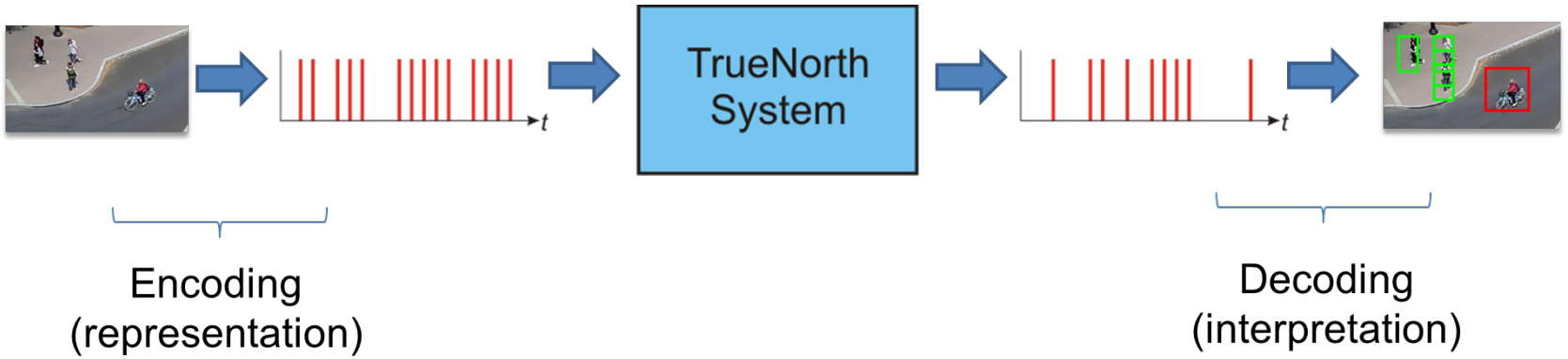
- 4096 cores
- 256 neurons / core
- 100mW



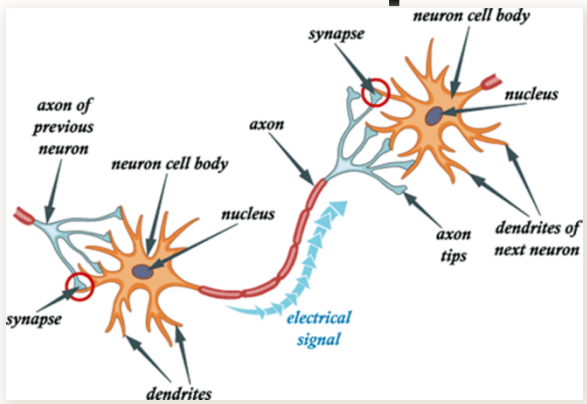


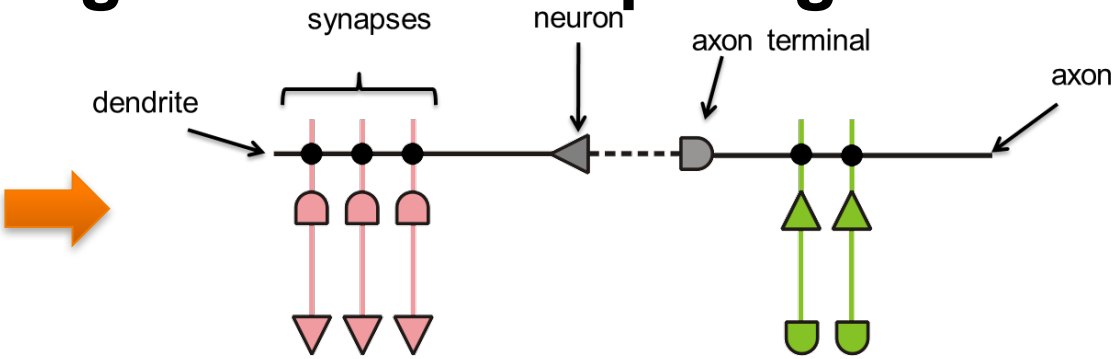
Spiking Neurons

- **TrueNorth System is spikes-in, spikes-out**

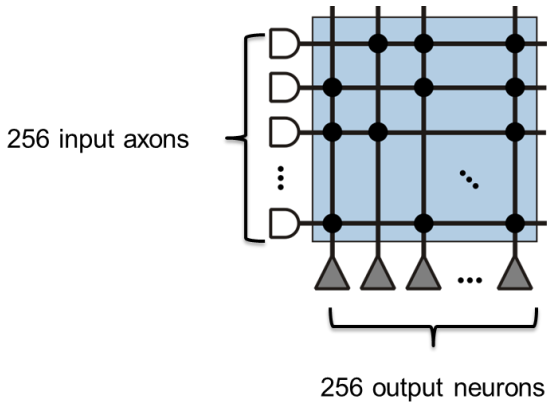


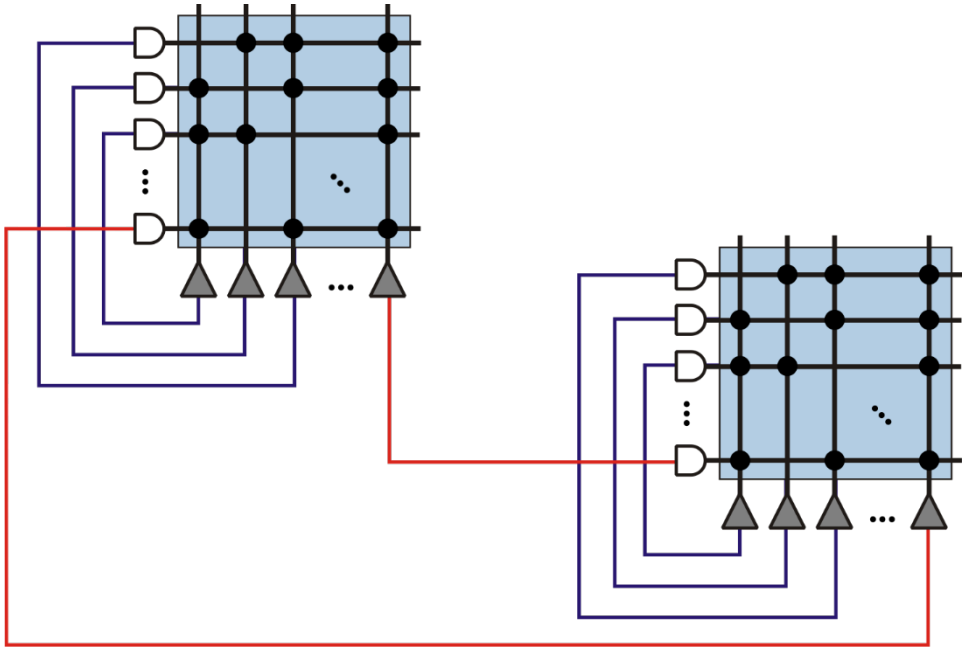
- **Internal processing is based on spiking**





Cores and Connectivity

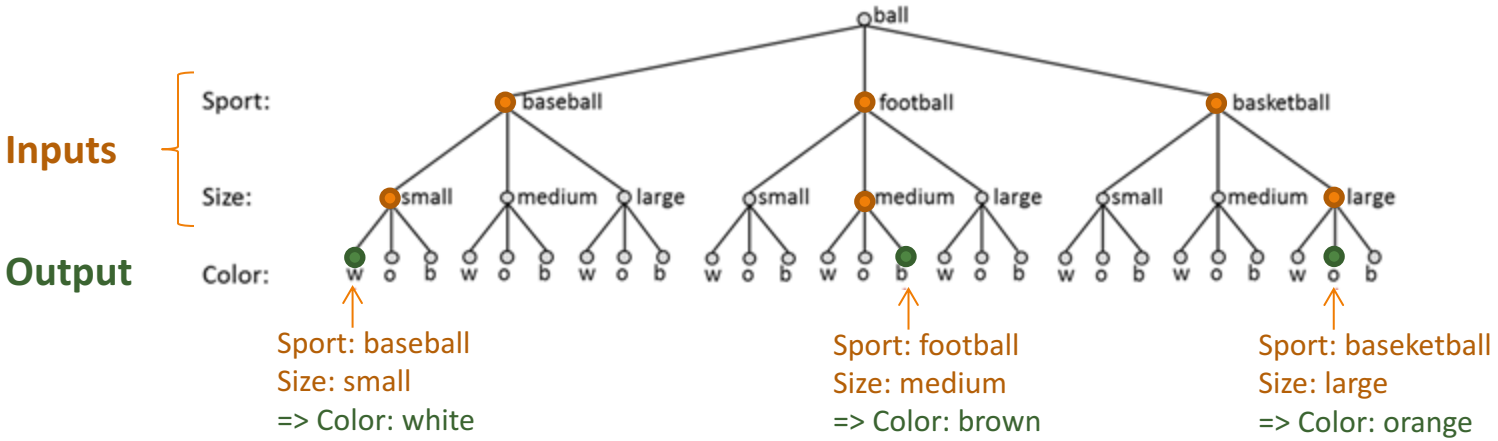




Mapping CDOs to Pattern Form

Constraints:

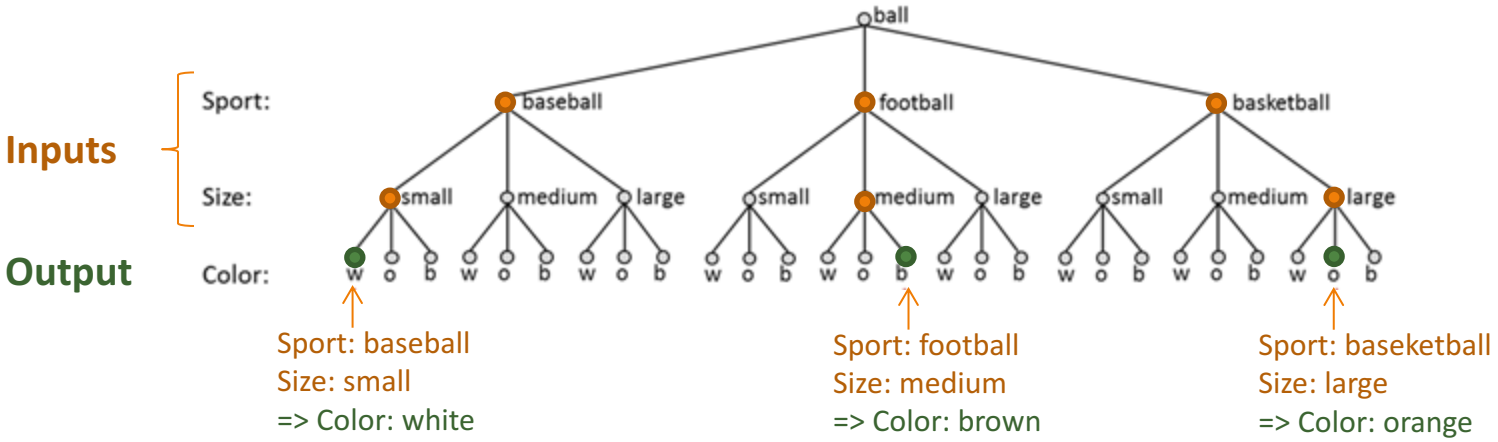
If (sport ==baseball) then (size =small) and (color =white)
If (sport ==football) then (size =medium) and (color=brown)
If (sport ==basketball) then (size =large) and (color=orange)



Mapping CDOs to Pattern Form

Constraints:

If (sport ==baseball) then (size =small) and (color =white)
If (sport ==football) then (size =medium) and (color=brown)
If (sport ==basketball) then (size =large) and (color=orange)



Inputs	**Sport**	Baseball	Football	Basketball
	Size	Small	Medium	Large
Output	**Color**	White	Brown	Orange

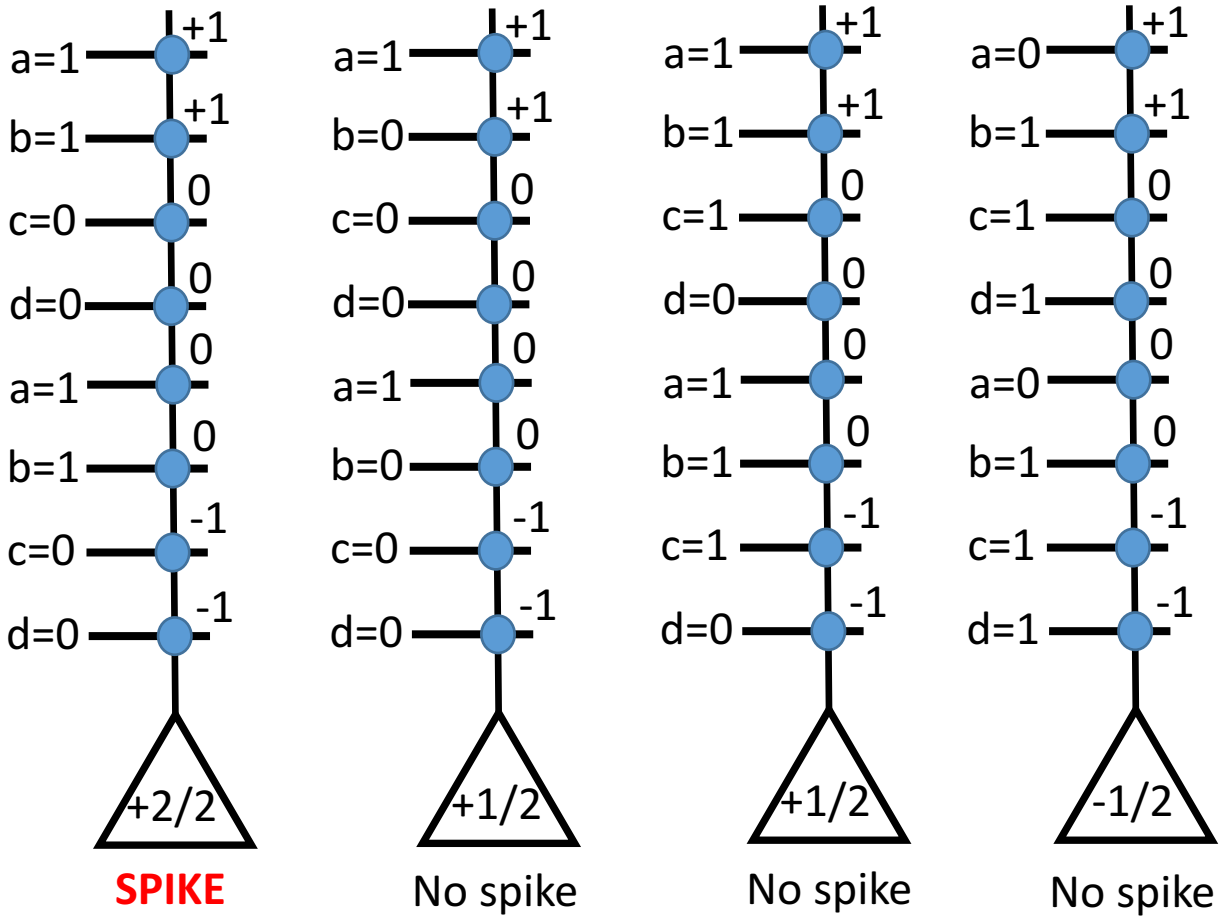
Mapping CDOs to Pattern Form

Inputs	Sport	Baseball	Football	Basketball
Inputs	Size	Small	Medium	Large
Output	Color	White	Brown	Orange

Inputs						Output		
Sport			Size			Color		
Baseball	Football	Basketball	Small	Medium	Large	White	Brown	Orange
1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1

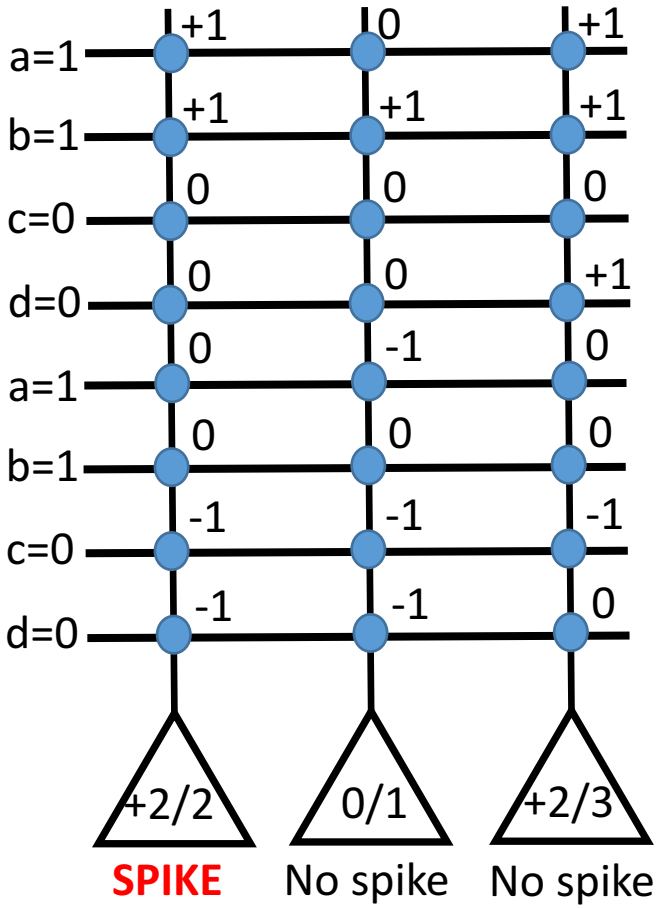
Lookup Table with Neurons

- Neurons spike only for a specific pattern
- Neuron set to fire for input 1100:



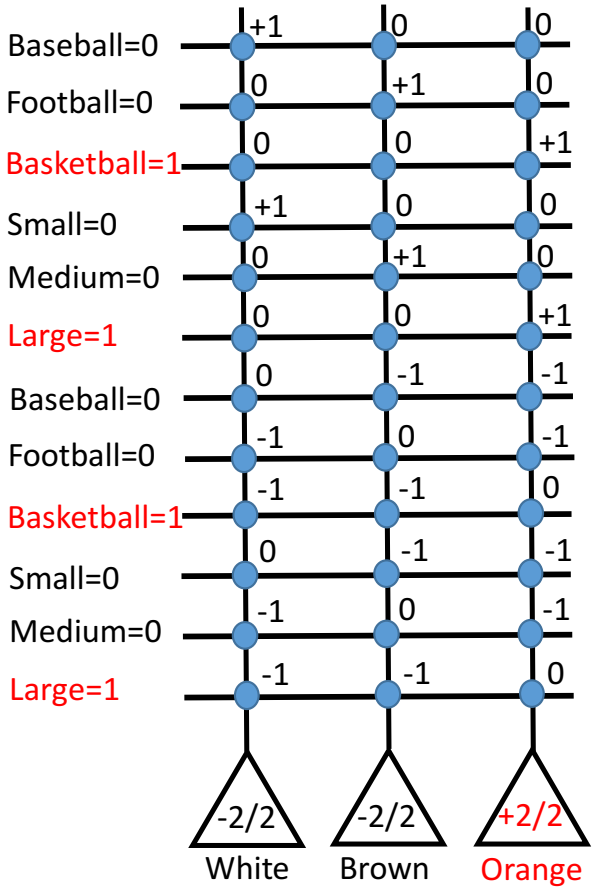
Multiple Patterns in a Crossbar

- Each neuron in a crossbar can detect a different input pattern

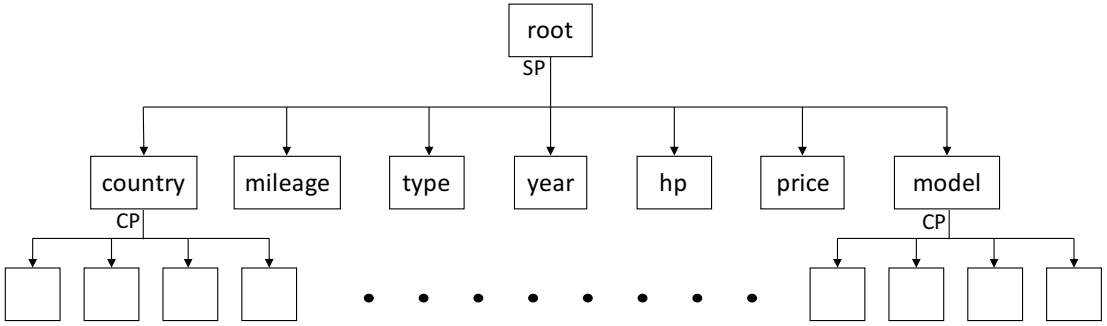


CDOs in Neural Form

Sport	Baseball	Football	Basketball
Size	Small	Medium	Large
Color	White	Brown	Orange



Larger CDO: Vehicles



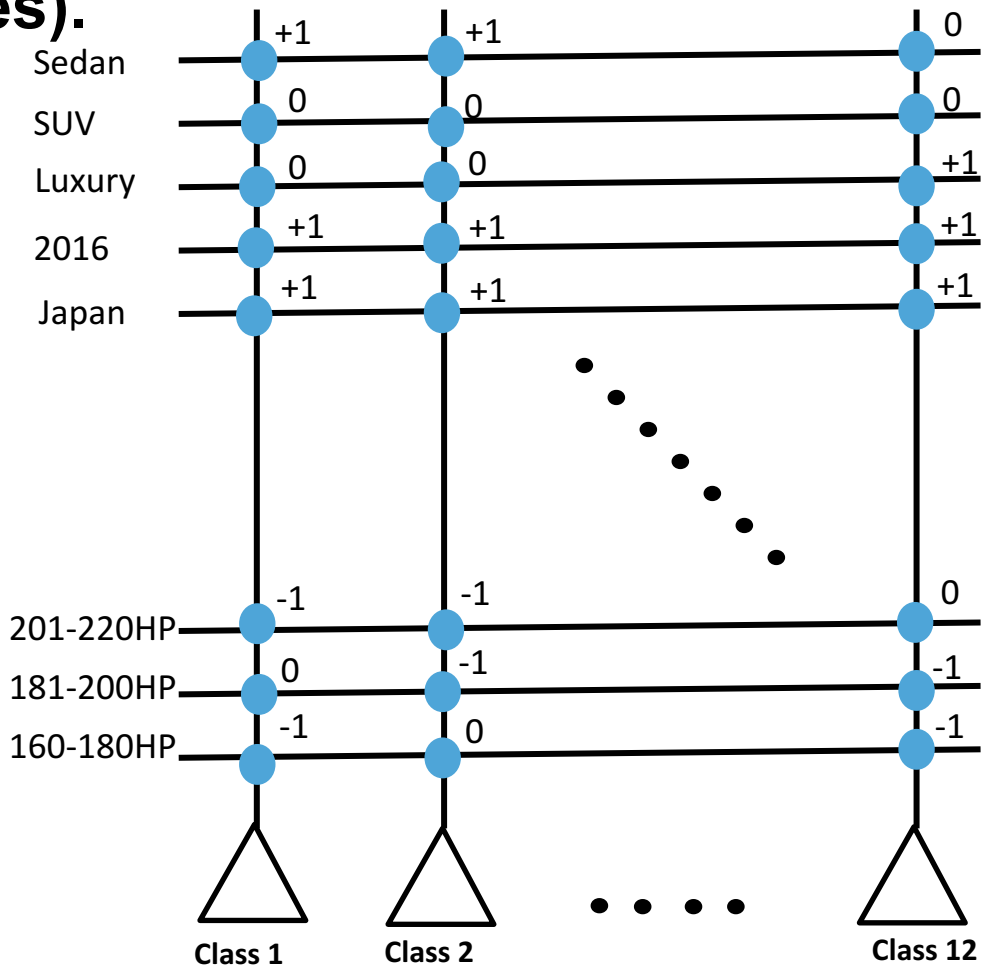
14 constrains
12 classes

Choice-points	Entities
Country	'japan', 'usa', 'south korea', 'germany'
Mileage	'25_above/35_above', '25_below/35_above', '25_below/35_below'
Type	'family sedan', 'mid-size suv', 'luxury coup'
Year	2016
HP	'160-180', '181-200', '201-220', '221-240', '240-above'
Price	$21000-$23000', '$29000-$32000', '$38000-$43000'
Model	Next Slide output Classes.

Class	Content
1	honda_accord, nissan_altima, mazda_mazda6
2	toyota_camry, subaru_legacy
3	chevloret_malibu
4	ford_fusion
5	chrysler_200
6	kia_optima, hyundai_sonata
7	honda_pilot, toyota_highlander
8	dodge_durango, ford_explorer
9	audi_a5
10	bmw_4series
11	lexus_rc
12	inifinit_q6

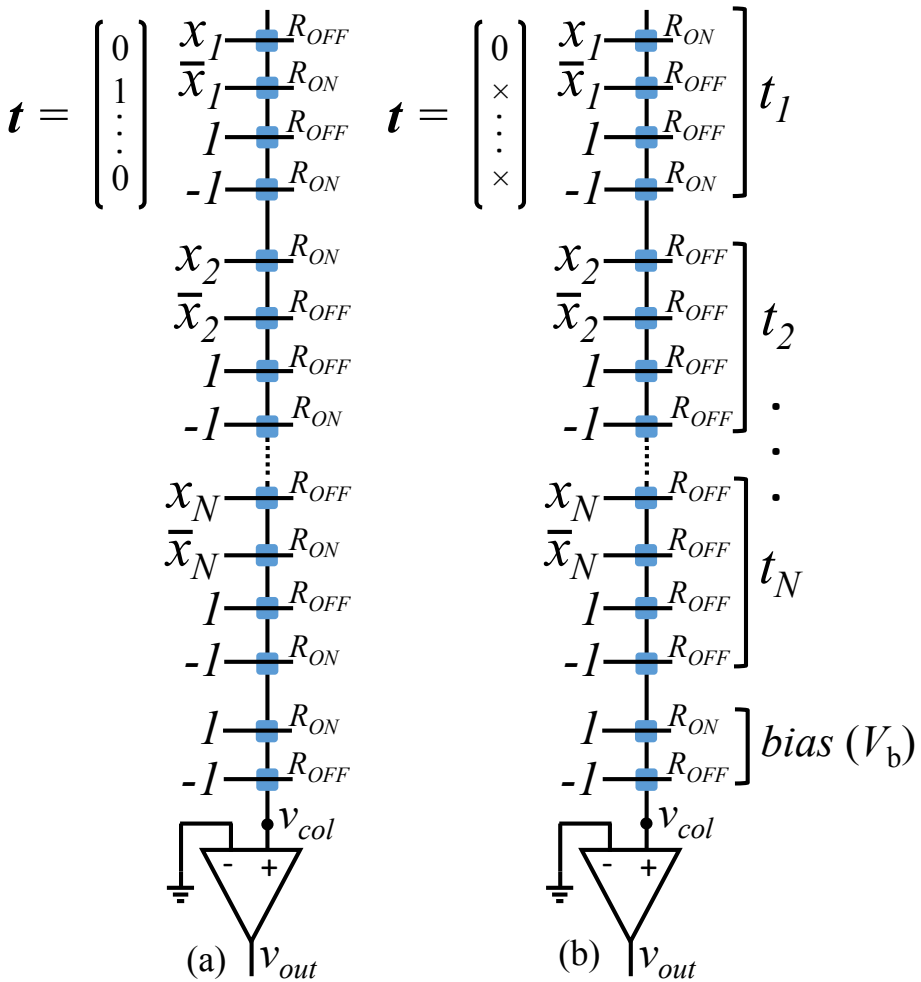
Vehicle CDO into Lookup Table

- 38 input axons. (19 input elements.)
- 12 neurons. (12 classes).



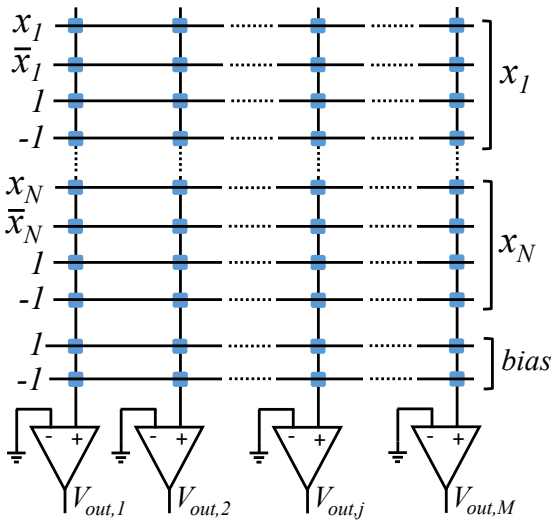
Memristor TCAM

- Four memristors are required to store a bit
  - Input
  - Complement
  - Positive Bias
  - Negative Bias
- R_{ON} represents a strong connection
- R_{OFF} represents no connection
  - There will always be some current through the memristors set to R_{OFF}
  - Biases are used to negate this effect

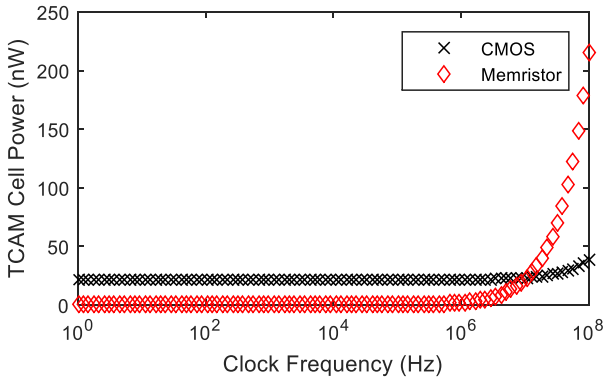


CECEP using Memristor TCAMs

- **Memristor based TCAM: High area efficiency, and hence high amount of knowledge per chip**



TCAM System	Kim CMOS	Huang CMOS	Zheng RRAM	Xu MRAM	Memristor
Configuration	512×144	256×144	256×144	256×144	256×144
Technology (nm)	130	65	180	180	45
Search Time (ns)	4.8	.38	2.3	8	1
Energy Metric (fJ/bit/search)	0.59	0.165	3	7.4	2.15
Area (mm^2)	2.55	0.434	0.167	1.548	**0.0012**
Normalized Area Factor	255	173	8.7	80.6	1



UNIVERSITY of DAYTON

CDOs as CNNs on TrueNorth

- Each network was trained for 8000 iterations
- Testing set = Training set

Layers	Learning Rate	Average Accuracy	Cores	Power (mW)
14	20, 2, 0.2	99.98%	218	52.7
13	20, 2, 0.2	99.97%	127	51.6
12	20, 2, 0.2	99.98%	106	51.3
9	20, 2, 0.2	99.99%	82	51.0
7	10, 1, 0.1	99.99%	74	50.9
6	10, 1, 0.1	99.99%	62	50.8
5	10, 1, 0.1	99.97%	56	50.7
3	10, 1, 0.1	99.99%	36	50.4
2	10, 1, 0.1	98.37%	7	50.0

UNIVERSITY of DAYTON

Conclusion

- CECEP cognitive agent can be mapped to pattern matching form
- Can be implemented on IBM TrueNorth processor
- Can be mapped to memristor crossbars
- Future work: Look at how to solve (as opposed to look up solutions) using neurons