Subiecte IS

# Multiple agent Decisions. Game Theory. AIMA17

## Morra Game.

**O1E1: E=2, O=-2;**

**O1E2: E=-3, O=3;**

**O2E1: E=-3, O=3;**

**O2E2: E=4, O=-4;**

**Show app of max technique;**

**Max equilibrium;**

**Explain Min-Max game tree**

-Poze de pus aici.

## Payoff matrix

Matrix with the values for all agents.

## Prisoners Dilemma

**2 prisoners: Alice and Bob. They can testify or refuse.**

**Ar, Br => -1, -1**

**At, Br => 0, -10**

**Ar, Bt => -10, 0**

**At, Bt => -5, -5**

Using min max we get for Bob testifies: Ur = -10, Ut = -5 => Testifying is a better choice.

Using min max we get for Bob refuses: Ur = -1, Ut = 0 => Testifying is a better choice.

Testifying is the dominant state. This state is also considered an equilibrium state.

## Nash equilibrium

Is achieved when the game reaches a point in which each agent knows that any attempt at drifting from the current state will result in the detriment of itself.

Can be more per game.

## Pareto Optimality

A state/strategy is so when we cannot change it without one of the other agents/players/individuals losing. Also, best state for all players.

Can be more per game.

# Knowledge in learning:

## Foil

**George x Mum -> [Margaret, Elizabeth]**

**Elizabeth x Phillip -> [Charles, Anne, Andrew, Edward]**

**Spencer x Kydd -> [Diana]**

**Diana x Charles -> [William, Harry]**

**Anne x Mark -> [Peter, Zara]**

**Andrew x Sarah -> [Beatrice, Eugenie]**

**Apply FOIL to learn the def of “ancestor/2” function.**

We presume we have the “parent/2” function.

-Solution

We divide the examples in positives and negatives. Used to determine if we need to shrink (AND) or widen (OR) the domain.

We first classify all the examples as true. => All examples are positives => we need to restrict.

We need to at least have 1 common attributes to each side.

Some optimizations are also present that help the algorithm use relevant functions for finding the solution.

Ockhams Razor: a heuristic that helps the alg. Determine what predicates to use next. Ex: by adding parent (a, b) => ancestor (a, b) -> restricts too much.

parent (a, b) = > ancestor (a, c) -> better but not restrictive enough => too many false-positives. (nu tre sa fie adevarate) => need to restrict the domain

parent (a, b) AND ancestor (b, c) => ancestor (a, c) -> we still have false-negatives (nu tre sa fie false). => we need to widen

(parent (a, b) AND ancestor (b, c)) OR parent (a, c) => ancestor (a, c)

## Explanation-Based Learning

**General rules from examples**

-having those examples, we get a deduction tree to which we add predicates till we solve it.

-extracted using variables instead of constants while building the trees

**Proof trees for simplification**

-the drawing to prove the predicate

**Improving efficiency**

- “Prune” -> we exclude some branches in order to obtain some more general rules. (Ex simp: we can stop at primitive/1 or at simplify(y+z,w)).

- 3 influencing factors: Branching factor (nr of used predicates, more is worse); Derived rules should be as general as possible; Choose rules that yield shorter proofs.

- Operationality -> short proogs

**Explicati pe Simplify(1\*(0+X),W)**

**a) Extragere regului generale din exemple**

**b) Imbunatatire eficienta**

-Poza

## Logical formulation of learning

-Find a hypothesis to classify our examples. => find a function

- 2 types of finding them: Current best vs Least statement

- For curr best: Operations to update the hypothesis: Generalization (Add a new case) or Specialization (remove a case).

- For least: At beginning, take all possible statements such that they are either too general or either too specific. Then take each example and test it for each statement => if the statement does not give the desired result, remove it.

Ex: grandparent = parent OR father OR mother OR (parent(x,z) AND parent (z,y)).

**Version space learning**

Is the space between the most general and the most specific hypothesis.

## Inductive logic programming

- 2 main approaches: Top down (FOIL. From general to specific), Inverse resolution.

**Inverse Resolution**

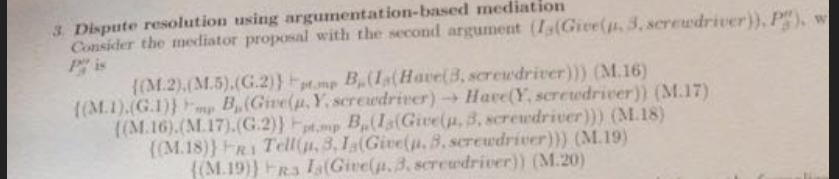
- Start from a specific example and get the general solution

- Poza grandparent

**Making decisions/discoveries (Inverse resolution)**

**-** Adds new predicates to the knowledge base if you need them to solve this function. (Ex. We may not have parent(G,\_), so we add one).

# Dispute resolution. Argumentation-based meditation. Trescak



-(1)Picture: Mediator tells an agent that he needs a screwdriver to hang the mirror and must request it from him.

-(2)Alternate Picture: How mediator figures out Y needs nail from other agent and that he needs to request it.

- A new agent, mediator, is used to solve disputes between multiple agents.

**Agent theories**

- Properties that must be respected by ALL agents (Agent theories)

G1-Ownership (X gives to Y sth => Y is the new owner of that sth)

G2-Reduction (Agent splits the big goal in smaller ones to solve first; Divid et impera)

G3-Generosity (Mediator wants to give all of his resources to other agents)

G4-Unicity (X, after giving someone sth, does not longer contain that sth)

G5-Benevolence (If X does not need sth, he will give it up if someone needs that sth)

G6-Parsimony (An agent will not do sth if he does not need to)

G7-Unique choice (If there are multiple ways to do sth, X will only chose one).

**Bridge rules**

-Properties for communication between agents.

R1/R2- Advice => Mediator will tell an agent how to operate, if the mediator thinks he knows what the agent is about to do OR how he wants to do sth.

R3- Trust in mediator => Agents trusts what the mediators tell them to do

R4- Request => If an agent needs sth, he will request it

R5- Accept request => If the agent can give a requested resource, he will give it.

**Arguments necessary to complete resolution.**

**-**(1) B ask A to give SCREW. B asks M to give SCREWDRIVER.

-(2) M infers that A to request NAIL from B. A requests nail from B.

# Making complex Decisions:

-When going from State S1 to state S2 we get a reward (can be + or -).

-A reward has a discount, that tells how far in the future that reward is.

## Value Iteration

**Utilities of states**

**-**The utility of a state is the expectancy of the sum of states that have a discount for a given policy.

-The TRUE utility of a state is the same but for the best possible policy.

**Bellman eq**

-Used to determine the utility of states.

-Utility(S1) = reward(S1) + discount(S2) \* max(utility). (utility of possible states from our current one)

-We can get multiple systems of bellman equations (that each have an unique solution) to determine the TRUE utility.

**Value iteration alg**

-We start from a random/approximate utility and we update it until the difference between 2 consecutive solutions is negligible.

-We use Bellman eq (Bellman update) to calculate the next utility: Uk+1 = max(Uk) \*discount(S2) + reward(S1).

**Policy iteration**

- Same as value but we update the policy.

- 2 Parts: Policy evaluation (Calc the utility of policies), Policy improvement (Choses policy with highest utility)

-U = policy evalution.

-foreach(state) Search what is the best move to do from all policies.

-\*\*\*\*

## Policy. Multiple sections.

**Policy loss**

**-** The most an agent can lose by using a policy other than the BEST policy.

**Modified policy iteration**

**-** We modify the Bellman eq: We need not calculate the max(U), since we will have a fixed policy. This applies to the policy evaluation step of the algorithm.

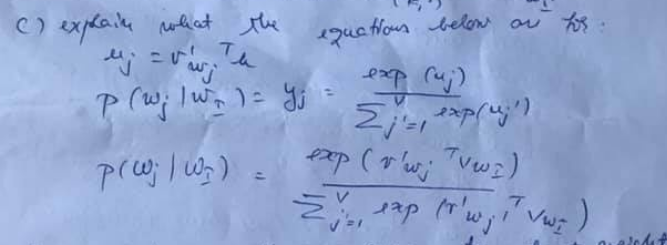
**Partialy observable MDP**

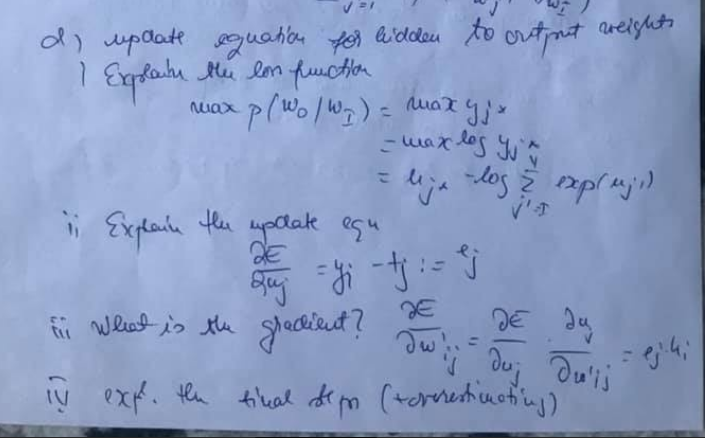
-

# Neural Networks. CBOW model. 1-word context

**Draw+explain the wildszec model uinder the one context simplification**

**Explain the working of the weights in the model: de=WTx = WT(k,\*) := vTW1**



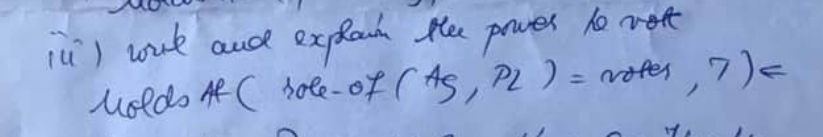


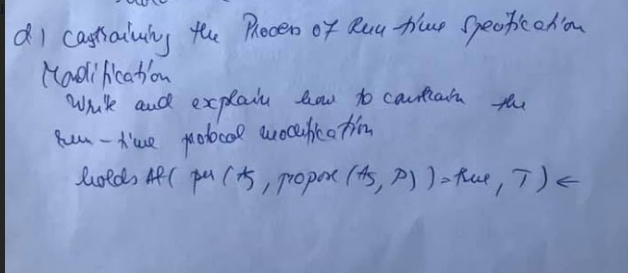
# Dynamic protocols for open agent Systems: CFCP

**Explain the figure for the cleaining bloor control protocol**

**Table for the specification of the conditions of a CFCP participant with phys capabilities, some power, permissiona and obligation to perform an action**

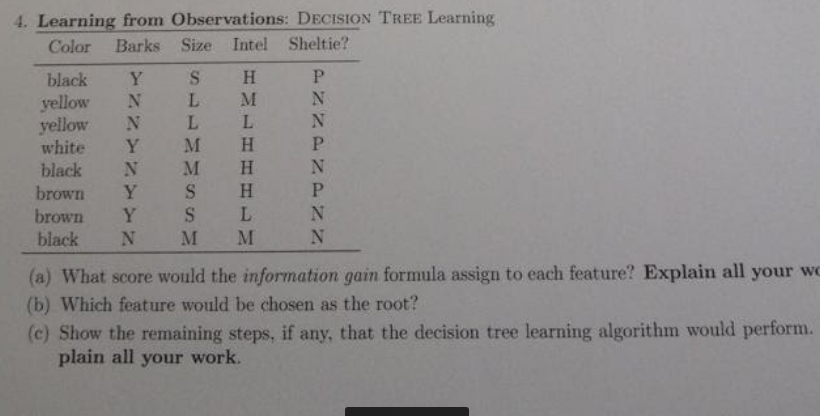
**Transition protocol: tule-set replacement; power to second a protocol;**





# Learning from observation. Decision Tree learining

**AdaBoost problem.**



Se dau urmatoarele features: F1 in {A,B}, F2 in

{C,D}, F3 in {E,G,H} si urmatoarele training examples:

A C H +

A D G +

A C G -

A D G +

B D H -

B D H -

B C H +

B C G -

(a) what score would the info gain formula assign to

each of 3 features?

(b) which would be the root of the tree?

(c) show remaining steps, if any, that decision tree

lerning algorithm would perform using the above

examples(d) what kind of search stratey?

# Communication.

## Semantic analysis

**Tim+sense**

**quantification**

## RNN abstractions

## Ambiguation and Disambiguiation

**Ambiguity syntactic/semantic**

**Metonimy5**

**Metaphor**

**Model FO disambiguation**

## Augmented grammars

# Agents that communicate

**write down the lexicon and the grammar rules for**

**"wumpus pidgin" (e0)**

**(a) which of the following are generated by the**

**grammar (why or why not?)?**

**se dau trei prop**

**(b) modify the rule for "the wumpus that the dogs see**

**stinks" to be generated**

**by the gramar**

**(c) give the parsing tree for this using your new rule**

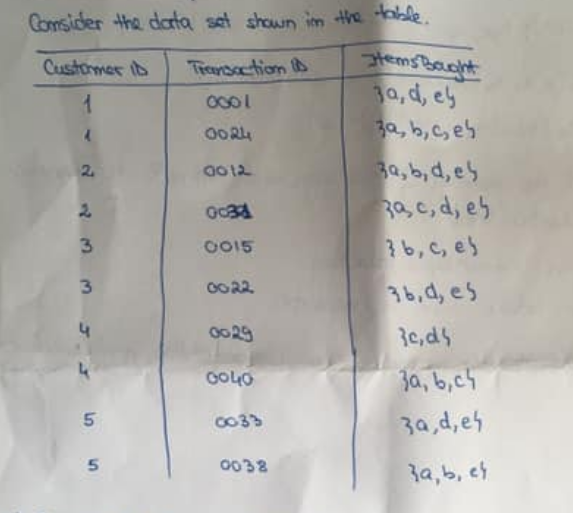
**(d) adjust grammar to allow 'the wumpus the dogs see**

**stinks", but to disallow "the wumpus the dogs I smell**

**see stinks"**

# Multiagent Communication for Intersection.

# Association analysis



**Support for itemsets {e}, {b,d}, {b,d,e} by treting intersection as market basket.**

**Compute the confidence for some association rules. Are they a sym measure?**

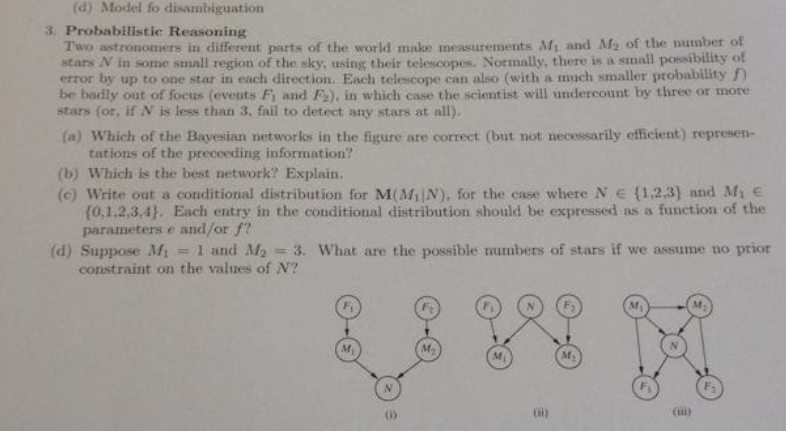
**Treat each customer ID as a market bucket. Item = binary variable.**

**Compute new confidence association.**

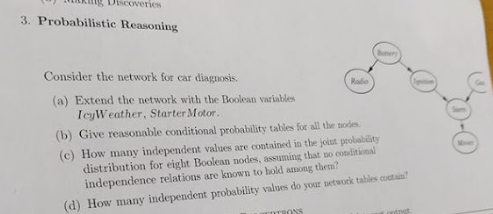
## Apriori principle

# Probabilistic Reasoning

## Astronauts



## Car problem



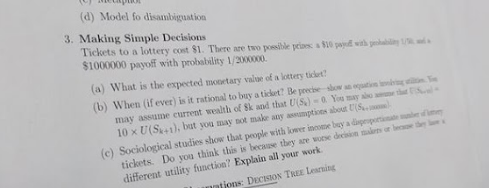
## Kalman Filters (Outdated)

**1-dimensinal example**

**Applicability**

# Making simple Decisions

## Lotto problem



# Semantics

## Wumpus

**It is a Wumpus. The Wumpus is dead. Wumpus is in 2.2**

**Can u define “It is a Wumpus” as “x belongs to Wumpus(x)”**

**Hint: “It was a a Wumpus”**

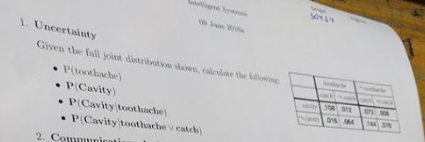
# Value based plan selection in BDI angets.

## No pizza 4 u.

# Uncertainty (Posibil sa nu intre dar ii bulangiu si facem totusi)

## Wumpus world

## Dentist



## Extra

**Explicati esenta urmatoarelor sectiuni din acest capitol**

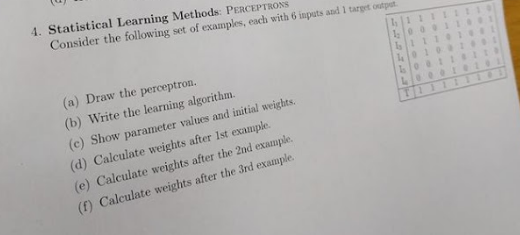
**a) The axioms of probability**

**b) Inference using ful join destribution**

**C) Independence**

# Statistical Learining Methods

## Perceptrons



# Planning (Outdated)

**Se da functie Ride(x,e,f1,f2) descrisa in STRIPS**

**care duce persoana x, cu liftul e, de la etajul f1 la**

**f2.**

**(a) write down a definition for Call(x,e,f) - care**

**cheam liftul la etajul f**

**(b) wite down an effect axiom for Ride**

**(c) write down a frame axiom needed for this world**

**(d) Jeb from floor 2 wants to go to 3, with the only**

**working elevator E, which is at floor 7. Using the**

**graphical notation for plans, give the initial**

**empty plan.**

**(e) Add the ride step to this plan**

**(f) Is there more than 1 way to do this?**

# (Outdated) Reinforcement Learning

**Explicati esenta urmatoarelor sectiuni din acest capitol**

**a) Passiv reinforcement learning**

**b) Activ reinforcement learning**

**c) Generalization in reinforcement learning**