Decision Theory: Single & Sequential Decisions. VE for Decision Networks.

CPSC 322 – Decision Theory 2

Textbook §9.2

April 1, 2011

Course Overview

Course Module

Representation

Reasoning Technique

Environment

Problem Type

		Deterministic	Stochastic	
Static	Constraint Satisfaction	*Variables + Constraints* Search Arc Consistency		
Static	Logic	*Logics* Search	*Bayesian Networks* Variable Elimination	Uncertainty
Sequential	Planning	*STRIPS* Search As CSP (using arc consistency)	*Decision Networks* Variable Elimination *Markov Processes* Value Iteration	Decision Theory

Decision theory: acting under uncertainty

Lecture Overview

- Recap: Utility and Expected Utility

- Single-Stage Decision Problems
  - Single-Stage decision networks
  - Variable elimination (VE) for computing the optimal decision

- Sequential Decision Problems
  - General decision networks
  - Time-permitting: Policies
  - Next lecture: variable elimination for
    finding the optimal policy in general decision networks

Utility

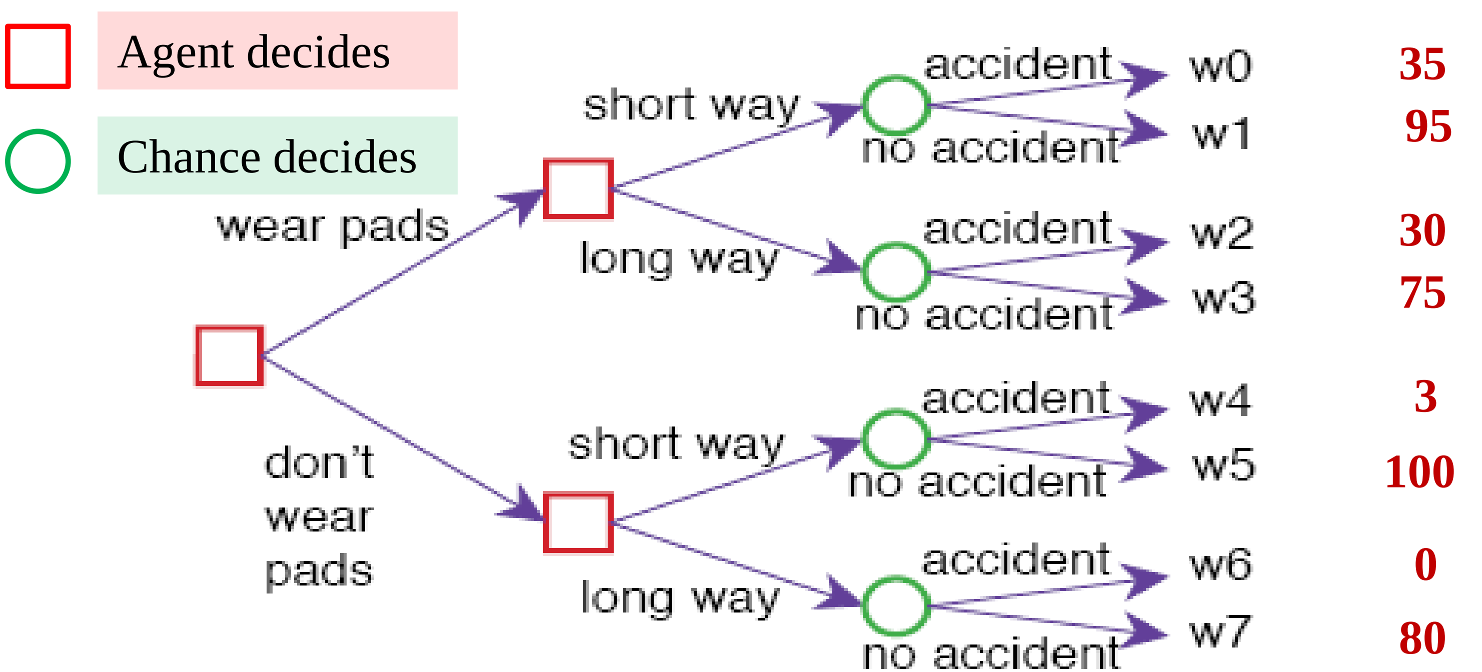
- Utility: a measure of desirability of possible worlds to an agent
  - Let U be a real-valued function such that U(w) represents an agent's degree of preference for world w

- Simple goals can still be specified: e.g.
  - Worlds that satisfy the goal have utility 100
  - Other worlds have utility 0

- Utilities can be more complicated
  - For example, in the robot delivery domains, they could involve
    - Amount of damage
    - Reached the target room?
    - Energy left
    - Time taken

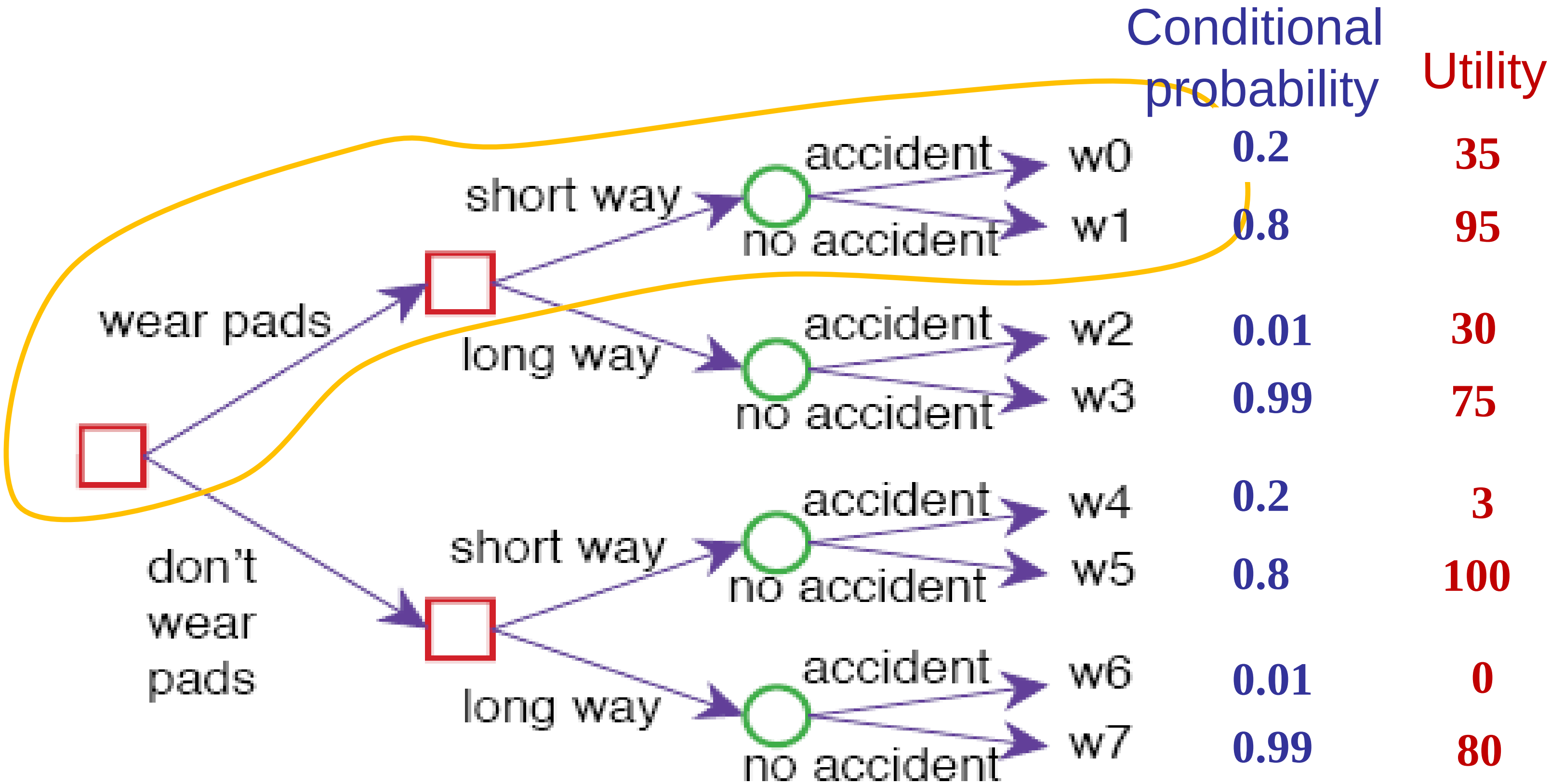
Delivery Robot Example

- Decision variable 1: the robot can choose to wear pads
  - Yes: protection against accidents, but extra weight
  - No: fast, but no protection
- Decision variable 2: the robot can choose the way
  - Short way: quick, but higher chance of accident
  - Long way: safe, but slow
- Random variable: is there an accident?



Possible worlds and decision variables

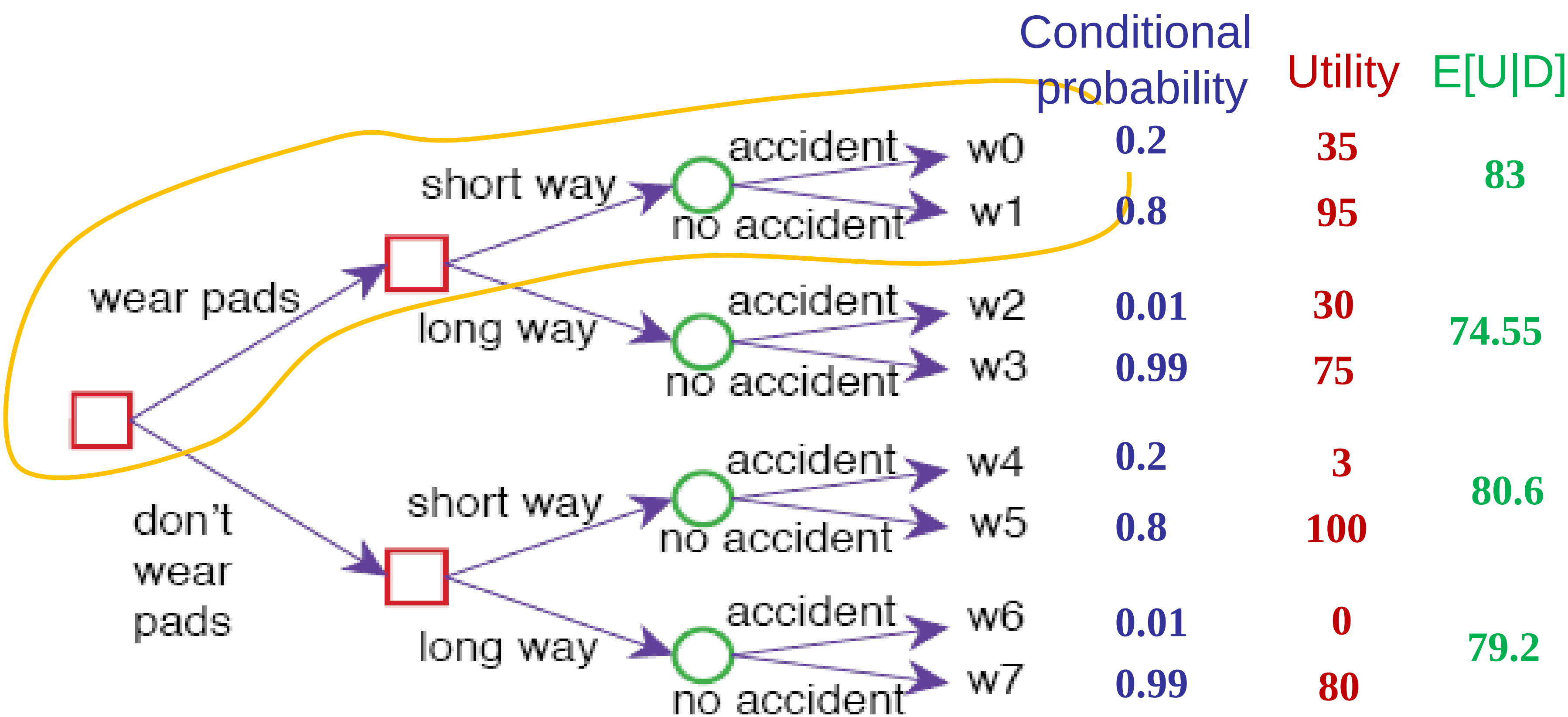
- A possible world specifies a value
  for each random variable and each decision variable
- For each assignment of values to all decision variables
  – the probabilities of the worlds satisfying that assignment sum to 1.



Expected utility of a decision

- The expected utility of a decision is:

$$E[U|D = d] = \sum_{w} P(w|D = d)U(w)$$



Lecture Overview

- Recap: Utility and Expected Utility
- Single-Stage Decision Problems
  - Single-Stage decision networks
  - Variable elimination (VE) for computing the optimal decision
- Sequential Decision Problems
  - General decision networks
  - Time-permitting: Policies
  - Next lecture: variable elimination for finding the optimal policy in general decision networks

Single Action vs. Sequence of Actions

- Single Action (aka One-Off Decisions)
  - One or more primitive decisions that can be treated as a single macro decision to be made before acting
  - E.g., "WearPads" and "WhichWay" can be combined into macro decision (WearPads, WhichWay) with domain {yes,no} × {long, short}

- Sequence of Actions (Sequential Decisions)
  - Repeat:
    - make observations
    - decide on an action
    - carry out the action
  - Agent has to take actions not knowing what the future brings
    - This is fundamentally different from everything we've seen so far
    - Planning was sequential, but we still could still think first and then act

Optimal single-stage decision

- Given a single (macro) decision variable D
  - the agent can choose $D=d_i$ for any value $d_i \in dom(D)$

Definition (optimal single-stage decision)
An optimal single-stage decision is the decision $D=d_{max}$ whose expected value is maximal:

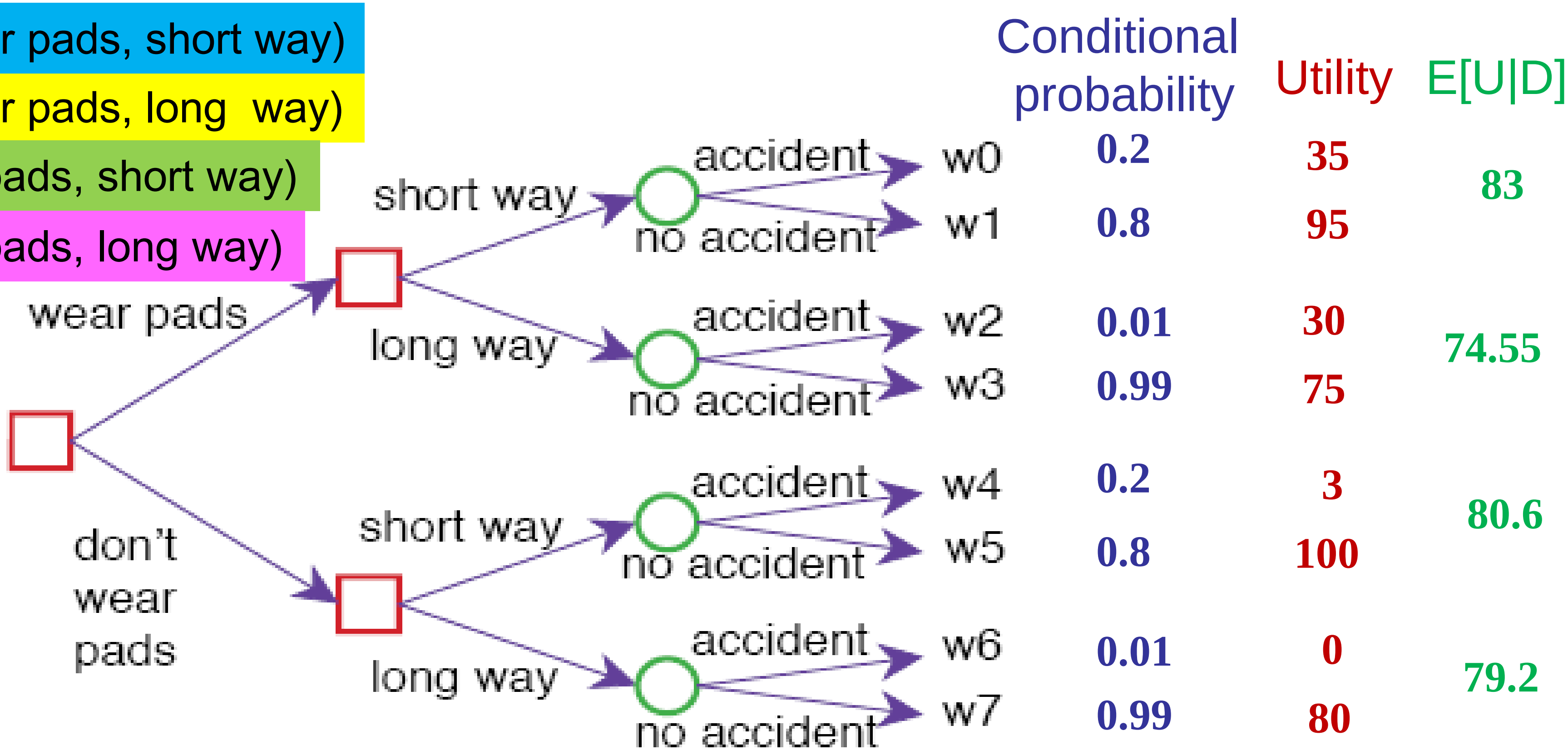
$$d_{max} \in \underset{d_i \in dom(D)}{\operatorname{argmax}} E[U|D=d_i]$$

What is the optimal decision in the example?

Definition (optimal single-stage decision)
An optimal single-stage decision is the decision $D{=}d_{max}$ whose expected value is maximal:

$$d_{max} \in \underset{d_i \in dom(D)}{\operatorname{argmax}} \; E[U|D{=}d_i]$$

(Wear pads, short way)
(Wear pads, long way)
(No pads, short way)
(No pads, long way)

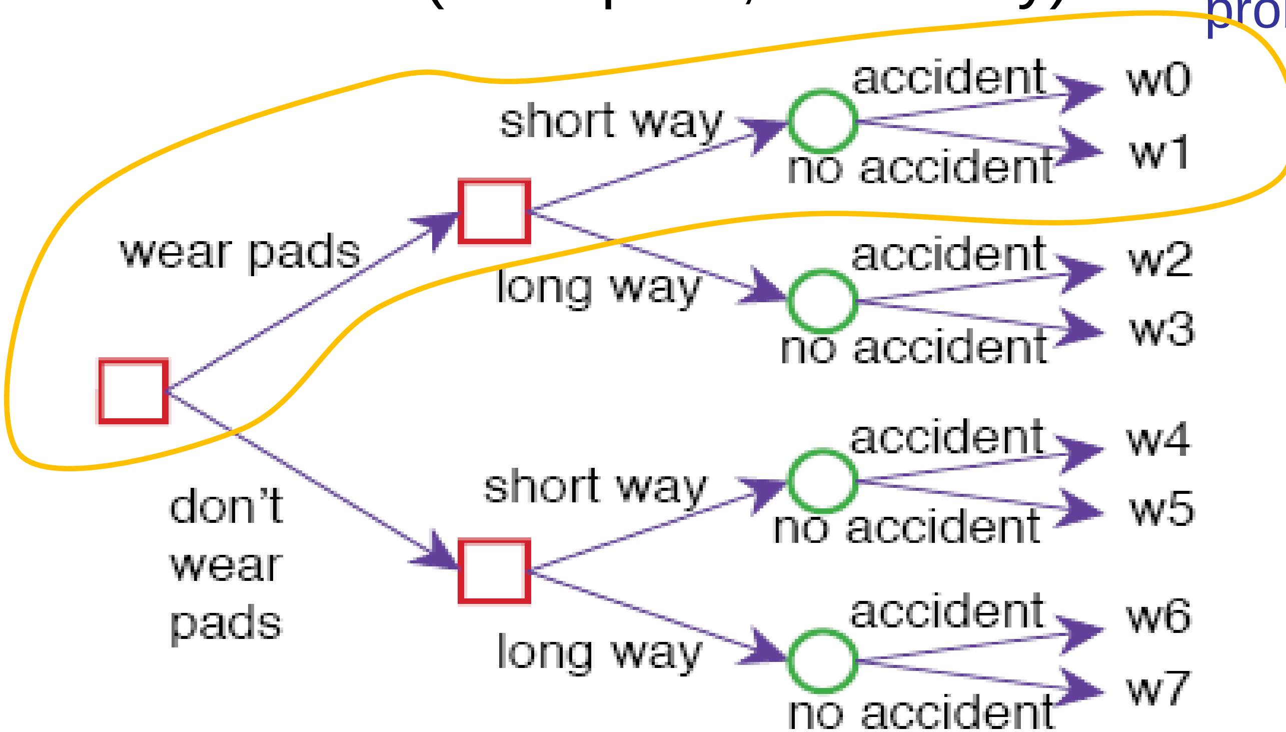


Optimal decision in robot delivery example

Definition (optimal single-stage decision)
An optimal single-stage decision is the decision $D=d_{max}$ whose expected value is maximal:

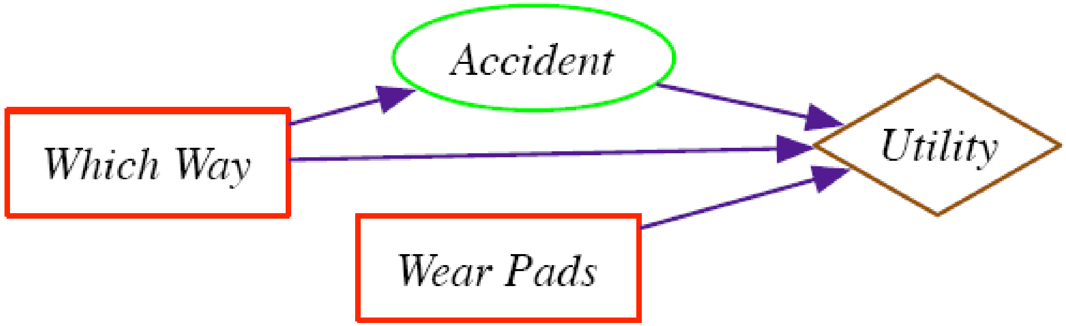
$$d_{max} \in \underset{d_i \in dom(D)}{\operatorname{argmax}} \; E[U|D{=}d_i]$$

Best decision: (wear pads, short way)



Conditional probability	Utility	E[U|D]
0.2	35	83
0.8	95	
0.01	30	74.55
0.99	75	
0.2	3	80.6
0.8	100	
0.01	0	79.2
0.99	80	

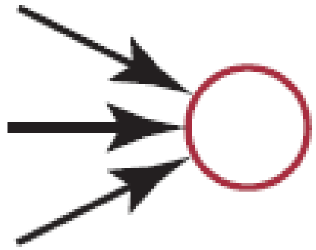
Single-Stage decision networks



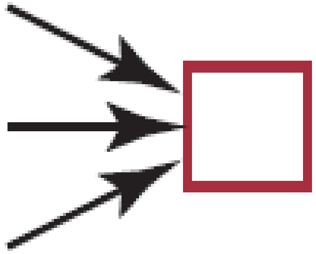
- Extend belief networks with:
  - Decision nodes, that the agent chooses the value for
    - Parents: only other decision nodes allowed
    - Domain is the set of possible actions
    - Drawn as a rectangle
  - Exactly one utility node
    - Parents: all random & decision variables on which the utility depends
    - Does not have a domain
    - Drawn as a diamond

- Explicitly shows dependencies
  - E.g., which variables affect the probability of an accident?

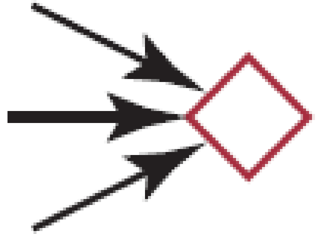
Types of nodes in decision networks



- A random variable is drawn as an ellipse.
  - Arcs into the node represent probabilistic dependence
  - As in Bayesian networks:
    a random variable is conditionally independent of its non-descendants given its parents



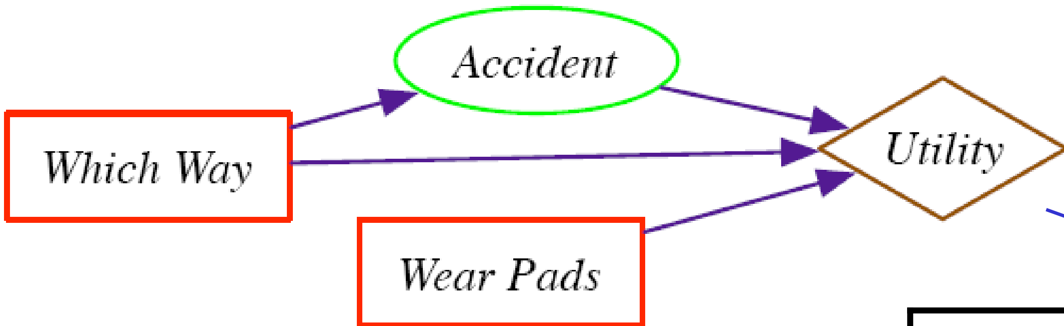
- A decision variable is drawn as an rectangle.
  - Arcs into the node represent
    information available when the decision is made



- A utility node is drawn as a diamond.
  - Arcs into the node represent variables that the utility depends on.
  - Specifies a utility for each instantiation of its parents

Example Decision Network

Which Way W	Accident A	P(A|W)
long	true	0.01
long	false	0.99
short	true	0.2
short	false	0.8

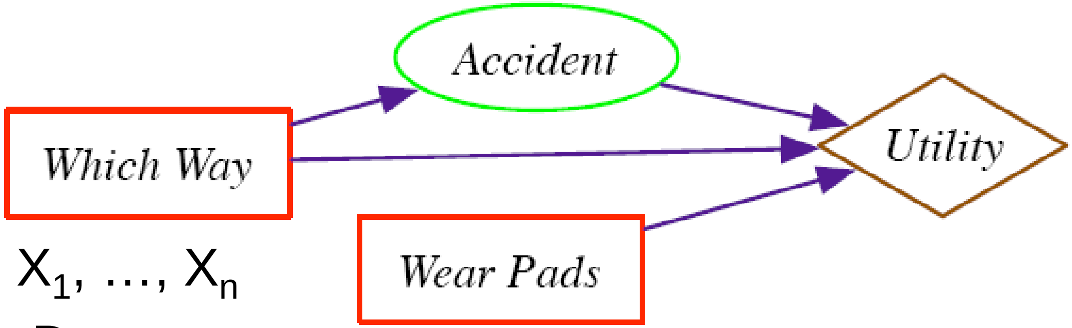


Which way	Accident	Wear Pads	Utility
long	true	true	30
long	true	false	0
long	false	true	75
long	false	false	80
short	true	true	35
short	true	false	3
short	false	true	95
short	false	false	100

Decision nodes do not have an associated table.

The utility node does not have a domain.

Computing the optimal decision: we can use VE



- Denote
  - the random variables as $X_1, \ldots, X_n$
  - the decision variables as D
  - the parents of node N as pa(N)

$$E(U) = \sum_{X_1,\ldots,X_n} P(X_1,\ldots,X_n \mid D)\, U(pa(U))$$

$$= \sum_{X_1,\ldots,X_n} \prod_{i=1}^{n} P(X_i \mid pa(X_i))\, U(pa(U))$$

- To find the optimal decision we can use VE:
  1. Create a factor for each conditional probability and for the utility
  2. Sum out all random variables, one at a time
     - This creates a factor on D that gives the expected utility for each d_i
  3. Choose the d_i with the maximum value in the factor

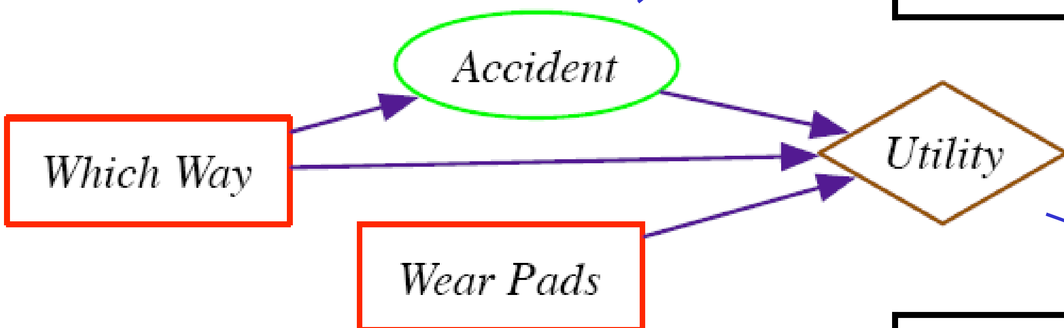
VE Example: Step 1, create initial factors

Abbreviations:

W = Which Way
P = Wear Pads
A = Accident

$f_1(A,W)$

Which Way W	Accident A	P(A|W)
long	true	0.01
long	false	0.99
short	true	0.2
short	false	0.8

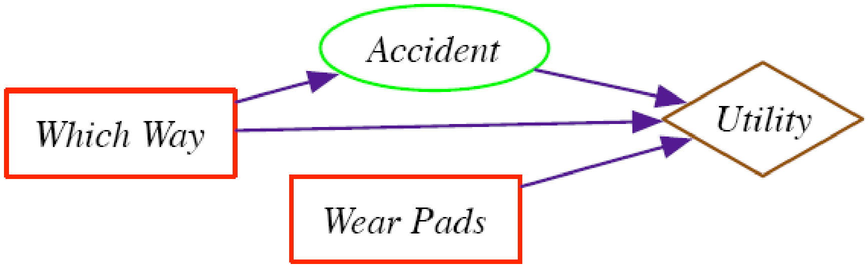


$f_2(A,W,P)$

Which way W	Accident A	Pads P	Utility
long	true	true	30
long	true	false	0
long	false	true	75
long	false	false	80
short	true	true	35
short	true	false	3
short	false	true	95
short	false	false	100

$$E(U) = \sum_A P(A|W)\,U(A,W,P)$$
$$= \sum_A f_1(A,W)\ f_2(A,W,P)$$

VE example: step 2, sum out A



Step 2a: compute product
$f_1(A,W) \times f_2(A,W,P)$

What is the right form for the product $f_1(A,W) \times f_2(A,W,P)$?

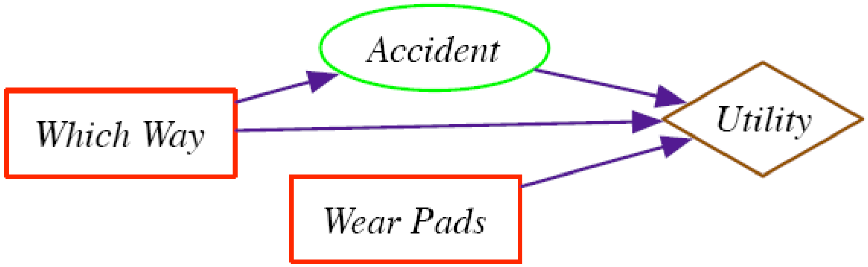
f (A,W)

f(A,P)

f(A)

f(A,P,W)

VE example: step 2, sum out A

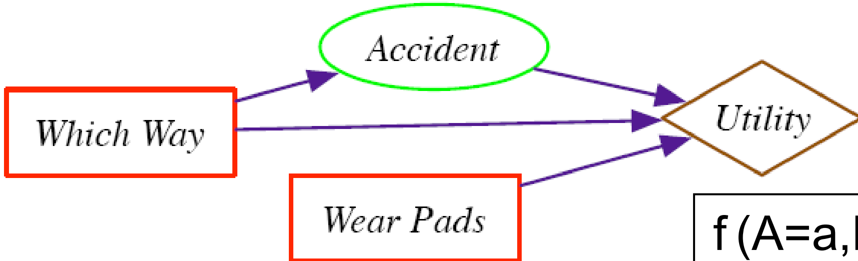


Step 2a: compute product
$f(A,W,P) = f_1(A,W) \times f_2(A,W,P)$

What is the right form for the product $f_1(A,W) \times f_2(A,W,P)$?

- It is $f(A,P,W)$:
  the domain of the product is the union of the multiplicands' domains
- $f(A,P,W) = f_1(A,W) \times f_2(A,W,P)$
  - I.e., $f(A=a,P=p,W=w) = f_1(A=a,W=w) \times f_2(A=a,W=w,P=p)$

VE example: step 2, sum out A



Step 2a: compute product

$f(A,W,P) = f_1(A,W) \times f_2(A,W,P)$

$f(A=a,P=p,W=w) = f_1(A=a,W=w) \times f_2(A=a,W=w,P=p)$

Which way W	Accident A	$f_1(A,W)$
long	true	0.01
long	false	0.99
short	true	0.2
short	false	0.8

Which way W	Accident A	Pads P	$f_2(A,W,P)$
long	true	true	30
long	true	false	0
long	false	true	75
long	false	false	80
short	true	true	35
short	true	false	3
short	false	true	95
short	false	false	100

Which way W	Accident A	Pads P	f(A,W,P)
long	true	true	0.01 * 30
long	true	false	
long	false	true	
long	false	false	???
short	true	true	
short	true	false	
short	false	true	
short	false	false	

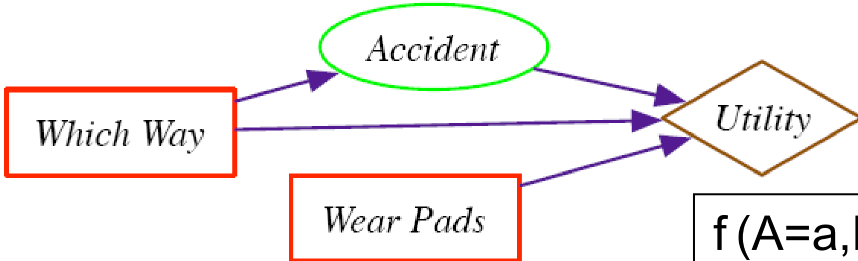
0.99 * 30

0.01 * 80

0.99 * 80

0.8 * 30

VE example: step 2, sum out A



Step 2a: compute product

$f(A,W,P) = f_1(A,W) \times f_2(A,W,P)$

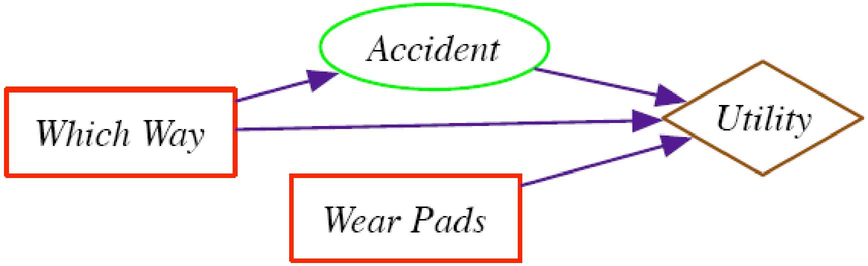
$f(A=a,P=p,W=w) = f_1(A=a,W=w) \times f_2(A=a,W=w,P=p)$

Which way W	Accident A	$f_1(A,W)$
long	true	0.01
long	false	0.99
short	true	0.2
short	false	0.8

Which way W	Accident A	Pads P	$f_2(A,W,P)$
long	true	true	30
long	true	false	0
long	false	true	75
long	false	false	80
short	true	true	35
short	true	false	3
short	false	true	95
short	false	false	100

Which way W	Accident A	Pads P	f(A,W,P)
long	true	true	0.01 * 30
long	true	false	0.01*0
long	false	true	0.99*75
long	false	false	0.99*80
short	true	true	0.2*35
short	true	false	0.2*3
short	false	true	0.8*95
short	false	false	0.8*100

VE example: step 2, sum out A



Step 2b: sum A out of the product f(A,W,P):

$$f_3(W,P) = \sum_A f(A,W,P)$$

Which way W	Pads P	f_3(W,P)
long	true	0.01*30+0.99*75=74.55
long	false	
short	true	??
short	false	

Which way W	Accident A	Pads P	f(A,W,P)
long	true	true	0.01 * 30
long	true	false	0.01*0
long	false	true	0.99*75
long	false	false	0.99*80
short	true	true	0.2*35
short	true	false	0.2*3
short	false	true	0.8*95
short	false	false	0.8*100

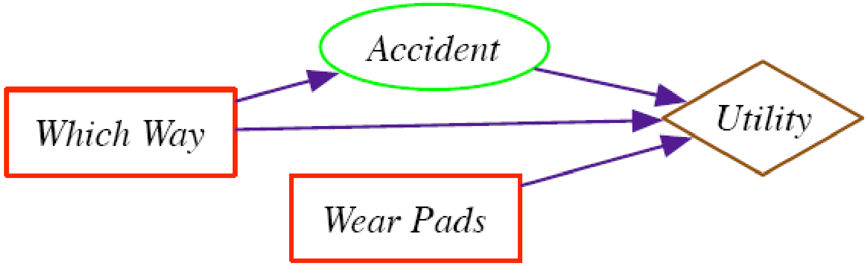
0.2*35 + 0.2*0.3

0.2*35 + 0.8*95

0.99*80 + 0.8*95

0.8 * 95 + 0.8*100

VE example: step 2, sum out A



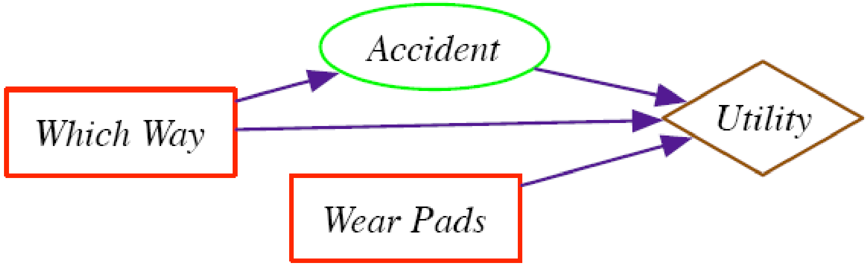
Step 2b: sum A out of the product f(A,W,P):

$$f_3(W,P) = \sum_A f(A,W,P)$$

Which way W	Pads P	f_3(W,P)
long	true	0.01*30+0.99*75=74.55
long	false	0.01*0+0.99*80=79.2
short	true	0.2*35+0.8*95=83
short	false	0.2*3+0.8*100=80.6

Which way W	Accident A	Pads P	f(A,W,P)
long	true	true	0.01 * 30
long	true	false	0.01*0
long	false	true	0.99*75
long	false	false	0.99*80
short	true	true	0.2*35
short	true	false	0.2*3
short	false	true	0.8*95
short	false	false	0.8*100

VE example: step 3, choose decision with max E(U)



Step 2b: sum A out of the product f(A,W,P):

$$f_3(W,P) = \sum_A f(A,W,P)$$

Which way W	Pads P	f_3(W,P)
long	true	0.01*30+0.99*75=74.55
long	false	0.01*0+0.99*80=79.2
short	true	0.2*35+0.8*95=83
short	false	0.2*3+0.8*100=80.6

Which way W	Accident A	Pads P	f(A,W,P)
long	true	true	0.01 * 30
long	true	false	0.01*0
long	false	true	0.99*75
long	false	false	0.99*80
short	true	true	0.2*35
short	true	false	0.2*3
short	false	true	0.8*95
short	false	false	0.8*100

The final factor encodes the expected utility of each decision

- Thus, taking the short way but wearing pads is the best choice, with an expected utility of 83

Variable Elimination for Single-Stage Decision Networks: Summary

1. Create a factor for each conditional probability
   and for the utility
2. Sum out all random variables, one at a time
   – This creates a factor on D that gives the expected utility for each d_i
3. Choose the d_i with the maximum value in the factor

Lecture Overview

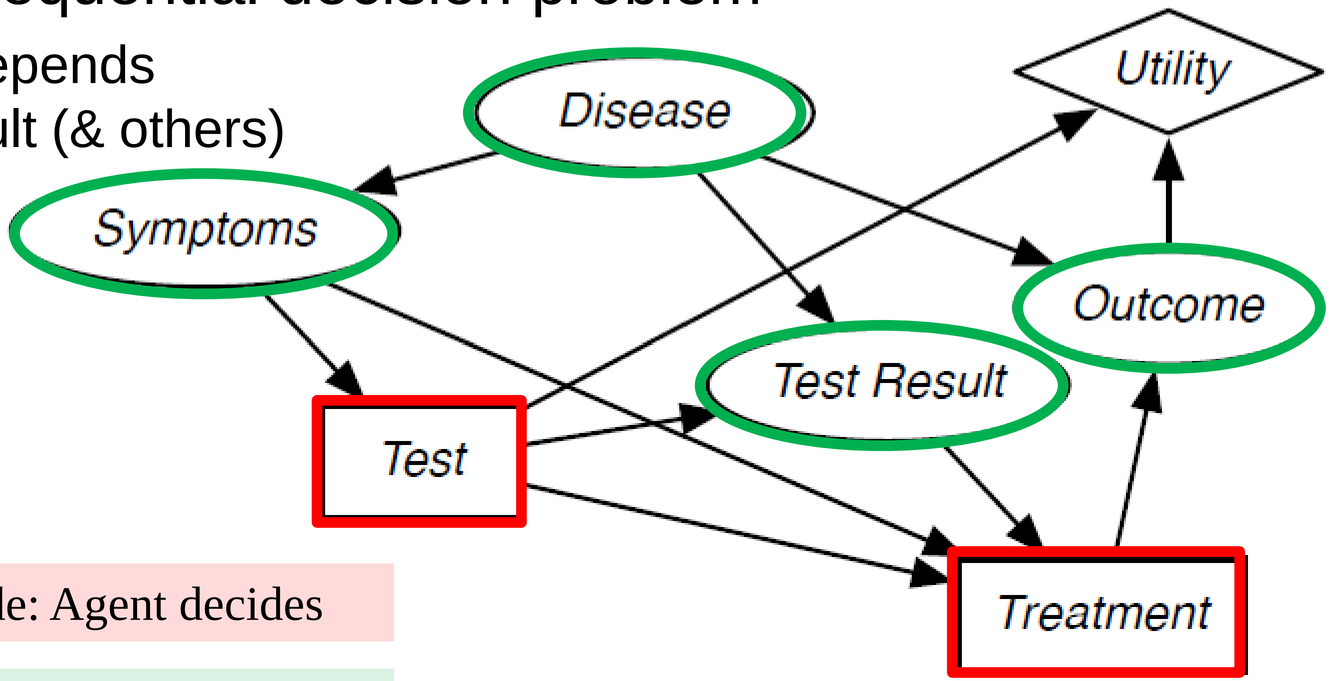
- Recap: Utility and Expected Utility
- Single-Stage Decision Problems
  - Single-Stage decision networks
  - Variable elimination (VE) for computing the optimal decision
- Sequential Decision Problems
  - General decision networks and Policies
  - Next lecture: variable elimination for finding the optimal policy in general decision networks

Sequential Decision Problems

- An intelligent agent doesn't make a multi-step decision and carry it out blindly
  - It would take new observations it makes into account
- A more typical scenario:
  - The agent observes, acts, observes, acts, …
- Subsequent actions can depend on what is observed
  - What is observed often depends on previous actions
  - Often the sole reason for carrying out an action is to provide information for future actions
    - For example: diagnostic tests, spying
- General Decision networks:
  - Just like single-stage decision networks, with one exception: the parents of decision nodes can include random variables

Sequential Decision Problems: Example

- Example for sequential decision problem
  - Treatment depends on Test Result (& others)



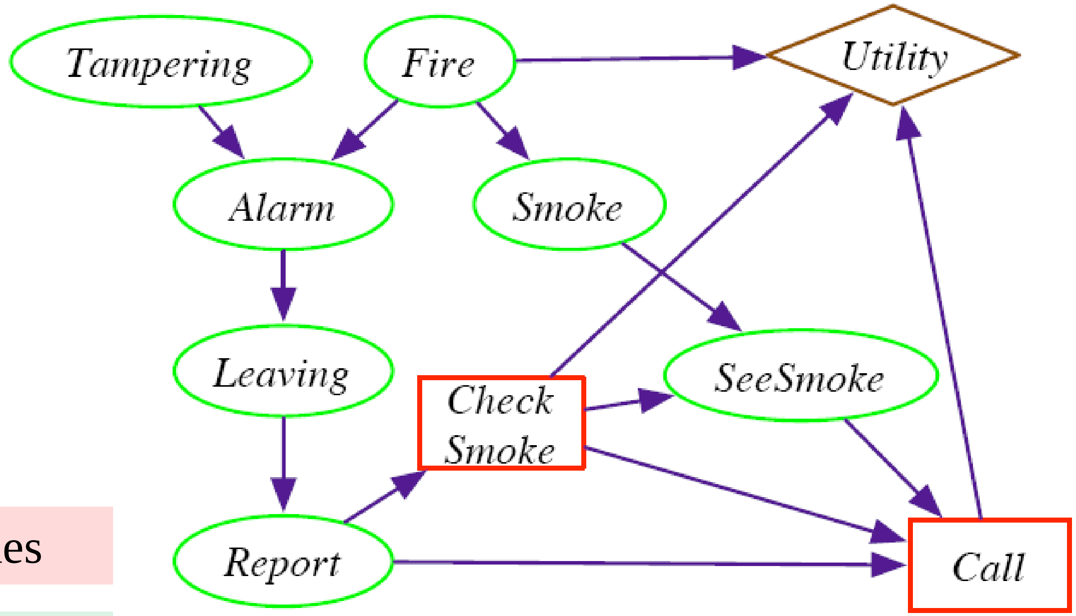
Decision node: Agent decides

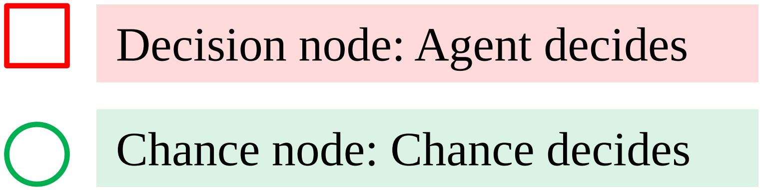
Chance node: Chance decides

- Each decision D_i has an information set of variables $pa(D_i)$, whose value will be known at the time decision D_i is made
  - pa(Test) = {Symptoms}
  - pa(Treatment) = {Test, Symptoms, TestResult}

Sequential Decision Problems: Example

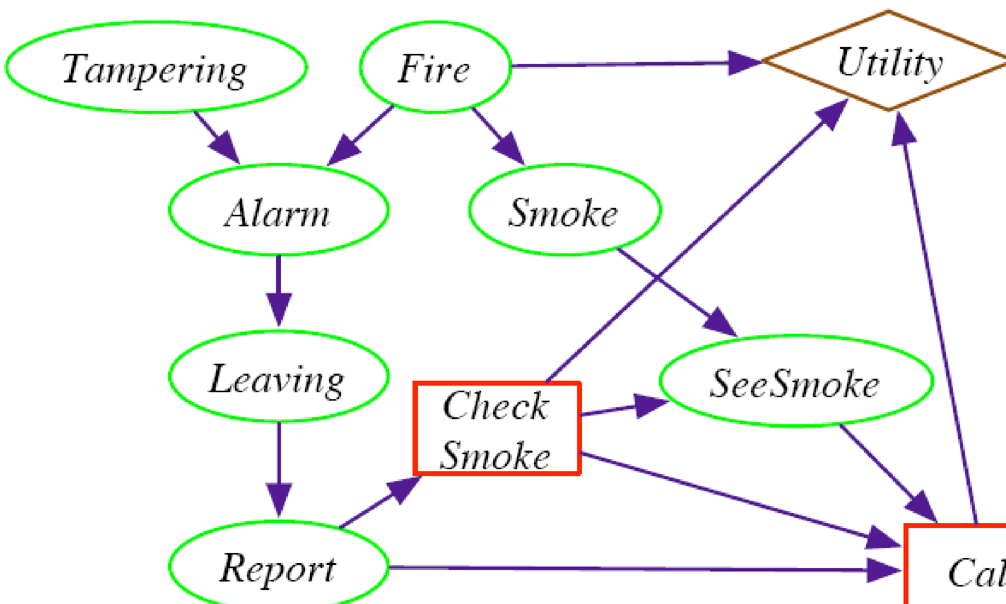
- Another example for sequential decision problems
  - Call depends on Report and SeeSmoke (and on CheckSmoke)





Sequential Decision Problems

- What should an agent do?
  - What an agent should do depends on what it will do in the future
    - E.g. agent only needs to check for smoke if that will affect whether it calls
  - What an agent does in the future depends on what it did before
    - E.g. when making the decision it needs to whether it checked for smoke
  - We will get around this problem as follows
    - The agent has a conditional plan of what it will do in the future
    - We will formalize this conditional plan as a policy



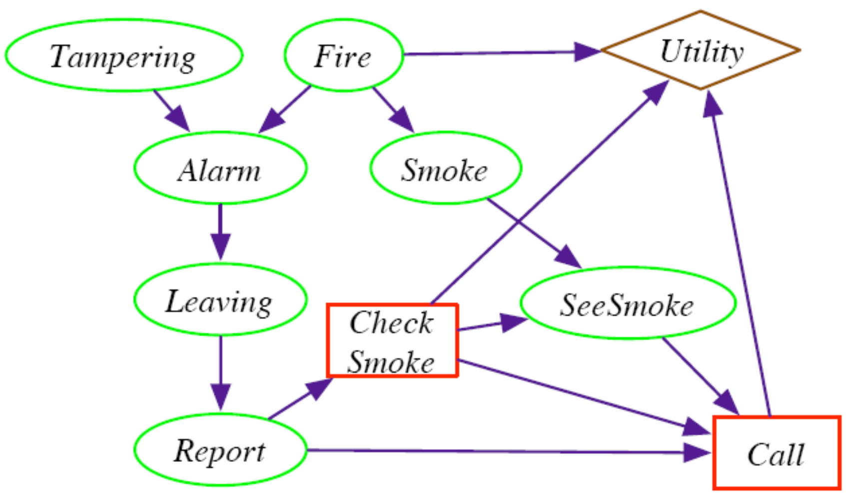
Policies for Sequential Decision Problems

Definition (Policy)

A policy is a sequence of $\delta_1, \ldots, \delta_n$ decision functions

$$\delta_i : \text{dom}(pa(D_i)) \rightarrow \text{dom}(D_i)$$

This policy means that when the agent has observed $o \in \text{dom}(pD_i)$, it will do $\delta_i(o)$



There are $2^2=4$ possible decision functions δ_{cs} for Check Smoke:

- Decision function needs to specify a value for each instantiation of parents

CheckSmoke

Report	$\delta_{cs}1$	$\delta_{cs}2$	$\delta_{cs}3$	$\delta_{cs}4$
T	T	T	F	F
F	T	F	T	F

Call

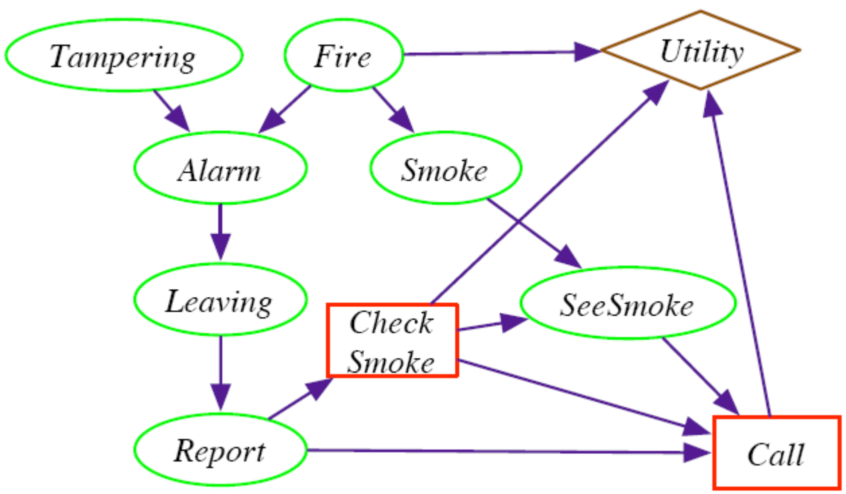
Policies for Sequential Decision Problems

Definition (Policy)
A policy is a sequence of $\delta_1, \ldots, \delta_n$ decision functions

$$\delta_i : \mathrm{dom}(pa(D_i)) \rightarrow \mathrm{dom}(D_i)$$

There are 2^8=256 possible decision functions δ_{cs} for Call:

- Decision function needs to specify a value for each instantiation of parents



Call

Report	CheckS	SeeS	$\delta_{call}1$		$\delta_{call}n$
true	true	true	true		false
true	true	false	true		false
true	false	true	true		false
true	false	false	true		false
false	true	true	true		false
false	true	false	true	......	false
false	false	true	true		false
false	false	false	true		false

How many policies are there?

- If a decision D has k binary parents, how many assignments of values to the parents are there?

$2k$ $2+k$ k^2 2^k

How many policies are there?

- If a decision D has k binary parents, how many assignments of values to the parents are there?
  - 2^k

- If there are b possible value for a decision variable, how many different decision functions are there for it if it has k binary parents?

2^{kp} | $b*2^k$ | b^{2^k} | 2^{k^b}

Learning Goals For Today's Class

- Compare and contrast stochastic single-stage (one-off) decisions vs. multistage decisions
- Define a Utility Function on possible worlds
- Define and compute optimal one-off decisions
- Represent one-off decisions as single stage decision networks
- Compute optimal decisions by Variable Elimination

- Next time:
  – Variable Elimination for finding optimal policies

Announcements

- Assignment 4 is due on Monday
- Final exam is on Monday, April 11
  - The list of short questions is online … please use it!
- Office hours next week
  - Simona: Tuesday, 10-12 (no office hours on Monday!)
  - Mike: Wednesday 1-2pm, Friday 10-12am
  - Vasanth: Thursday, 3-5pm
  - Frank:
    - X530: Tue 5-6pm, Thu 11-12am
    - DMP 110: 1 hour after each lecture

- Optional Rainbow Robot tournament: Friday, April 8
  - Hopefully in normal classroom (DMP 110)
  - Vasanth will run the tournament,
    I'll do office hours in the same room (this is 3 days before the final)