CES -161 - Modelos Probabilísticos em Grafos

Sequential Decision Making with Bayesian Networks

Prof. Paulo André Castro
pauloac@ita.br
www.comp.ita.br/~pauloac Sala 110, IEC-ITA

Sequential Decision Making

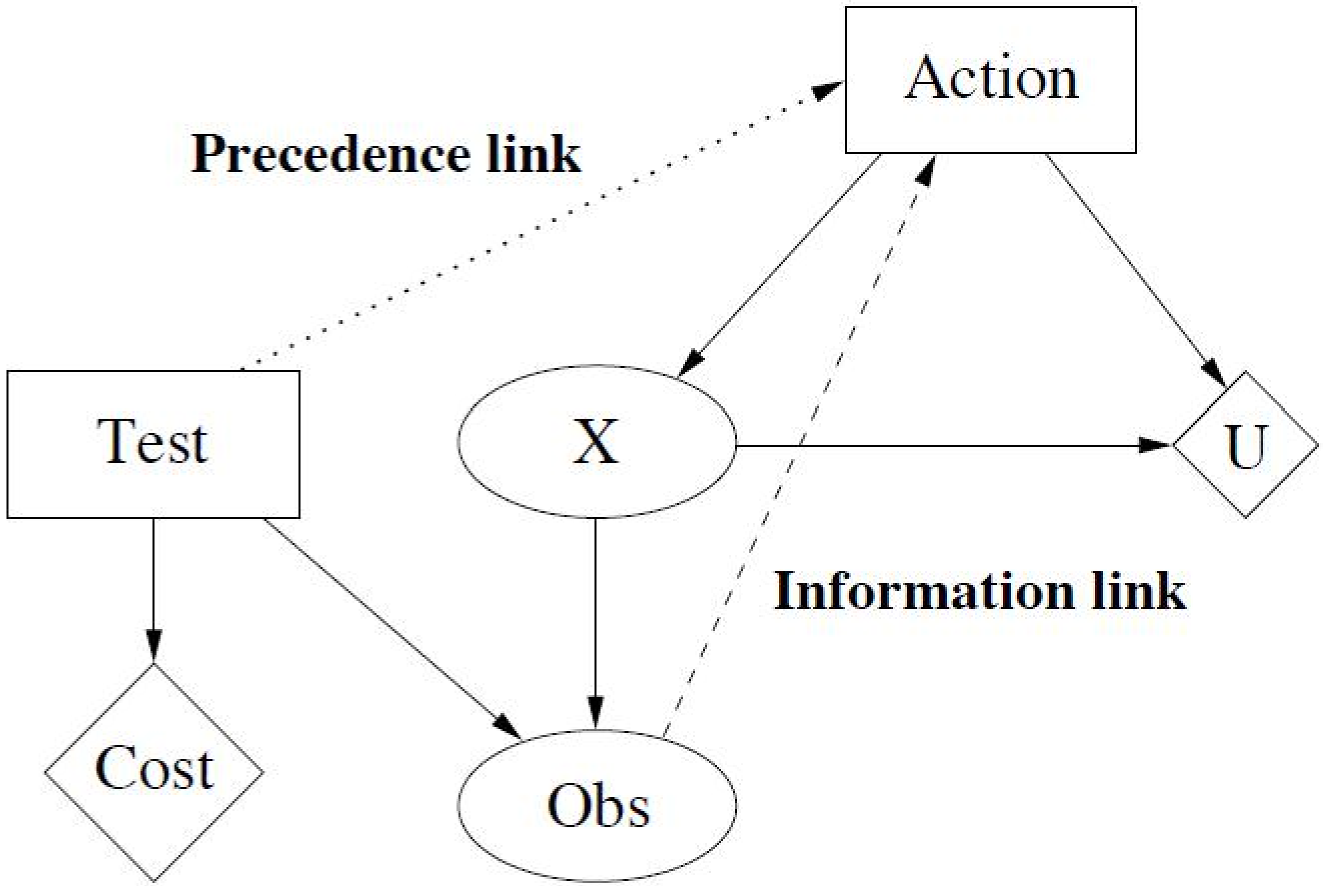
- Thus far, we have considered only single decision problems. Often however a decision maker has to select a sequence of actions, or a plan. Sometimes, a sequence of action is not enough and it is required a function given observations

- In the football decision problem used before, Clare might have a choice as to whether to obtain the weather forecast (perhaps by calling the weather bureau)

- In the diagnostics example, the physician must decide whether to order another exam, before deciding on a treatment option.

- This type of decision problem has two stages:
  1. The decision whether to run a test or make an observation
  2. The selection of a final action

A decision network showing the general structure for these test-act decision sequences....



Test-action Decision Sequence

- If the decision is made to run the test, evidence will be obtained for the observation node Obs, before the Action decision is made; hence there is an information link from Obs to Action.

- The question then arises as to the meaning of this information link if the decision is made **not to run the test**. This situation is handled by adding an additional state, **unknown**, to the Obs node and setting the CPT for Obs

$$P(Obs = unknown | Test = no) = 1$$
$$P(Obs = unknown | Test = yes) = 0$$

Test-action Decision Sequence - 2

- In this generic network, there are **arcs from the Action node to both the chance node X and the utility node U**, indicating intervening actions with a direct associated cost. However, either of these arcs may be omitted, representing a non-intervening action or one with no direct cost, respectively

- There is an implicit assumption of no-forgetting in the semantics of a decision network. The decision maker remembers the past observations and decisions, indicated explicitly by the information and precedence links

Algorithm for Test-action Decision Sequence

1. *Evaluate decision network with any available evidence (other than for the Test result).*

   Returns Test decision.

2. *Enter Test decision as evidence.*
3. *If Test decision is 'yes'*

   Run test, get result;

   Enter test result as evidence to network.

   Else

   Enter result 'unknown' as evidence to network.

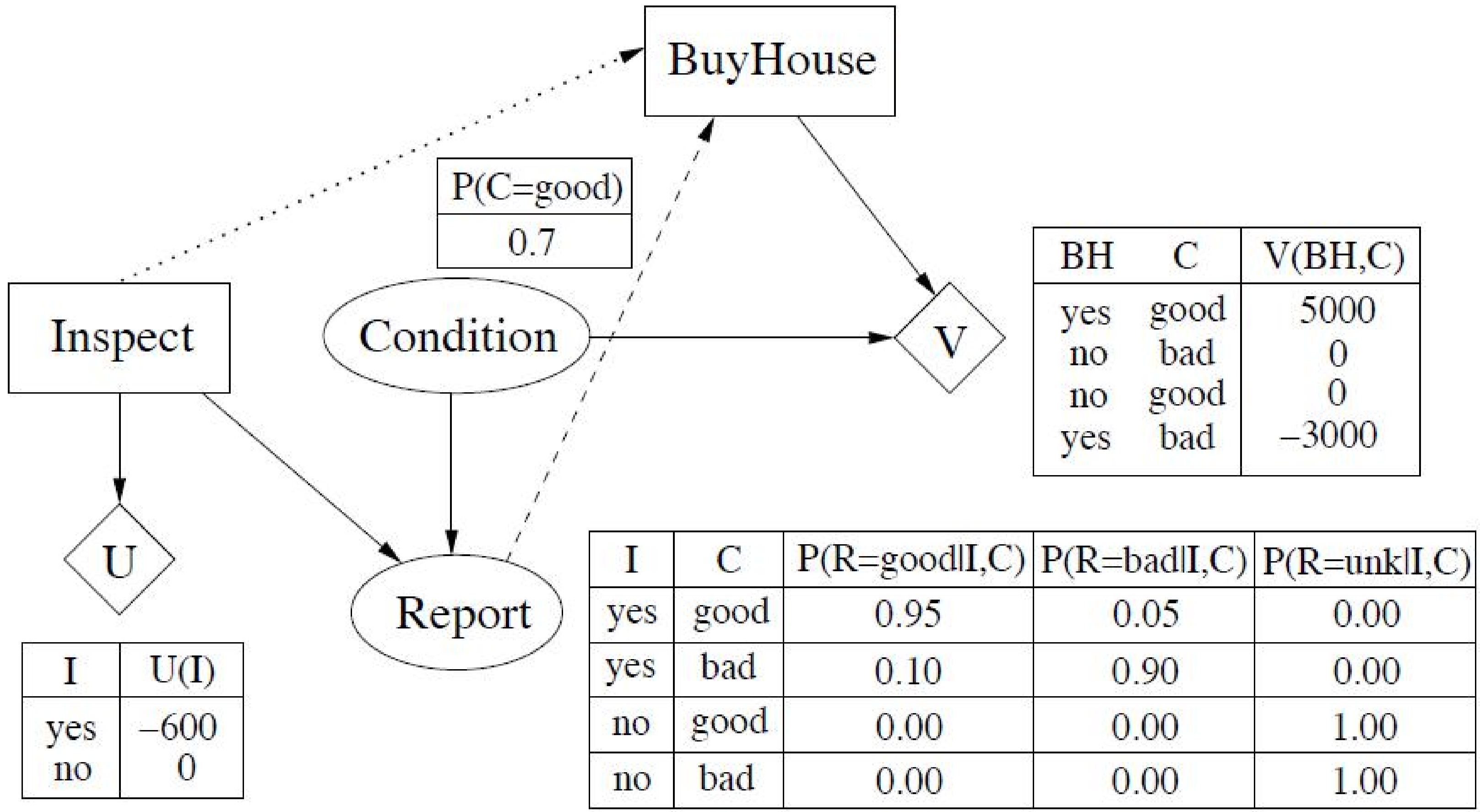
4. *Evaluate decision network.*

   Returns Action decision.

Real estate investment example

Paul is thinking about buying a house as an investment. While it looks fine externally, he knows that there may be structural and other problems with the house that aren't immediately obvious. He estimates that there is a 70% chance that the house is really in good condition, with a 30% chance that it could be a real dud. Paul plans to re-sell the house after doing some renovations. He estimates that if the house really is in good condition (i.e., structurally sound), he should make a \$5,000 profit, but if it isn't, he will lose about \$3,000 on the investment. Paul knows that he can get a building surveyor to do a full inspection for \$600. He also knows that the inspection report may not be completely accurate. Paul has to decide whether it is worth it to

A Decision Network



P(C=good)
0.7

BH	C	V(BH,C)
yes	good	5000
no	bad	0
no	good	0
yes	bad	–3000

I	U(I)
yes	–600
no	0

I	C	P(R=good|I,C)	P(R=bad|I,C)	P(R=unk|I,C)
yes	good	0.95	0.05	0.00
yes	bad	0.10	0.90	0.00
no	good	0.00	0.00	1.00
no	bad	0.00	0.00	1.00

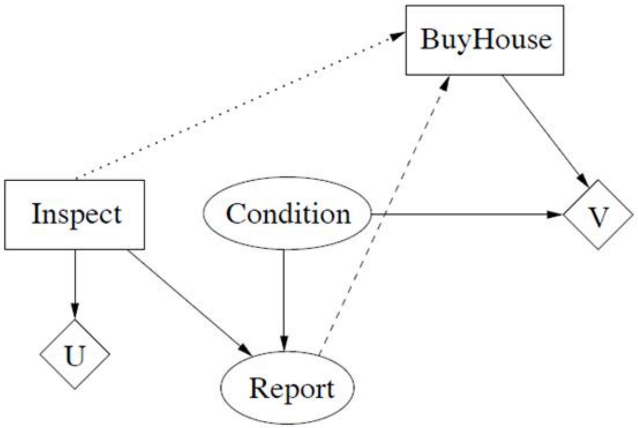
Evaluation using a decision tree model

- In order to show the evaluation of the decision network, we will use a decision tree representation

- To understand a decision tree, we start with the root node, which in this case is the first decision node, whether or not to inspect the house. Taking the path from the root to leaves each path means:
  - From a decision node, it indicates which decision is made
  - From a chance node, it indicates which value has been observed

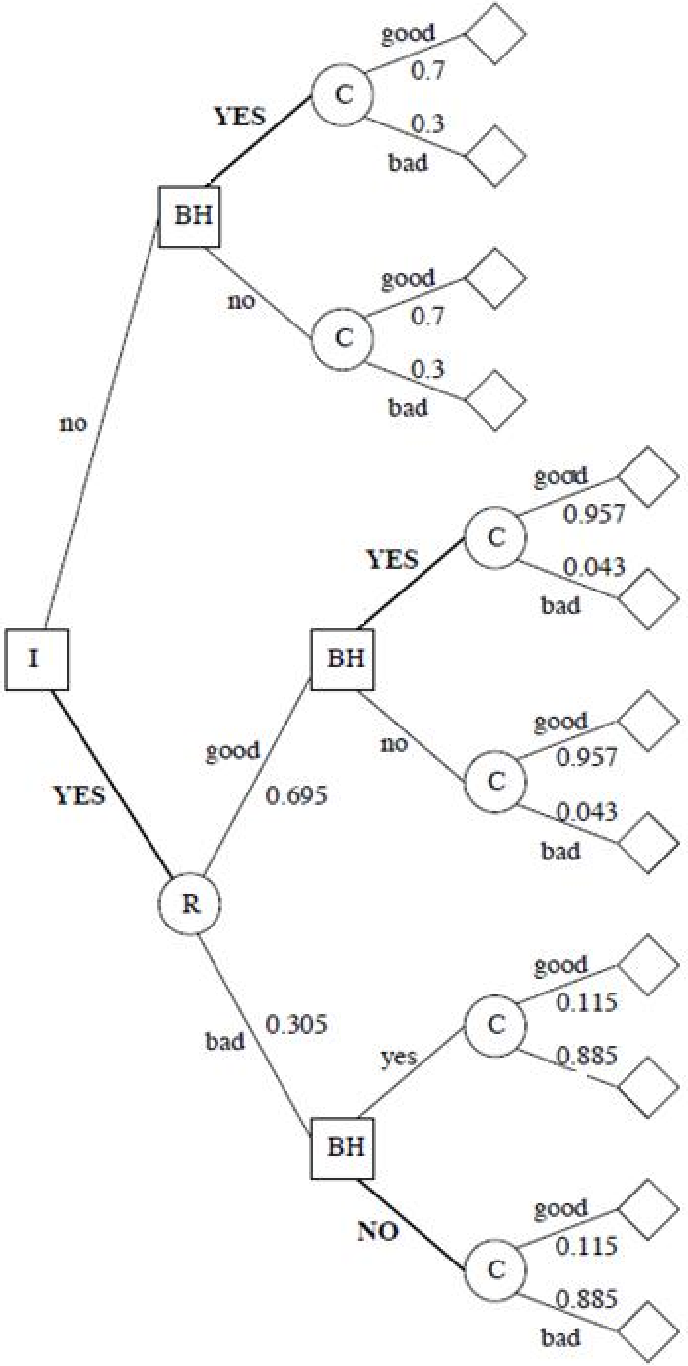
Evaluating by Decision tree

- When Paul decides about Inspection, he doesn't have any information about the chance nodes, so there are no information links entering the Inspect decision node.



- When he decides whether or not to buy, he will know the outcome of that decision hence the **information link** from Report to BuyHouse.
- The temporal ordering of his decisions, first about the inspection, and then whether to buy, is represented by the **precedence link** from Inspect to BuyHouse.
- Note that there is a directed path from Inspect to BuyHouse (via Report) so even if there was no explicit precedence link added by the knowledge engineer for this problem, the precedence could be inferred from the rest of the network structure

- I: Inspect the house
- BH: Buy the house
- C: Condition
- R: Report



- Note: We could include
- R=unknown, when I=no
- But it wouldn’t change
- anything

How to decide? Decision Tree Evaluation Algorithm

1. *Starting with nodes that have only leaves (utility nodes) as children.*
2. *If the node X is a chance node, each outgoing link has a probability and each child has an associated utility. Use these to compute its expected utility*

$$EU(X) = \sum_{C \in Children(X)} U(C) \times P(C)$$

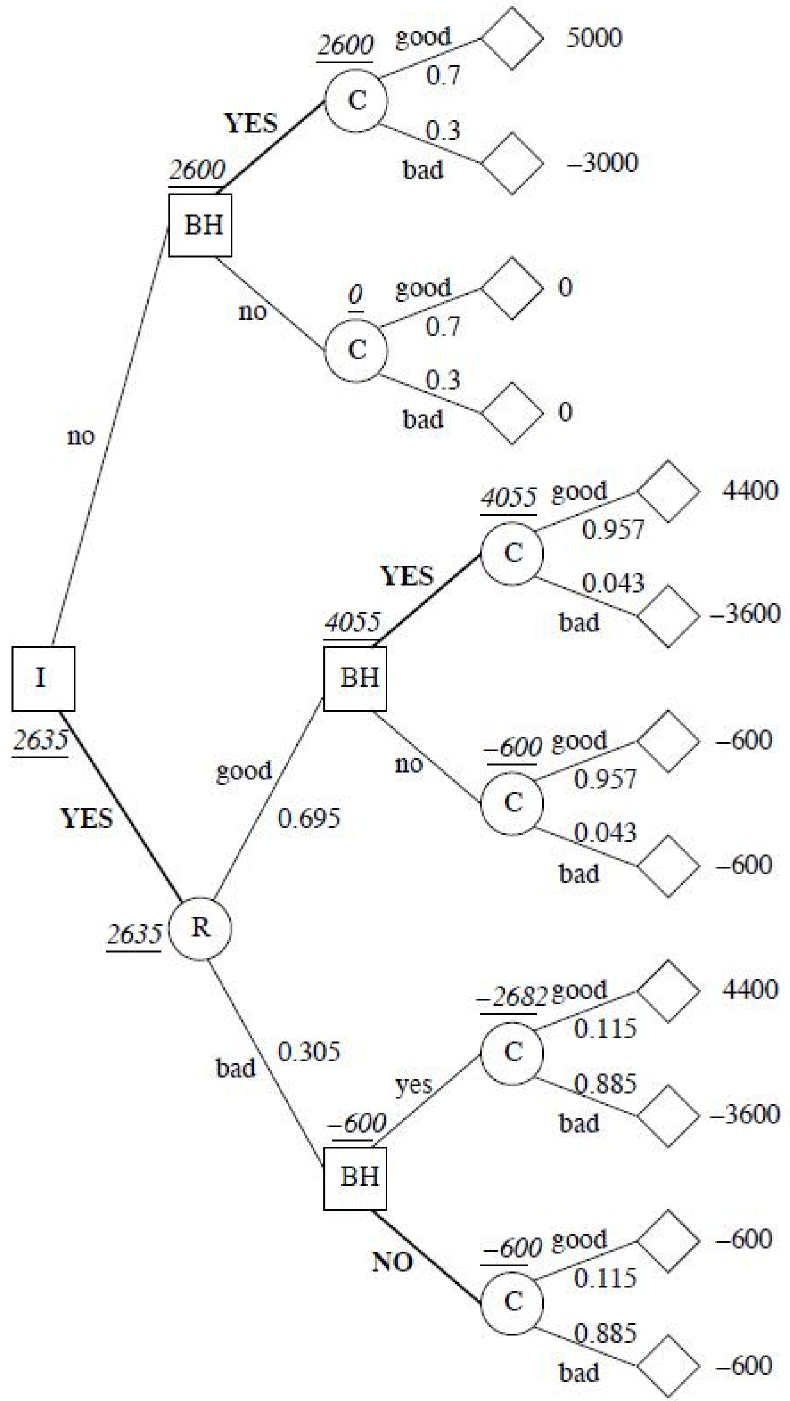
If the node is a decision node, each child has a utility or expected utility attached. Choose the decision whose child has the maximum expected utility and

$$EU(X) = max_{C \in Children(X)}(EU(C))$$

3. *Repeat recursively at each level in the tree, using the computed expected utility for each child.*
4. *The value for the root node is the maximal expected utility obtained if the expected utility is maximized at each decision.*

Evaluated Decision Tree

Expected Utilities are shown underlined



Decisions and their Expected Utilities

Decisions calculated for the real estate investment problem.

Evidence	*Bel(C=good)*	EU(I=yes)	EU(I=no)	Decision
None	0.70	**2635**	2600	I=yes
Given *I=no*		EU(BH=yes)	EU(BH=no)	
Report=unknown	0.70	**2600**	0	BH=yes
Given *I=yes*		EU(BH=yes)	EU(BH=no)	
Report=good	0.957	**4055**	-600	BH=yes
Report=bad	0.115	-2682	**-600**	BH=no

- The report may change the decision of buying the house!

Value of information

- The decision of whether to gather new information is based on the value of the information.

$$EB(Test) = EU(Test = yes) - EU(Test = no)$$

- In the real estate investment problem,

$$\begin{aligned} EB(Inspect) &= EU(Inspect = yes) - EU(Inspect = no) \\ &= 2635 - 2600 = 35 \end{aligned}$$

- Note that it already computes the cost of the inspection. So the price is worth paying

Sequential Decision Making

- Sequential Decision Making may be approached using another type of Graph Probabilistic Model (Modelos Probabilísticos em Grafos)
  - Markov Chain (Redes ou Cadeias de Markov)

- Sequential Decision making is complex in Bayesian Networks, one way of dealing with it is using Dynamic Bayesian Network (DBN) which is going to be our next subject

Dynamic Bayesian networks

Dynamic Bayesian networks

- DBN are also called dynamic belief networks (Russell and Norvig, 1995, Nicholson, 1992), probabilistic temporal networks (Dean and Kanazawa, 1989, Dean and Wellman, 1991) and dynamic causal probabilistic networks (Kjærulff, 1992).

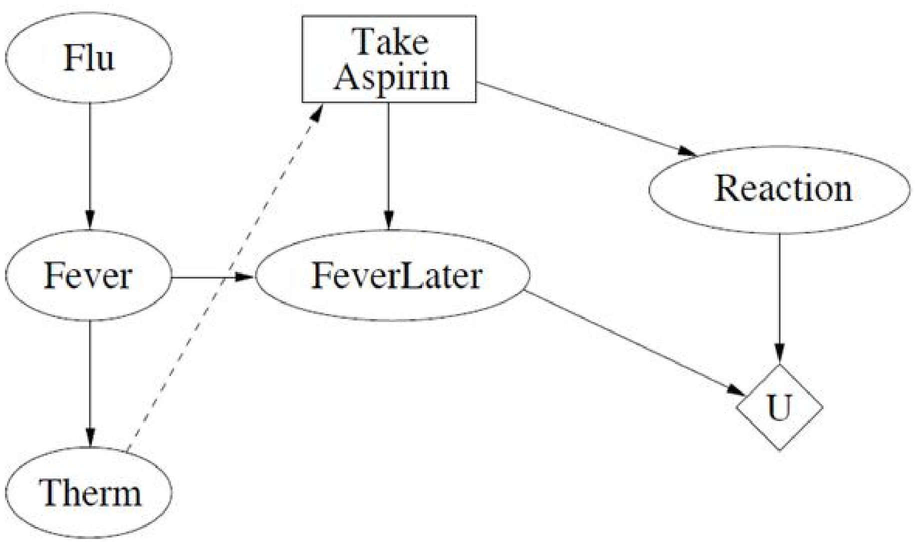
- Dynamic Bayesian networks (DBNs) explicitly model change over time.

- In the next chapter, we will extend these DBNs with decision and utility nodes, to give dynamic decision networks, which are a general model for sequential decision making or planning under uncertainty.

Bayesian Networks and time

- Although a causal relationship represented by an arc implies a temporal relationship, BNs do not explicitly model temporal relationships between variables.

- And the only way to model the relationship between the current value of a variable, and its past or future value, is by adding another variable with a different name. We saw an example of this with the fever example earlier with the use of the FeverLater node.



BN and time

- A Bayesian network is defined by a set of random variable and arcs connecting them

$$\mathbf{X} = \{X_1, \ldots, X_n\},$$

- When constructing a DBN for modeling changes over time, we include one node for each X_i for each time step. If the current time step is represented by t, the previous time step by t-1, and the next time step by t +1, then the corresponding DBN nodes will be:

- Current: $\{X_1^t, X_2^t, \ldots, X_n^t\}$
- Previous: $\{X_1^{t-1}, X_2^{t-1}, \ldots, X_n^{t-1}\}$
- Next: $\{X_1^{t+1}, X_2^{t+1}, \ldots, X_n^{t+1}\}$

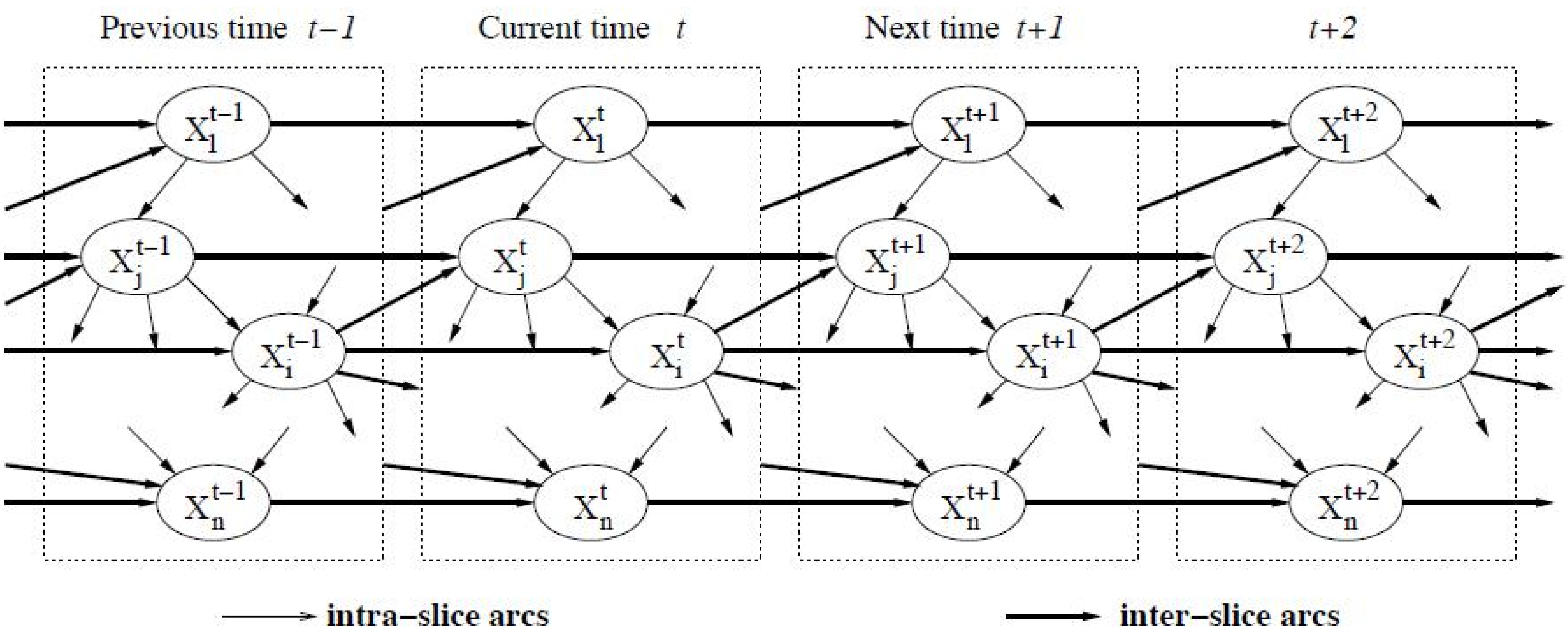
Dynamic Bayesian Network

- The relationships between variables in a time-slice are represented by **intra-slice arcs**, Although it is not a requirement, the structure of a time-slice does not usually change over time

- The relationships between variables at successive time steps are represented by **inter-slice arcs**, also called **temporal arcs**, including relationships between the same variable over time, i.e:

  - $X_i^{t} \longrightarrow X_i^{t+1}$

General structure of a Dynamic Bayesian Network



- Note that there are no arcs that span more than a single time step. This is another example of the Markov assumption

Variables and their relations intra-slice and inter-slice

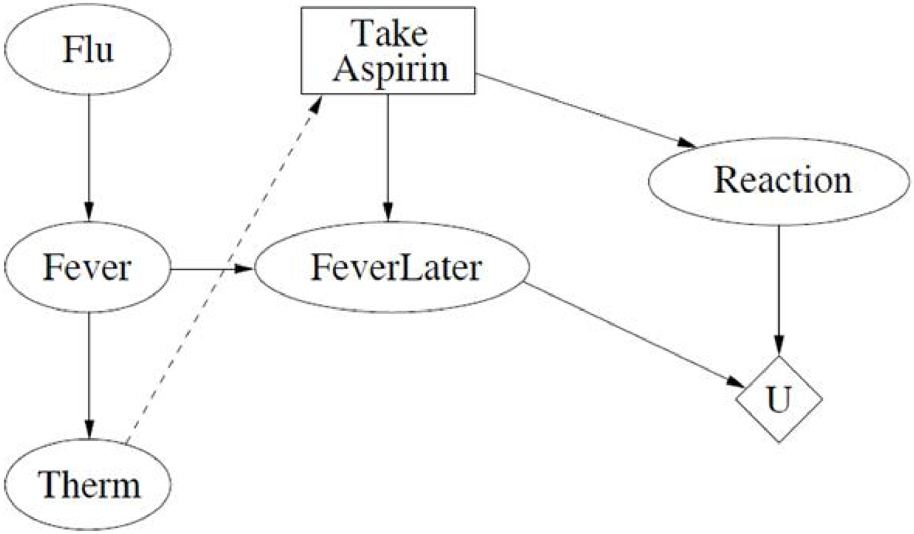
The relationships between variables, both intra-slice and inter-slice, are quantified by the conditional probability distribution associated with each node. In general, for node X_i^t with intra-slice parents $Y_1^t, \ldots, Y_m^t$ and inter-slice parents X_i^{t-1} and $Z_1^{t-1}, \ldots, Z_r^{t-1}$, the CPT is

$$P(X_i^t | Y_1^t, \ldots, Y_m^t, X_i^{t-1}, Z_1^{t-1}, \ldots, Z_r^{t-1}).$$

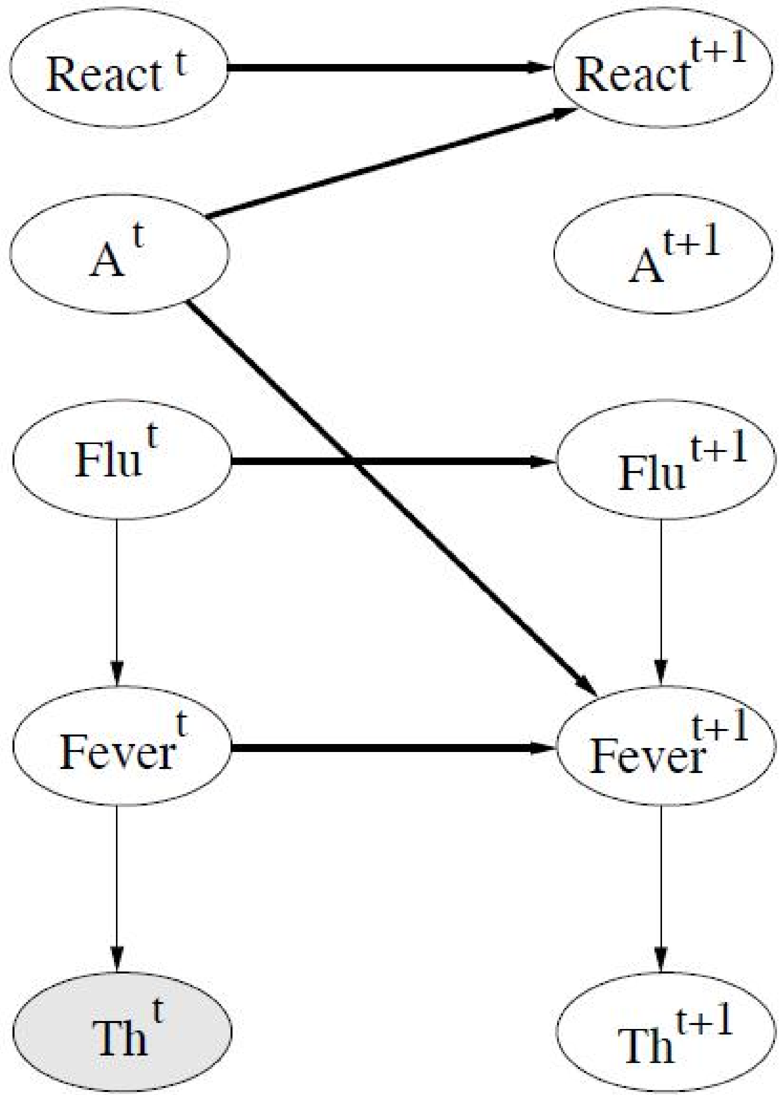
- Given the usual restriction that the networks for each time slice are exactly the same and that the changes over time also remain the same (i.e., both the structure and the CPTs are unchanging), a DBN can be specified very compactly. The specification must include:
  - Node names
  - Intra-slice arcs
  - Temporal (inter-slice) arcs
  - CPTs for the first time slice t0 (when there are no parents from a previous time)
  - CPTs for t +1 slice (when parents may be from t or t +1 time-slices).

The Fever Aspirin Example

Suppose that you know that a fever can be caused by the flu. You can use a thermometer, which is fairly reliable, to test whether or not you have a fever. Suppose you also know that if you take aspirin it will almost certainly lower a fever to normal. Some people (about 5% of the population) have a negative reaction to aspirin. You'll be happy to get rid of your fever, as long as you don't suffer an adverse reaction if you take aspirin.



DBN Fever Aspirin Example



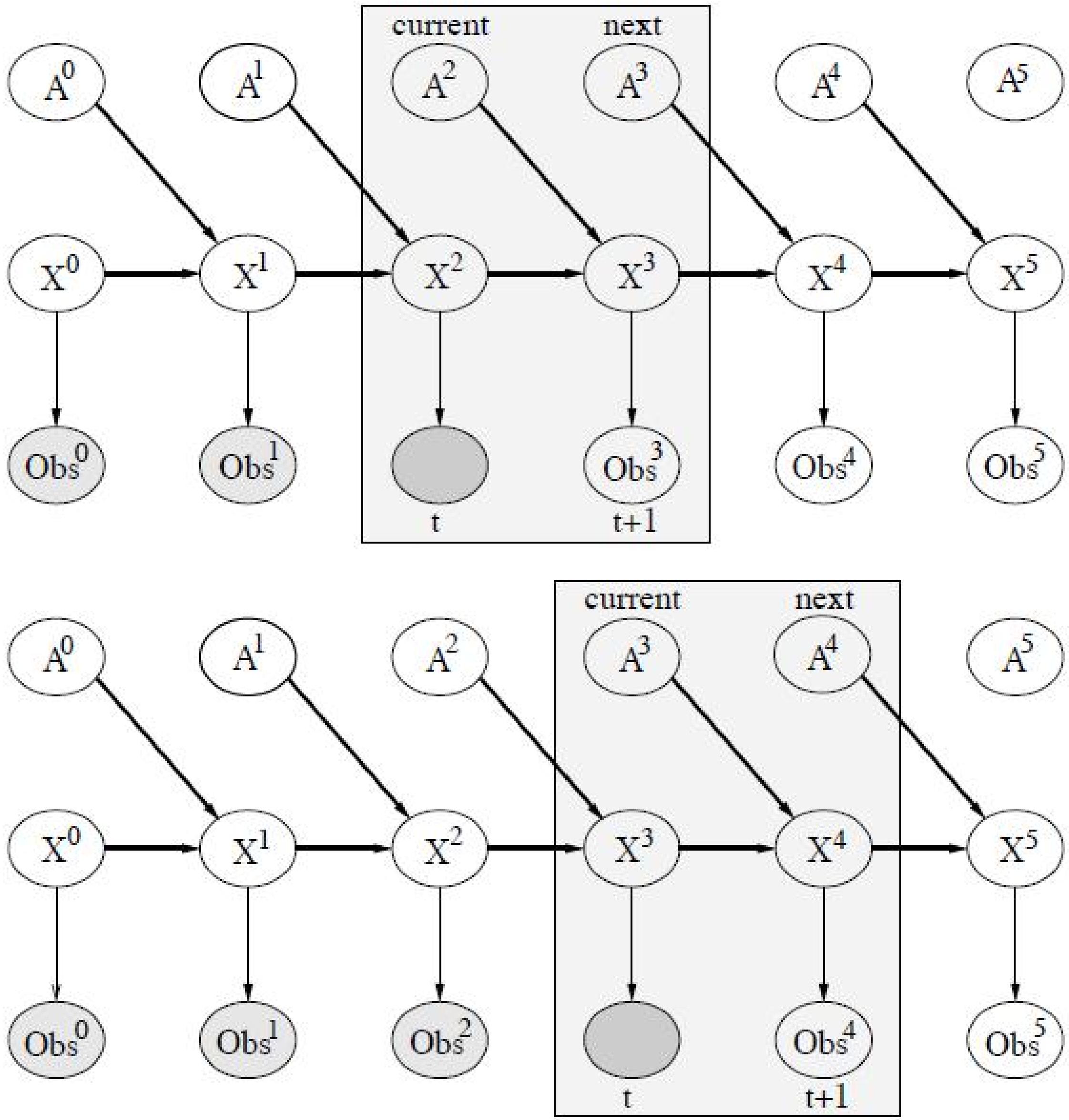
Reasoning in DBN

- Given evidence about a set of nodes, from the first time slice up to and including the current time-slice t, we can perform belief updating on the full DBN, using standard BN inference algorithms.

- This means obtaining new posterior distributions for all the non-evidence nodes, including nodes in the t+1 and later time-slices. This updating into the future is called **probabilistic projection**

- However, this type of DBN gets very large, very quickly, especially if the interval between time slices is short. To cope, in most cases the DBN is not extended far into the future. Instead, a fixed size, **sliding "window"** of time slices is maintained.

DBN and sliding "window" of two time-slices (Shading indicates evidence node.)



Sliding window…

- As the reasoning process moves forward with time, one older time slice is dropped off the DBN, while another is added.

- This use of a window means that every time we move the window along, the previous evidence received is no longer directly available. Instead, it is summarized taking the current belief for (root) nodes, and making these distributions the new priors

- The DBN updating process is given in the next Algorithm.
  - Note: steps of this DBN updating algorithm are exactly those of a technique used in classical control theory, called a Kalman Filter

DBN updating process

1. **Sliding:** *Move window along.*
2. **Prediction:**

   (a) *We already know* $Bel(X_{t-1}|\mathbf{E}_{\{1,t-1\}})$*, the estimated probability distribution over* X_{t-1}.

   (b) *Calculate the predicted beliefs,* $\widehat{Bel}(X_t|\mathbf{E}_{\{1,t-1\}})$,

3. **Rollup:**

   (a) *Remove time-slice* $t-1$.

   (b) *Use the predictions for the* t *slice as the new prior by setting* $P(X)$ *to* $\widehat{Bel}(X_t|\mathbf{E}_{\{1,t-1\}})$.

4. **Estimation:**

   (a) *Add new observations* $\mathbf{E}_t$.

   (b) *Calculate* $Bel(X_t|\mathbf{E}_{\{1,t\}})$*, the probability distribution over the current state.*

   (c) *Add the slice for* $t+1$.

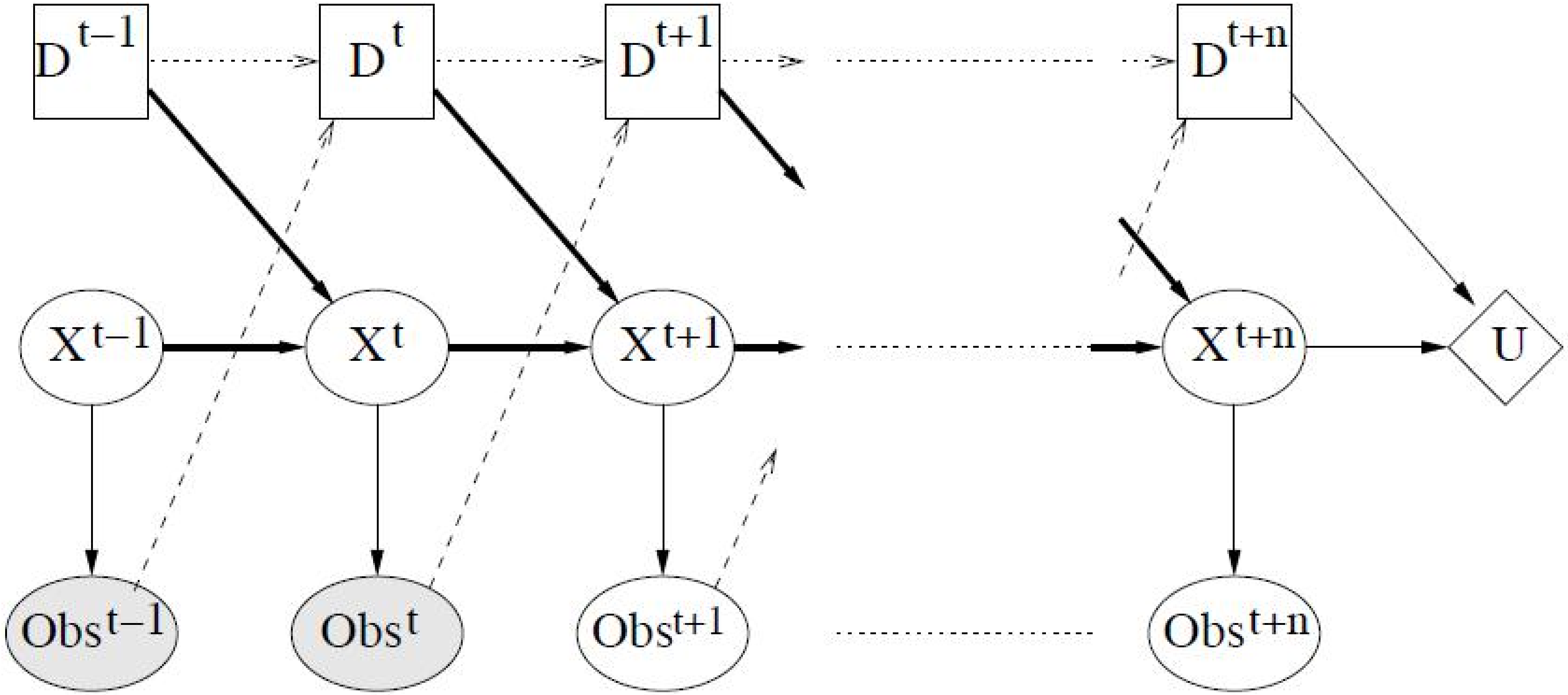
Inference in DBN

- Exact **clustering algorithms** can be applied to DBNs, particularly if the inference is restricted to two time-slices

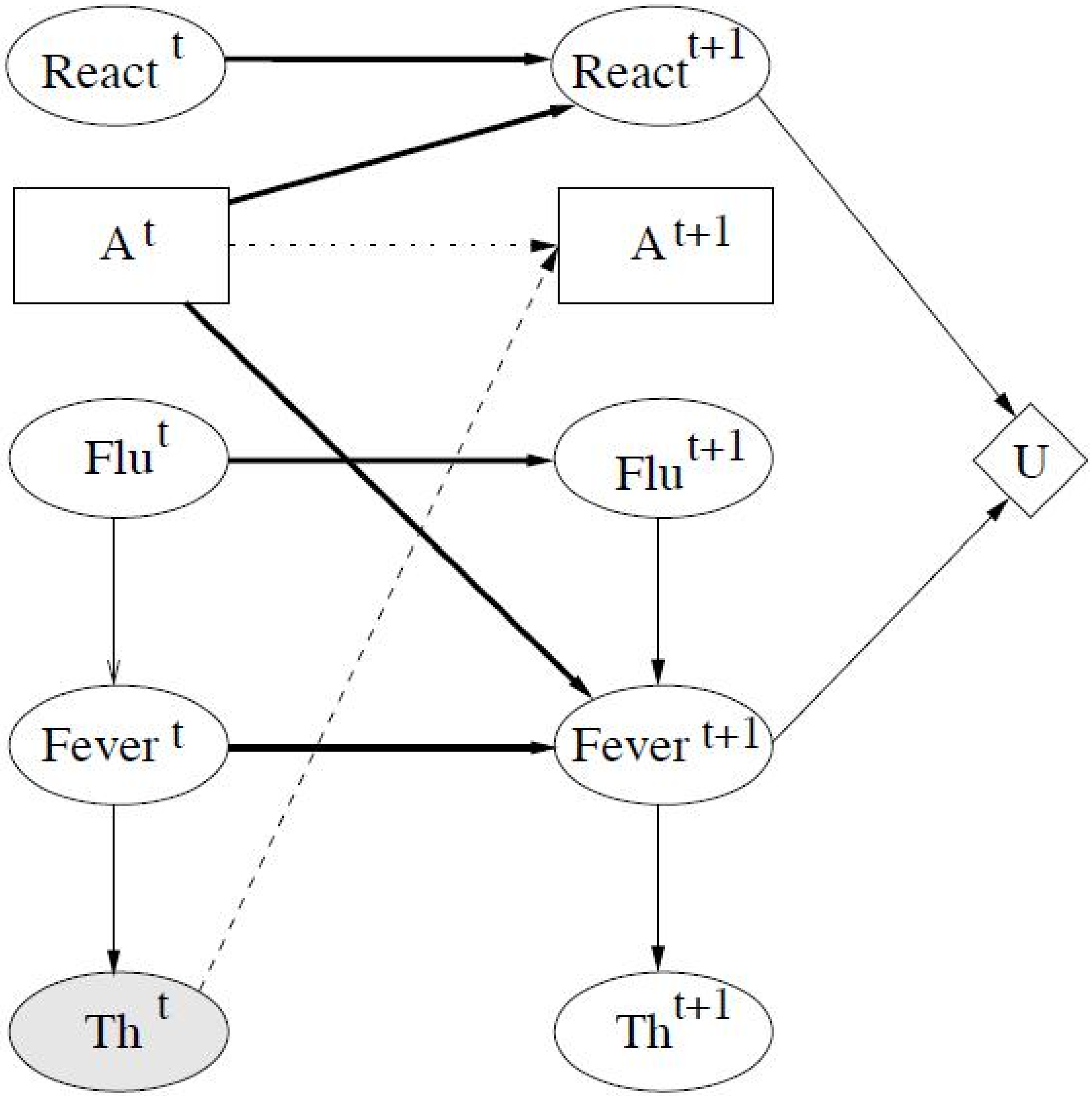
- Unfortunately, it is common that there is a cluster containing all the nodes in a time slice with inter-slice connections, so the clusters become hard computationally

Dynamic decision networks

- Just as Bayesian networks can be extended with a temporal dimension to give DBNs so can decision networks be extended to give dynamic decision networks (DDNs).



A DDN for the Fever problem



Mobile Robot Example

The robot's task is to detect and track a moving object, using sonar and vision sensor information, given a global map of the office floor environment. The robot must also continually reassess its own position (called localization) to avoid getting lost. At any point in time, the robot can make observations of its position with respect to nearby walls and corners and of the target's position with respect to the robot.

- The problem of a mobile robot that does localization and tracking can be modeled with a DDN as follows: The nodes S_T and S_R represent the locations of the target and the robot, respectively
- The decision node is M, representing the robot' s movement actions options
- The nodes O_R and O_T represent the robot' s observations of its own and the target' s location, respectively
- The overall utility is the weighted sum over time of the utility at each step U_t , which is a measure of the distance between the robot and its target.

Mobile Robot Example