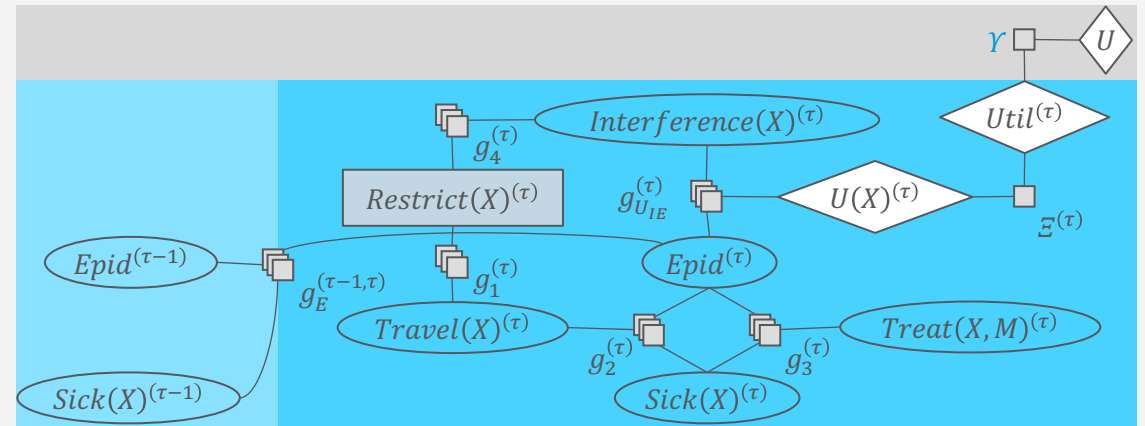


Decision-theoretic Models and Inference

Statistical Relational Artificial Intelligence
(StaRAI)



Contents

1. Introduction

- Notions of AI, agents, StaRAI

2. Modelling Formalisms

- a. *Logic*
- b. *Probabilistic Graphical Models (PGMs)*
- c. *Probabilistic Relational Models (PRMs)*
- Syntax, semantics, inference tasks

3. Inference

- a. *Exact Inference*
 - i. *(Weighted) Model Counting*
 - ii. *Variable Elimination*
 - iii. *Lifted Inference*

b. *Approximate Inference*

- Inference algorithms

4. Learning

- Relation learning in logic
- Parameter / structure learning for PGMs
- From PGMs to PRMs, PRM learning

5. Sequential Models and Inference

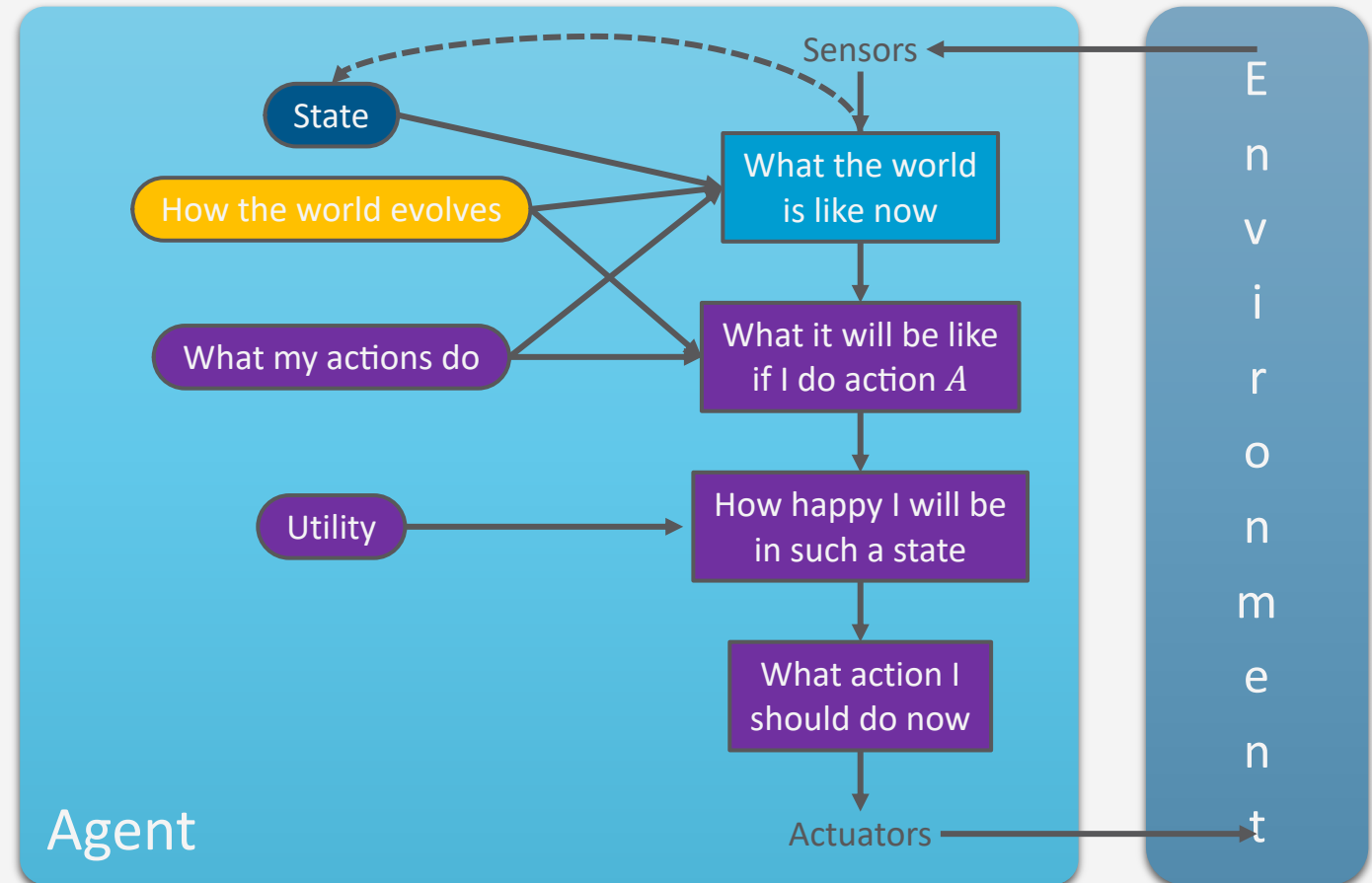
- Syntax, semantics, inference tasks, algorithms

6. Decision-theoretic Models and Inference

- Preferences, utility
- Syntax, semantics, inference task, algorithm

Contents in this Lecture Related to *Utility-based Agents*

- Further topics
 2. Modelling formalisms
 3. Inference (in episodic models)
 4. Learning (of episodic models)
 5. Sequential models and inference
 6. Decision-theoretic models and inference



Setting

- Agent can perform actions in an environment
 - Episodic, i.e., not sequential, environment
 - Next episode does not depend on the previous episode
 - Or sequential environment
 - Non-deterministic environment
 - Outcomes of actions not unique
 - Associated with probabilities
→ probabilistic model
 - Partially observable
 - Latent, i.e., not observable, random variables
- Agent has preferences over states / action outcomes
 - Encoded in utility or utility function → Utility theory
 - “Decision theory”
= Utility theory + Probability theory
 - Model the world with a probabilistic model
 - Model preferences with a utility (function)
 - Find action that leads to the maximum expected utility, also called decision making

Outline: 6. Decision-theoretic Models and Inference

a. Utility Theory

- Preferences, maximum expected utility (MEU) principle
- Utility function, multi-attribute utility theory

b. Static Decision Making

- Modelling, semantics, inference tasks
- Inference algorithm: LVE for MEU as an example

c. Sequential Decision Making

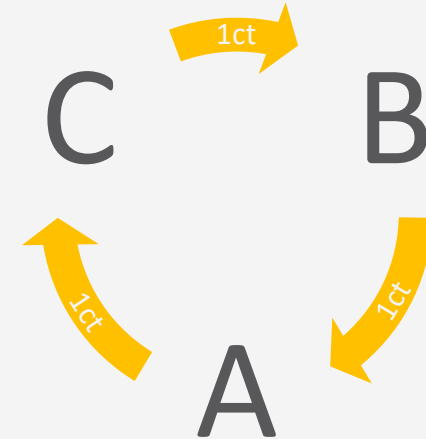
- Modelling, semantics, sequential MEU problem
- Inference algorithm: LDJT for MEU as an example
- Acting

Preferences

- An agent chooses among prizes (A , B , etc.) and lotteries, i.e., situations with uncertain prizes
 - Outcome of a nondeterministic action is a lottery
- Lottery $L = [p, A; (1 - p), B]$
 - A and B can be lotteries again
 - Prizes are special lotteries: $[1, R; 0, \text{not } R]$
 - More than two outcomes:
 - $L = [p_1, S_1; p_2, S_2; \dots; p_M, S_M], \sum_{i=1}^M p_i = 1$
- Notation
 - $A \succ B$ A preferred to B
 - $A \sim B$ indifference between A and B
 - $A \succeq B$ B not preferred to A

Rational Preferences

- Idea: preferences of a rational agent must obey constraints
 - As prerequisite for reasonable preference relations
- Rational preferences $\rightarrow$ behaviour describable as maximisation of expected utility
- Violating constraints leads to self-evident irrationality
 - Example
 - An agent with intransitive preferences can be induced to give away all its money
 - If $B \succ C$, then an agent who has C would pay (say) 1 cent to get B
 - If $A \succ B$, then an agent who has B would pay (say) 1 cent to get A
 - If $C \succ A$, then an agent who has A would pay (say) 1 cent to get C



Axioms of Utility Theory

1. Orderability

- $(A \succ B) \vee (A \prec B) \vee (A \sim B)$
 - $\{\prec, \succ, \sim\}$ jointly exhaustive, pairwise disjoint

2. Transitivity

- $(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$

3. Continuity

- $A \succ B \succ C \Rightarrow \exists p [p, A; 1 - p, C] \sim B$

4. Substitutability

- $A \sim B \Rightarrow [p, A; 1 - p, C] \sim [p, B; 1 - p, C]$
 - Also holds if replacing $\sim$ with $\succ$

5. Monotonicity

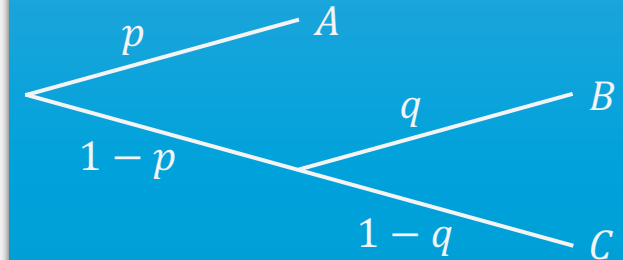
- $A \succ B \Rightarrow (p \geq q \Leftrightarrow [p, A; 1 - p, B] \succeq [q, A; 1 - q, B])$

6. Decomposability

- $[p, A; 1 - p, [q, B; 1 - q, C]] \sim [p, A; (1 - p)q, B; (1 - p)(1 - q), C]$

Decomposability:
There is no fun in gambling.

Equivalent lotteries:



And Then There Was Utility

- Theorem (Ramsey, 1931; von Neumann and Morgenstern, 1944):
 - Given preferences satisfying the constraints, there exists a real-valued function U such that
$$U(A) \geq U(B) \Leftrightarrow A \succeq B$$

- Existence of a utility function
- Expected utility of a lottery:

$$U([p_1, S_1; \dots; p_M, S_M]) = \sum_{i=1}^M p_i U(S_i)$$

- MEU principle
 - Choose the action that maximises expected utility

Utilities

- Utilities map states to real numbers.
Which numbers?
- Standard approach to assessment of human utilities:
 - Compare a given state A to a standard lottery L_p that has
 - “best possible outcome” T with probability p
 - “worst possible catastrophe” $\perp$ with probability $(1 - p)$
 - Adjust lottery probability p until $A \sim L_p$



Utility Scales

- **Normalised** utilities: $u_{\top} = 1.0, u_{\perp} = 0.0$
 - Utility of lottery $L \sim (\text{pay-}\$30\text{-and-continue-as-before})$: $U(L) = u_{\top} \cdot 0.9999999 + u_{\perp} \cdot 0.0000001 = 0.9999999$
- **Micromorts**: one-millionth chance of death
 - Useful for Russian roulette, paying to reduce product risks, etc.
 - Example for low risk
 - Drive a car for 370km $\approx$ 1 micromort $\rightarrow$ lifespan of a car: 150,000km $\approx$ 400 micromorts
 - Studies showed that many people appear to be willing to pay US\$10,000 for a safer car that halves the risk of death $\rightarrow$ US\$50/micromort
- **QALYs**: quality-adjusted life years
 - Useful for medical decisions involving substantial risk
- In planning: task becomes minimisation of **cost** instead of maximisation of utility

Utility Scales

- Behaviour is **invariant** w.r.t. positive linear transformation
$$U'(r) = k_1 U(r) + k_2$$
 - No unique utility function; $U'(r)$ and $U(r)$ yield same behaviour
- With deterministic prizes only (no lottery choices), only **ordinal** utility can be determined, i.e., total order on prizes
 - Ordinal utility function also called **value function**
 - Provides a ranking of alternatives (states), but not a meaningful metric scale (numbers do not matter)
- Note:

An agent can be entirely rational (consistent with MEU) without ever representing or manipulating utilities and probabilities

 - E.g., a lookup table for perfect tic-tac-toe

Multi-attribute Utility Theory

- A given state may have multiple utilities
 - ...because of multiple evaluation criteria
 - ...because of multiple agents (interested parties) with different utility functions
- There are:
 - Cases in which decisions can be made *without* combining the attribute values into a single utility value
 - **Strict dominance**
 - Not this lecture
 - Cases in which the utilities of attribute combinations can be specified very concisely
 - This lecture!

Preference Structure

- To specify the complete utility function $U(r_1, \dots, r_M)$, we need d^M values in the worst case
 - M attributes
 - each attribute with d distinct possible values
 - Worst case meaning: Agent's preferences have no regularity at all
- Supposition in multi-attribute utility theory
 - Preferences of typical agents have much more structure
- Approach
 - Identify regularities in the preference behaviour
 - Use so-called **representation theorems** to show that an agent with a certain kind of preference structure has a utility function
$$U(r_1, \dots, r_M) = \mathcal{E}[f_1(r_1), \dots, f_M(r_M)]$$
 - where $\mathcal{E}$ is hopefully a simple function such as *addition*

Preference Independence

- R_1 and R_2 **preferentially independent** (PI) of R_3 iff
 - Preference between $\langle r_1, r_2, r_3 \rangle$ and $\langle r'_1, r'_2, r_3 \rangle$ does not depend on r_3
 - E.g., $\langle \text{Noise}, \text{Cost}, \text{Safety} \rangle$
 - $\langle 20,000 \text{ suffer}, \$4.6 \text{ billion}, 0.06 \text{ deaths/month} \rangle$
 - $\langle 70,000 \text{ suffer}, \$4.2 \text{ billion}, 0.06 \text{ deaths/month} \rangle$
- Theorem (Leontief, 1947)
 - If every pair of attributes is PI of its complement, then every subset of attributes is PI of its complement
 - Called **mutual PI (MPI)**

Preference Independence

- Theorem (Debreu, 1960):
 - MPI $\Rightarrow \exists$ *additive value function*

$$V(r_1, \dots, r_M) = \sum_{i=1}^M V_i(r_i)$$

- Hence assess M single-attribute functions
 - Decomposition of V into a set of summands (additive semantics)
similar to
 - Decomposition of P_R into a set of factors (multiplicative semantics)
- Often a good approximation
- Example:

$$V(\text{Noise}, \text{Cost}, \text{Deaths}) = -\text{Noise} \cdot 10^4 - \text{Cost} - \text{Deaths} \cdot 10^{12}$$

Interim Summary

- Preferences
 - Preferences of a rational agent must obey constraints
- Utilities
 - Rational preferences = describable as maximisation of expected utility
 - Utility axioms
 - MEU principle
- Multi-attribute utility theory
 - Preference structure
 - (Mutual) preferential independence

Outline: 6. Decision-theoretic Models and Inference

a. Utility Theory

- Preferences, maximum expected utility (MEU) principle
- Utility function, multi-attribute utility theory

b. Static Decision Making

- Modelling, semantics, inference tasks
- Inference algorithm: LVE for MEU as an example

c. Sequential Decision Making

- Modelling, semantics, sequential MEU problem
- Inference algorithm: LDJT for MEU as an example
- Acting

Decision Networks/Models

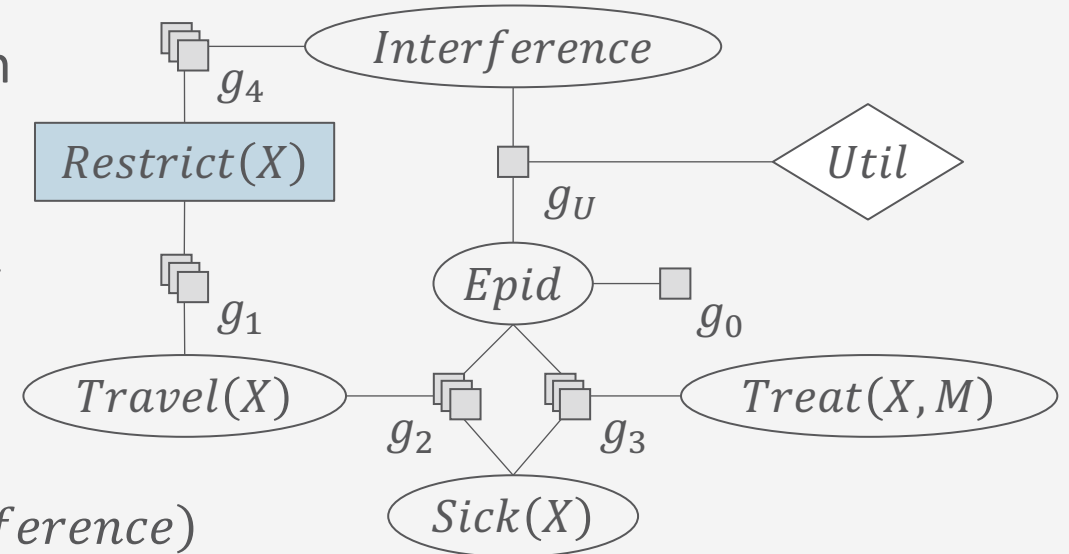
- Extend a PGM to handle actions and utilities
 - Decision variables
 - Utility variables
- Also called influence diagrams
- Given a decision model, use an inference method of one's choosing to find actions that lead to the highest expected utility
- Also allows to perform so-called *Value of Information* calculations
 - Is it worth it to spend resources on getting more information (in the form of evidence)?

Decision PRVs

• Decision PRV D

- Range $\text{ran}(D) = \{d_i\}_{i=1}^K$ set of possible actions
 - Actions d_i mutually exclusive (consistent with range definition)
 - Always have to get a value assigned
 - Cannot *not* make a decision!
- Depicted by a rectangle in a graphical representation
- E.g., travel restrictions for people X : $\text{Restrict}(X)$
 - Range values: ban, free
- Set of decision PRVs D in a model, i.e., $R = D \cup V$
 - D can occur as arguments to any parfactor
 - Example:
 - $\phi_1(\text{Restrict}(X), \text{Travel}(X)), \phi_4(\text{Restrict}(X), \text{Interference})$

$R(X)$	I	ϕ_4	$R(X)$	$Tl(X)$	ϕ_1
free	false	1	free	false	1
free	true	0	free	true	1
ban	false	0	ban	false	1
ban	true	1	ban	true	0



Utility PRVs & Utility Parfactors

- Utility PRV U

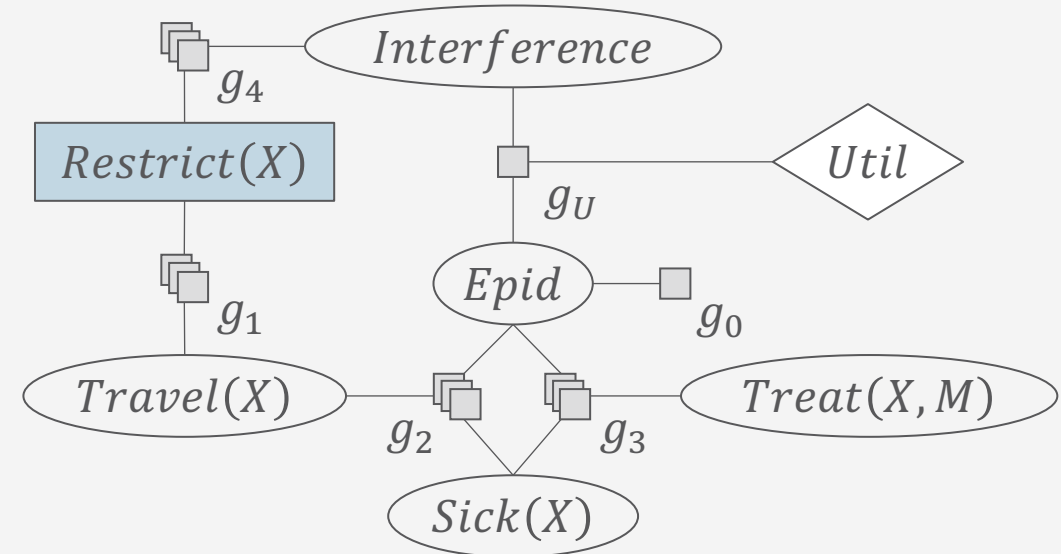
- Range $\text{ran}(U) = \mathbb{R}$
- Output variable, i.e., gets assigned a value by utility function
- Depicted by a diamond in a graphical representation

- Utility parfactor $\phi_U(\mathcal{A})|_C$

- Arguments $\mathcal{A}$ a sequence of (decision) PRVs
- U a utility PRV
- Function $\phi_U: \times_{i=1}^l \text{ran}(R_i) \mapsto \text{ran}(U)$
 - Tabular representation, additive function, ...
 - Tabular example $\phi_{Util}(Interference, Epid)$
 - Example from previous slide additive:

$$V(N, C, D) = -Noise \cdot 10^4 - Cost - Deaths \cdot 10^{12}$$

I	E	$Util$
false	false	10
false	true	-10
true	false	-20
true	true	2

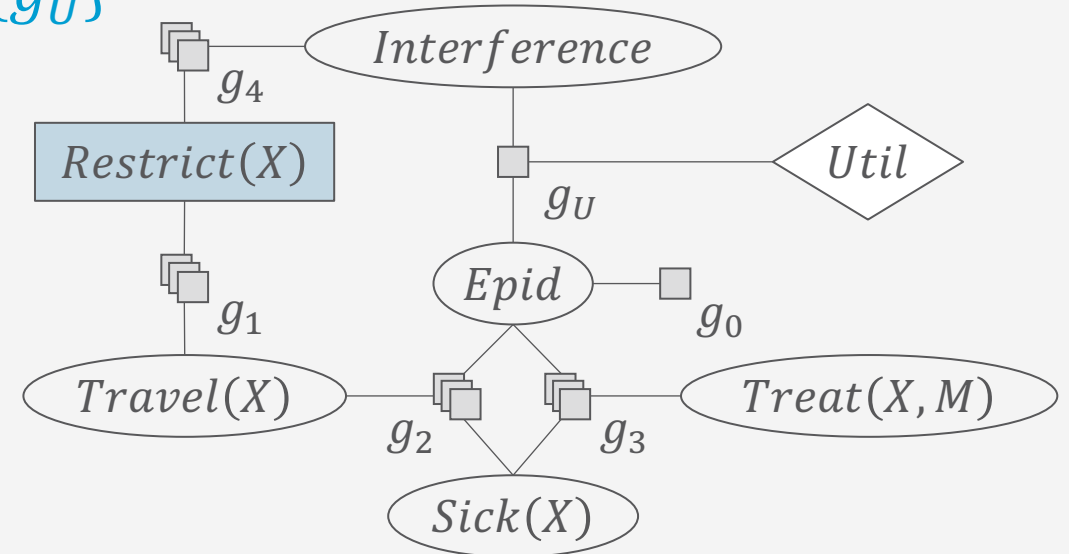


Parfactor-based Decision Model

- Decision model = Parfactor model that allows decision PRVs in the arguments of its parfactors as well as utility parfactors
 - For ease of exposition, we start with models with a utility *factor* mapping to a utility *variable*
 - Formally,

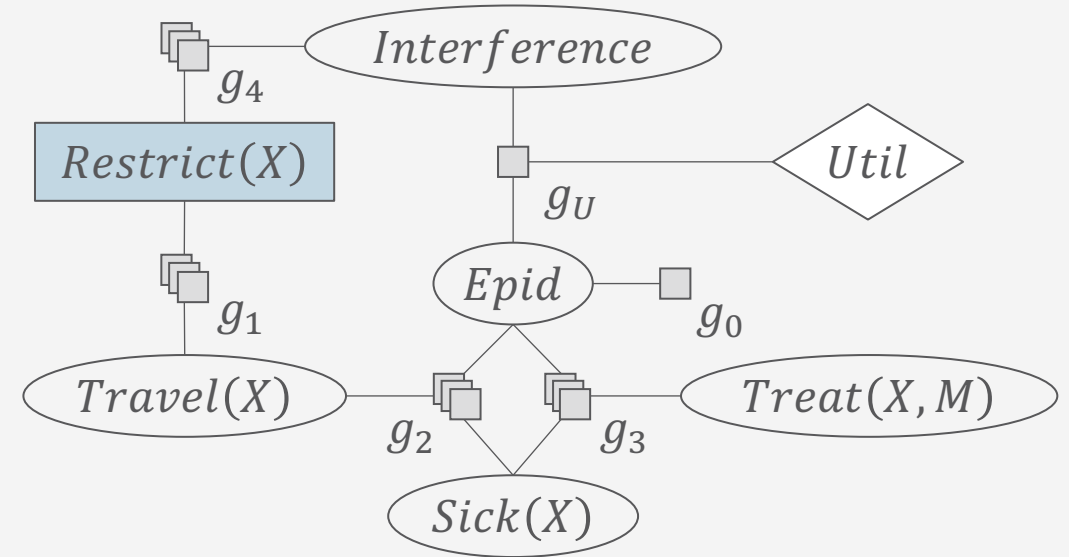
$$G = \{g_i\}_{i=1}^n \cup \{g_U\}$$

- g_i parfactors with (decision) PRVs as arguments
- g_U utility factor mapping to a utility variable U
 - $rv(g_U) = \emptyset$ for now
- E.g.,
 - $G = \{g_0, g_1, g_2, g_3, g_4, g_U\}$
 - T constraints



Decision Model: Action Assignments

- Let $\mathbf{D} = \{D_1, \dots, D_k\}_{|C}$ be the set of decision PRVs in G with a constraint C for the logical variables in $\mathbf{D}$
- Then, $\mathbf{d}$ is a compound event for $\mathbf{D}$ that assigns each decision PRV D_i a range value d_i
 - Refer to $\mathbf{d}$ as an **action assignment**
- E.g., without evidence in G ($\mathbf{e} = \emptyset$, $\top$ constraints)
 - Action $Restrict(X)$ with range $\{\text{ban}, \text{free}\}$
 - $\mathbf{d}_1 = \{\text{ban}\}$
 - $\mathbf{d}_2 = \{\text{free}\}$
 - Given another action D with range $\{d', d'', d'''\}$
 - $\mathbf{d}_1 = \{\text{ban}, d'\}$ $\mathbf{d}_4 = \{\text{free}, d'\}$
 - $\mathbf{d}_2 = \{\text{ban}, d''\}$ $\mathbf{d}_5 = \{\text{free}, d''\}$
 - $\mathbf{d}_3 = \{\text{ban}, d'''\}$ $\mathbf{d}_6 = \{\text{free}, d'''\}$

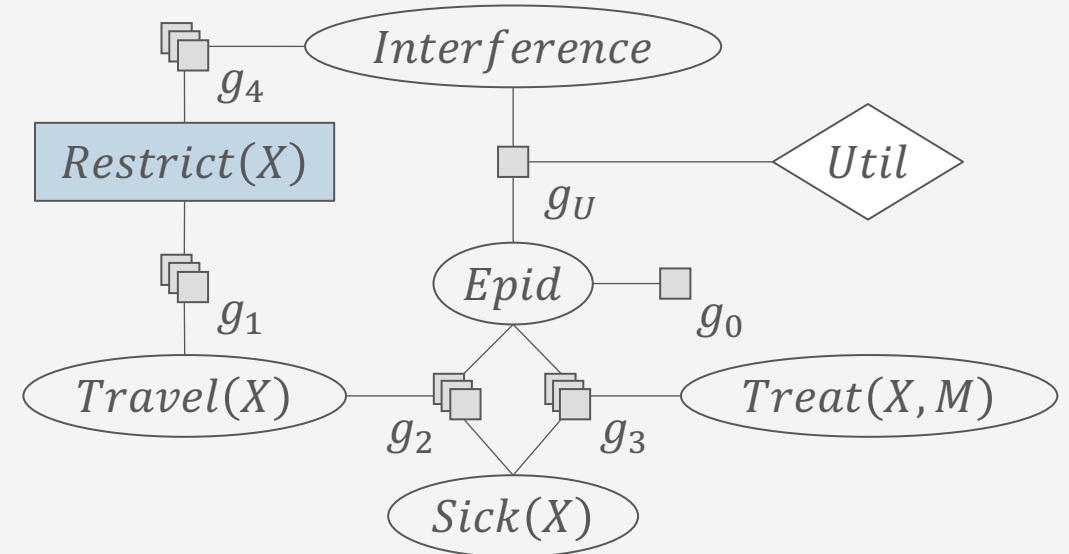


Decision Model: Setting Decisions

- Given a decision model G and an action assignment $\mathbf{d}$
- Let $G[\mathbf{d}]$ refer to G with $\mathbf{d}$ set, i.e.,

$$G[\mathbf{d}] = \text{absorb}(G, \mathbf{d})$$
 - In each g with decision PRV D_i ,
 - Drop the lines where $D_i \neq d_i$ and the column of D_i
- E.g., set $\mathbf{d}_1 = \{\text{ban}\}$ in $G = \{g_0, g_1, g_2, g_3, g_4, g_U\}$
 - $e = \emptyset$
 - Absorb $\mathbf{d}_1$ in g_1
 - $G[\mathbf{d}_1] = \{g_0, g'_1, g_2, g_3, g'_4, g_U\}$
 - $g'_1 = \phi'_1(\text{Travel}(X))$
 - $g'_4 = \phi'_4(\text{Interference})$

$R(X)$	I	ϕ_4	$R(X)$	$Tl(X)$	ϕ_1
free	false	1	free	false	1
free	true	0	free	true	1
ban	false	0	ban	false	1
ban	true	1	ban	true	0



Decision Model: Semantics

- **Semantics** of decision model $G = \{g_i\}_{i=1}^n \cup \{g_U\}$
 - Given an action assignment $\mathbf{d}$ for the *grounded* set of decision PRVs $\mathbf{D} = \{D_1, \dots, D_k\}_{|C}$ occurring in G
 - Then, the semantics is given by grounding and building a full joint distribution for the non-utility parafactors

$$P_G[\mathbf{d}] = \frac{1}{Z} \prod_{f \in \text{gr}(G[\mathbf{d}] \setminus \{g_U\})} f$$

$$Z = \sum_{r_1 \in \text{ran}(R_1)} \dots \sum_{r_N \in \text{ran}(R_N)} \prod_{f \in \text{gr}(G[\mathbf{d}] \setminus \{g_U\})} f$$

Semantics *multiplicative* with an inner product and outer sum: Multiply parafactors, then sum out PRVs.
→ Sum-product algorithms

- Utility parafactors irrelevant for probabilistic behaviour

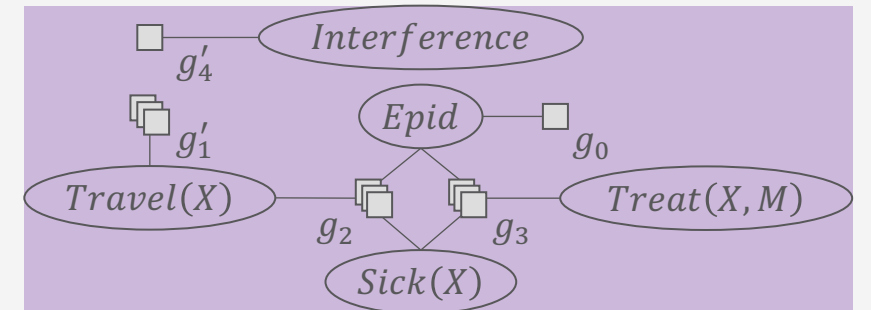
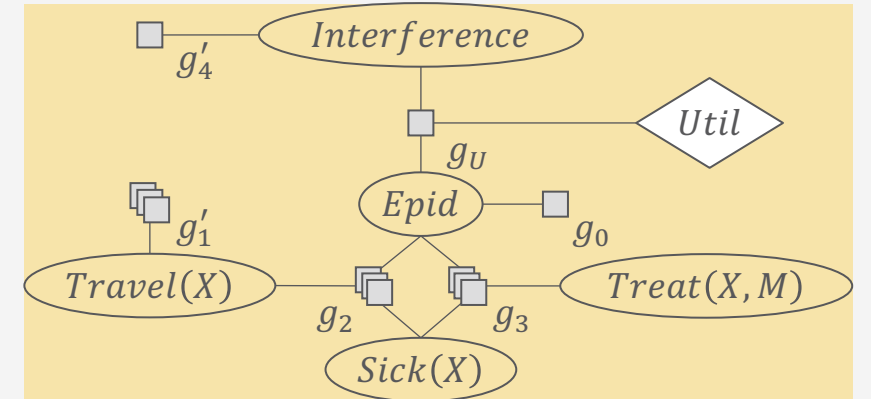
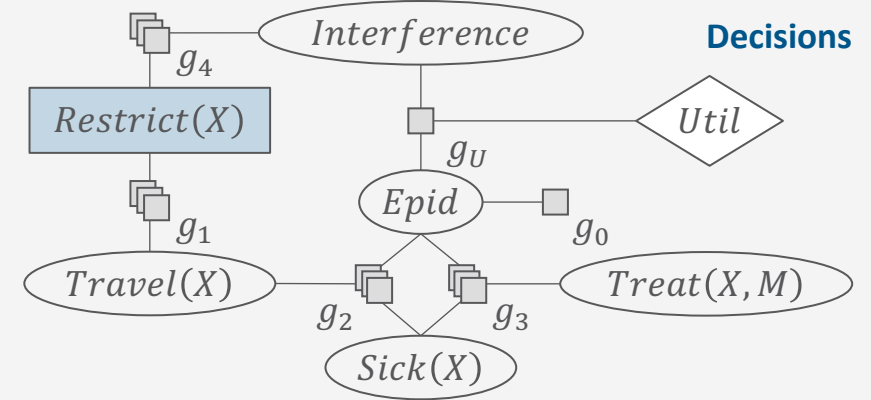
Decision Model: Example

- Decision model

$$G = \{g_0, g_1, g_2, g_3, g_4, g_U\}$$
- T constraints
- G with $\mathbf{d}_1 = \{\text{ban}\}$ set

$$G[\mathbf{d}_1] = \{g_0, g'_1, g_2, g_3, g'_4, g_U\}$$
 - $g'_1 = \phi'_1(\text{Travel}(X))$
 - $g'_4 = \phi'_4(\text{Interference})$
- Model relevant for probabilistic query answering:

$$G[\mathbf{d}_1] \setminus \{g_U\} = \{g_0, g'_1, g_2, g_3, g'_4\}$$



Expected Utility Queries

- Given a decision model $G = \{g_i\}_{i=1}^n \cup \{g_U\}$
 - One can ask queries for (conditional) marginal distributions or events as before given an action assignment $\mathbf{d}$ based on the semantics, $P_G[\mathbf{d}]$
 - New query type: query for an **expected utility (EU)**
 - What is the expected utility of making decisions $\mathbf{d}$ in G ?

$$eu(\mathbf{e}, \mathbf{d}) = \sum_{\mathbf{r} \in \text{ran}(\text{gr}(\text{rv}(g_U) \setminus \mathbf{E} \setminus \mathbf{D}))} P(\mathbf{r} | \mathbf{e}, \mathbf{d}) \cdot \phi_U(\mathbf{r}, \mathbf{e}, \mathbf{d})$$

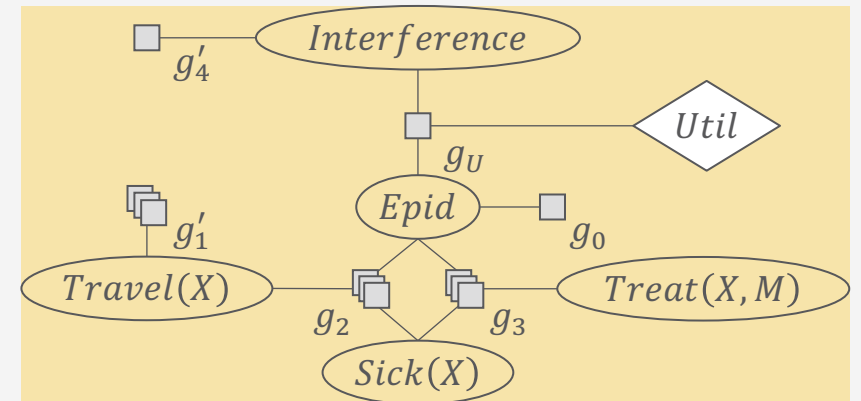
- $P(\mathbf{r} | \mathbf{e}, \mathbf{d})$ means that the PRVs not occurring in this expression need to be eliminated accordingly
 - i.e., $\mathbf{V} = \text{rv}(G) \setminus \mathbf{D} \setminus \mathbf{E} \setminus \text{rv}(g_U)$

EU Query: Example

- Expected utility of $\mathbf{d}_1 = \{\text{ban}\}$ in $G = \{g_0, g_1, g_2, g_3, g_4, g_U\}$

$$eu(\mathbf{d}_1) = \sum_{i \in \text{ran}(\text{Interference})} \sum_{e \in \text{ran}(\text{Epid})} P(e, i | \mathbf{d}_1) \cdot \phi_U(e, i)$$

- With $e = \emptyset$
- Compute $P(\text{Epid}, \text{Interference} | \mathbf{d}_1)$ in G
 - By computing $P(\text{Epid}, \text{Interference})$ in $G[\mathbf{d}_1]$
 - E.g., using LVE with model $G[\mathbf{d}_1] \setminus \{g_U\} = \{g_0, g'_1, g_2, g_3, g'_4\}$
 - $G[\mathbf{d}_1]$ depicted on the right



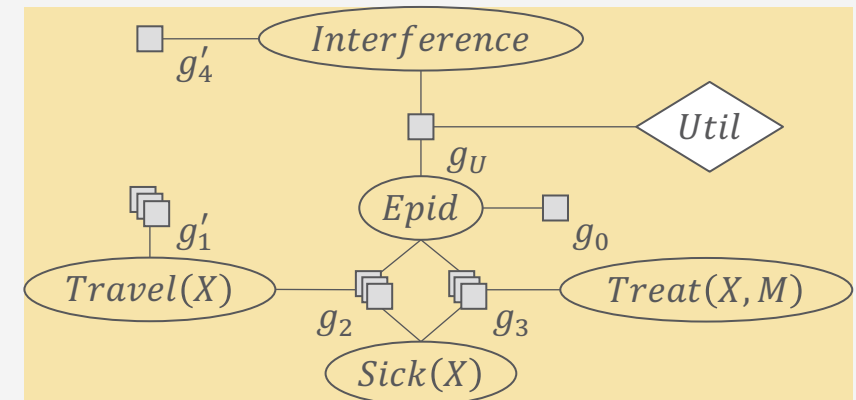
EU Query: Example

- Compute $P(Epid, Interference)$ in $G[\mathbf{d}_1] = \{g_0, g'_1, g_2, g_3, g'_4, g_U\}$
- Using LVE, eliminate all other PRVs in $G[\mathbf{d}_1]$:
 1. Eliminate $Treat(X, M)$
 2. Eliminate $Travel(X)$
 3. Eliminate $Sick(X)$
 4. Multiply all factors and normalise result
 - Result: $P(Epid, Interference)$ in $G[\mathbf{d}_1]$: $\phi(Epid, Interference)$
 - Corresponds to $P(Epid, Interference | \mathbf{d}_1)$ in G

Parfactors g'_1 and g'_4 would look differently, had we set $\mathbf{d}_2 = \{free\}$.

I	ϕ'_4
false	0
true	1

$Tl(X)$	ϕ'_1
false	1
true	0

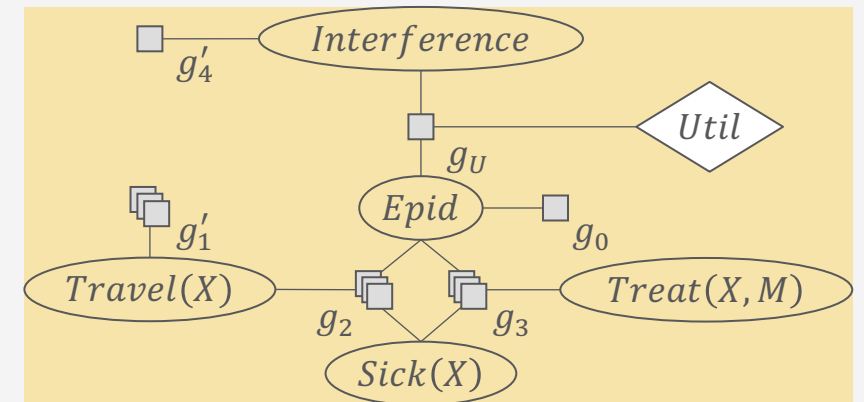


EU Query: Example

- Calculations with $|\text{dom}(M)| = 2$, $|\text{dom}(X)| = 3$:
 - Sum out $Treat(X, M)$, exponentiate result for M

E	$S(X)$	$Tt(X, M)$	ϕ_3
false	false	false	9
false	false	true	1
false	true	false	5
false	true	true	6
true	false	false	3
true	false	true	4
true	true	false	4
true	true	true	5

E	$S(X)$	ϕ'_3
false	false	$(9 + 1)^2 = 100$
false	true	$(5 + 6)^2 = 121$
true	false	$(3 + 4)^2 = 49$
true	true	$(4 + 5)^2 = 81$



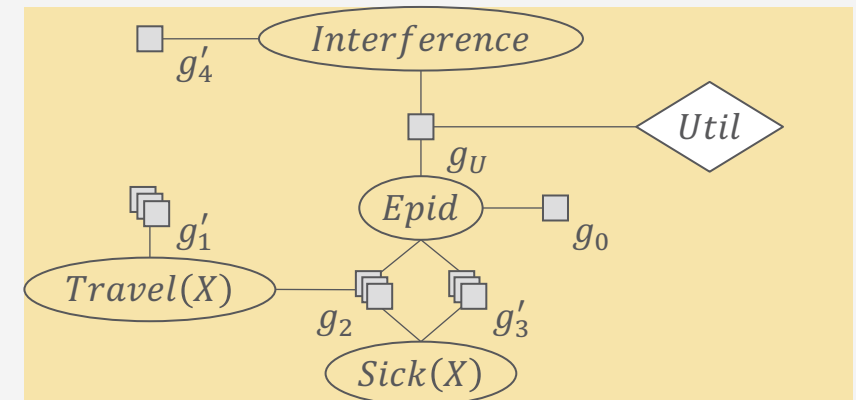
EU Query: Example

- Calculations with $|\text{dom}(M)| = 2, |\text{dom}(X)| = 3$:
 - Multiply g'_1, g_2 , sum out $Travel(X)$

E	$S(X)$	$Tl(X)$	$\phi_2 \cdot \phi'_1$
false	false	false	$10 \cdot 1 = 10$
false	false	true	$9 \cdot 0 = 0$
false	true	false	$4 \cdot 1 = 4$
false	true	true	$2 \cdot 0 = 0$
true	false	false	$8 \cdot 1 = 8$
true	false	true	$3 \cdot 0 = 0$
true	true	false	$5 \cdot 1 = 5$
true	true	true	$1 \cdot 0 = 0$

E	$S(X)$	ϕ'_{12}
false	false	$10 + 0 = 10$
false	true	$4 + 0 = 4$
true	false	$8 + 0 = 8$
true	true	$5 + 0 = 5$

$Travel(X)$	ϕ'_1
false	1
true	0



EU Query: Example

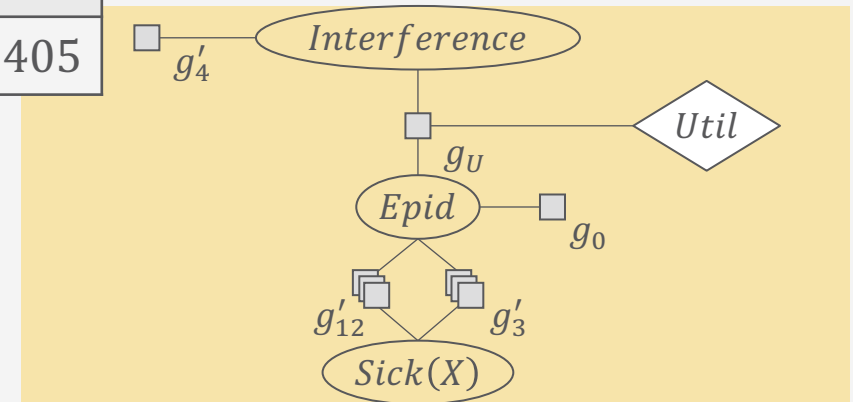
- Calculations with $|\text{dom}(M)| = 2$, $|\text{dom}(X)| = 3$:
 3. Multiply g'_{12}, g'_3 , sum out $Sick(X)$, exponentiate for X

E	$S(X)$	ϕ'_{12}
false	false	10
false	true	4
true	false	8
true	true	5

E	$S(X)$	ϕ'_3
false	false	100
false	true	121
true	false	49
true	true	81

E	$S(X)$	$\phi'_{12} \cdot \phi'_3$
false	false	$10 \cdot 100 = 1000$
false	true	$4 \cdot 121 = 484$
true	false	$8 \cdot 49 = 392$
true	true	$5 \cdot 81 = 405$

E	ϕ'_{123}
false	$(1000 + 484)^3 = 3,268,147,904$
true	$(392 + 405)^3 = 506,261,573$



EU Query: Example

- Calculations with $|\text{dom}(M)| = 2, |\text{dom}(X)| = 3$:
- 3. Multiply g'_{123}, g_0, g'_4 , normalise

E	ϕ'_{123}
false	$(1000 + 484)^3 = 3,268,147,904$
true	$(392 + 405)^3 = 506,261,573$

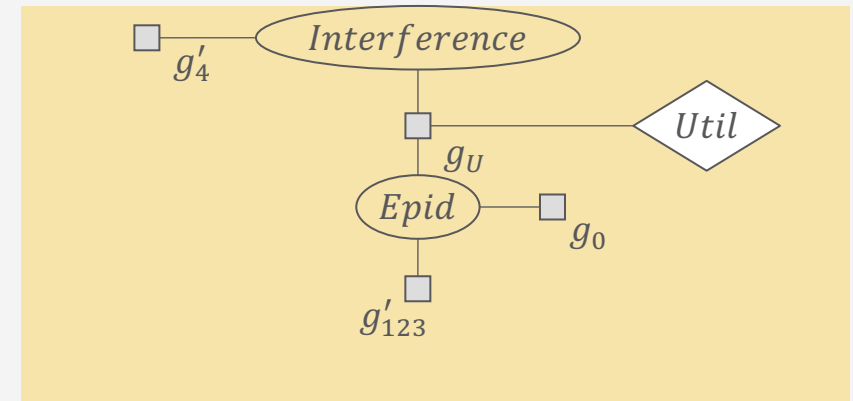
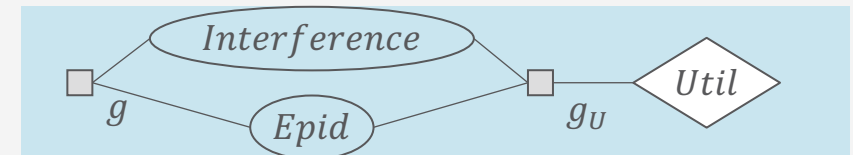
E	ϕ_0
false	10
true	1

I	ϕ'_4
false	0
true	1

I	E	$\phi'_{123} \cdot \phi_0 \cdot \phi'_4$	
false	false	$3,268,147,904 \cdot 10 \cdot 0 =$	0
false	true	$506,261,573 \cdot 1 \cdot 0 =$	0
true	false	$3,268,147,904 \cdot 10 \cdot 1 = 30,268,147,904$	0.984
true	true	$506,261,573 \cdot 1 \cdot 1 = 506,261,573$	0.016

ϕ
0.000
0.000
0.984
0.016

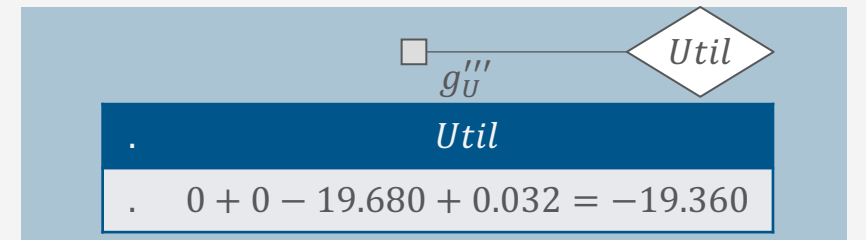
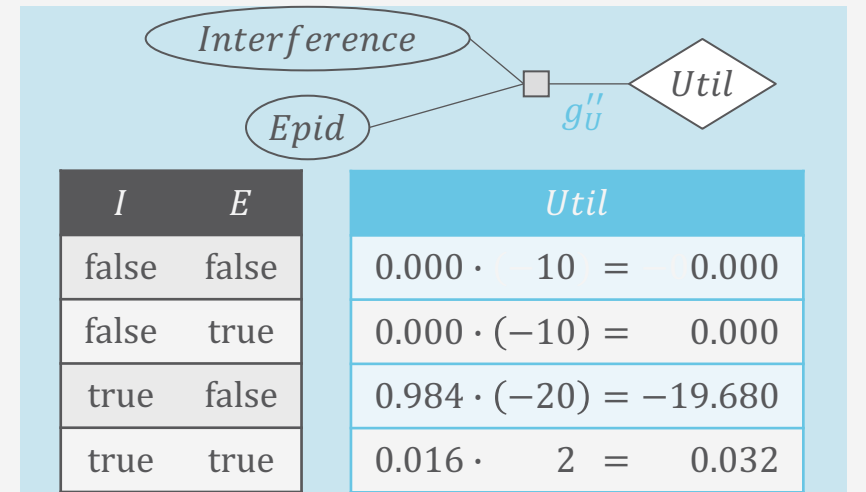
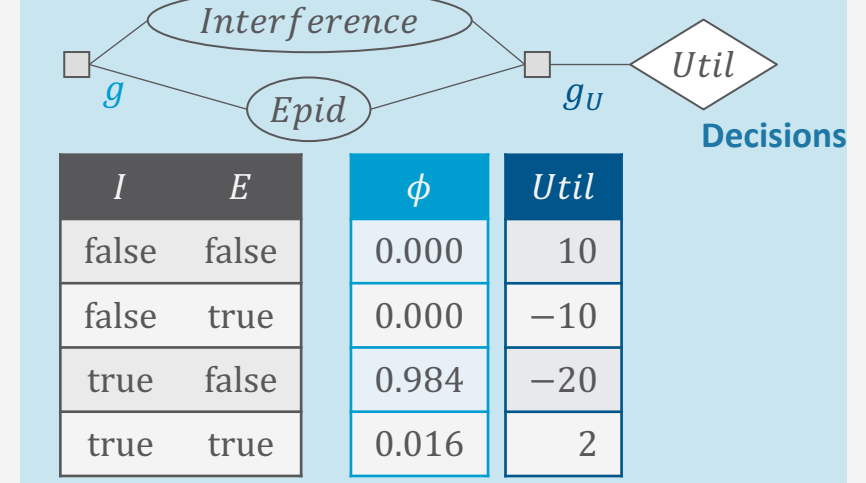
Result after normalising:
 $g = \phi(\text{Interference}, \text{Epid})$



EU Query: Example

- Result $\phi(Epid, Interference)$ for $P(e, i | \mathbf{d}_1)$ in G
- Expected utility of $\mathbf{d}_1 = \{\text{ban}\}$ in $G = \{g_0, g_1, g_2, g_3, g_4, g_U\}$

$$\begin{aligned}
 eu(\mathbf{d}_1) &= \sum_{i \in \text{ran}(Interference)} \sum_{e \in \text{ran}(Epid)} P(e, i | \mathbf{d}_1) \cdot \phi_U(e, i) \\
 &= \sum_{i \in \text{ran}(Interference)} \sum_{e \in \text{ran}(Epid)} \phi(e, i) \cdot \phi'_U(e, i) \\
 &= \sum_{i \in \text{ran}(Interference)} \sum_{e \in \text{ran}(Epid)} \phi''_U(Epid = e) \\
 &= \phi'''_U(.)
 \end{aligned}$$



Answering EU-Queries (with LVE)

- Given a decision model $G = \{g_i\}_{i=1}^n \cup \{g_U\}$, evidence e , and an action assignment d (*)
 - Absorb e and d in G
 - Calculate the posterior, $P(R|e, d)$, of the *Markov blanket of the utility node*
 - i.e., $R = \text{rv}(g_U) \setminus \text{rv}(d) \setminus \text{rv}(e)$ (remaining PRVs in g_U after previous step)
 - Using LVE: With R as the query terms, eliminate all non-query terms in G , i.e., call $\text{LVE}(G \setminus \{g_U\}, R, \emptyset)$
 - Evidence already absorbed, decisions made $\rightarrow e = \emptyset$ in the call
 - Calculate the expected utility by summing over the range values of R :

$$eu(e, d) = \sum_{r \in \text{ran}(R)} P(r|e, d) \cdot \phi_U(r)$$

(*) We need to talk about evidence and action assignments later.

- Using LVE: Eliminate remaining PRVs in G
 - Result: parfactor mapping empty argument to a single value (U)

MEU Problem

- Given a decision model G and evidence e , **maximum Expected Utility (MEU) problem**:
 - Find the action assignment that yields the highest expected utility in G
 - Formally,

$$\text{meu}(G|e) = (d^*, eu(e, d^*))$$

$$d^* = \arg \max_{d \in \text{ran}(D)} eu(e, d)$$

Additive semantics with inner sum and outer max: Sum up utilities, then pick maximum
→ Max-sum algorithms

- For an exact solution, $\text{meu}(G|e)$ requires an algorithm to go through **all** $d \in \text{ran}(D)$
 - Size of $\text{ran}(D)$ **exponential** in $|\text{gr}(D)|$

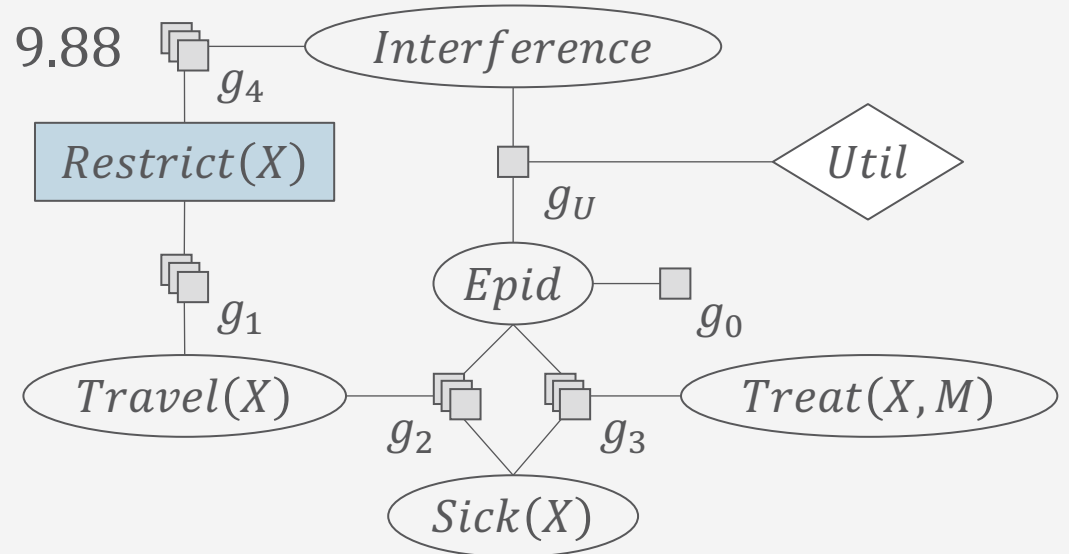
Alternative specification

$$\text{meu}(G|e) = \left(\arg \max_{d \in \text{ran}(D)} eu(e, d), \max_{d \in \text{ran}(D)} eu(e, d) \right)$$

MEU Problem: Example

- Problem instance with $G = \{g_0, g_1, g_2, g_3, g_U\}$, $e = \emptyset$:

$$\text{meu}(G) = (\mathbf{d}^*, eu(\mathbf{d}^*)) \quad \mathbf{d}^* = \arg \max_{d \in \{d_1, d_2\}} eu(d)$$
 - $d_1 = \{\text{ban}\}$, $d_2 = \{\text{free}\}$
 - Expected utility of $d_1 = \{\text{ban}\}$: $eu(d_1) = -19.36$
 - Expected utility of $d_2 = \{\text{free}\}$: $eu(d_2) = 9.88$
- Solution
 - $d^* = \arg \max_{d \in \{d_1, d_2\}} eu(d) = d_2$
 - $\text{meu}(G) = (d_2, 9.88)$
 - Decision that leads to maximum EU:
No travel restrictions



Lifted MEU

$$\begin{aligned} \text{meu}(G|e) &= (d^*, eu(e, d^*)) \\ d^* &= \arg \max_{d \in \text{ran}(D)} eu(e, d) \end{aligned}$$

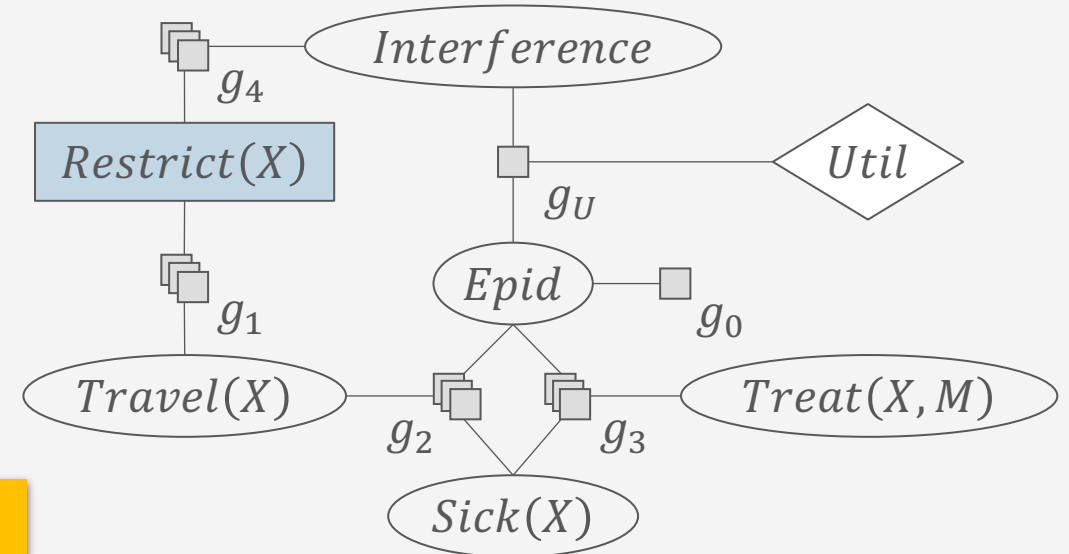
- In terms of semantics, $d \in \text{ran}(D)$ means
 - Grounding D and going through all possible combinations of assignments to $\text{gr}(D)$
- But: grounding is bad!
 - Combinatorial explosion: number of action assignments to test **exponential in size of $\text{gr}(D)$**
 - **Grounds** any parfactor in G containing a logvar of D
- Also: Grounding to full extent often unnecessary
 - Within **groups of indistinguishable constants**, the same decision will lead to its maximum influence in the MEU solution
 - Only need to test each assignment for complete group
- Thus: Test out all possible combinations of assignments w.r.t. the groups occurring in G
 - No longer exponential in the size of $\text{gr}(D)$!

Lifted MEU: Groups

- In parameterised models without evidence (or evidence for complete domains), $\mathbf{d} \in \text{ran}(\mathbf{D})$ means
 - Going through all possible combinations of assignments to $\mathbf{D}$
 - One group per logical variable
- In models with evidence affecting parfactors containing decision PRVs, $\mathbf{d} \in \text{ran}(\mathbf{D})$ means
 - Going through all possible combinations of assignments for each group of constants after evidence handling
 - Specifically, after shattering
- Effect: **size exponential in number of groups**

(*) Now is later.

$$\begin{aligned} \text{meu}(G|e) &= (\mathbf{d}^*, eu(e, \mathbf{d}^*)) \\ \mathbf{d}^* &= \arg \max_{\mathbf{d} \in \text{ran}(\mathbf{D})} eu(e, \mathbf{d}) \end{aligned}$$

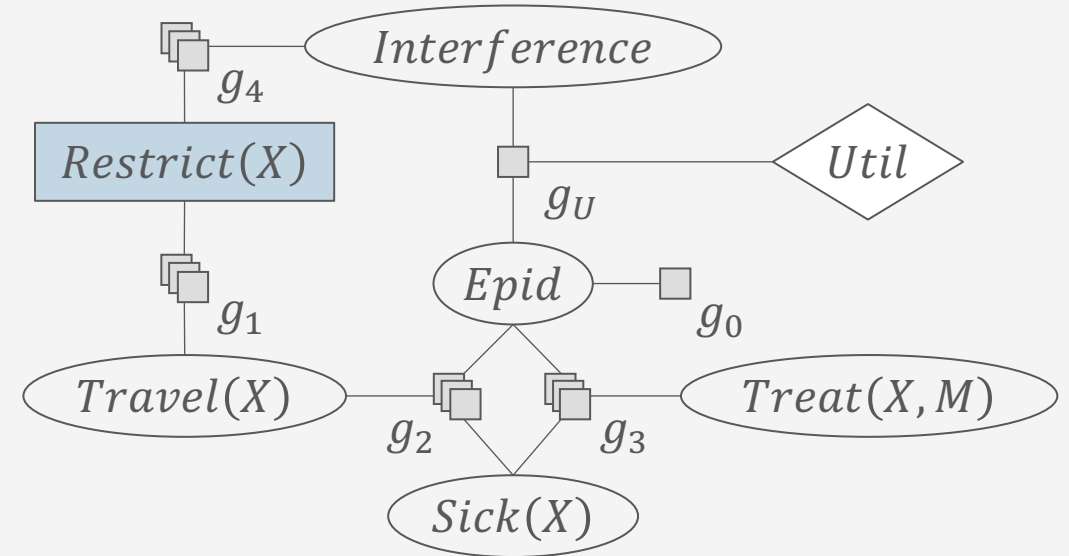


Lifted MEU: Groups – Example

- Decision model $G = \{g_0, g_1, g_2, g_3, g_4, g_U\}$
 - Decision PRV $Restrict(X)$ with range $\{\text{ban}, \text{free}\}$
 - Evidence $e = \{Sick(X') = \text{true}\}$, $\text{dom}(X') = \{x_1, \dots, x_{10}\}$
 - Overlap in domain of $X, X' \rightarrow$ Shattering duplicates g_1, g_2, g_3, g_4
 - For $\text{dom}(X') = \{x_1, \dots, x_{10}\}$, $\text{dom}(X'') = \{x_{10}, \dots, x_n\}$
 - Alternative: Duplicate + restrict constraints
- Action assignments
 - $R \triangleq Restrict, b \triangleq \text{ban}, f \triangleq \text{free}$
 - $d_1 = \{R(X'') = b, R(X') = b\}$
 - $d_2 = \{R(X'') = b, R(X') = f\}$
 - $d_3 = \{R(X'') = f, R(X') = b\}$
 - $d_4 = \{R(X'') = f, R(X') = f\}$

$$\text{meu}(G|e) = (d^*, eu(e, d^*))$$

$$d^* = \arg \max_{d \in \text{ran}(D)} eu(e, d)$$



Answering EU-Queries for MEU

- Given a decision model $G = \{g_i\}_{i=1}^n \cup \{g_U\}$, evidence e , and an action assignment d for groups in G after shattering
 - Calculate the posterior, $P(R|e, d)$, of the *Markov blanket of the utility node*
 - I.e., $R = \text{rv}(g_U) \setminus \text{rv}(a) \setminus \text{rv}(E)$ (remaining PRVs in g_u 's after previous step)
 - Using LVE: With R as the query terms and e, d as evidence, eliminate all non-query terms in G , i.e., call $\text{LVE}(G \setminus \{g_U\}, R, e \cup d)$
 - Calculate the expected utility by summing over the range values of R :

$$eu(e, d) = \sum_{r \in \text{ran}(R)} P(r|e, d) \cdot \phi_U(r)$$

- Using LVE: Eliminate remaining PRVs in $\{g\} \cup \{g_U\}$, $g = \text{LVE}(G \setminus \{g_U\}, R, e \cup d)$, i.e., call $\text{LVE}(\{g\} \cup \{g_U\}, R, e \cup d)$
 - e, d not yet handled in g_U ; alternatively: absorb e, d at beginning in G
 - Result: parfactor mapping empty argument to a single value (U)

LVE for MEU Problems

```
function MEU-LVE( $G = \{g_i\}_{i=1}^n \cup \{g_U\}, e$ )
```

```
  Absorb  $e$  in  $G$ 
```

```
   $d^* \leftarrow \emptyset$ 
```

```
   $eu_{max} \leftarrow -\infty$ 
```

```
  for each action assignment  $d$  in  $G$  do
```

```
     $g \leftarrow \text{LVE}(G \setminus \{g_U\}, \text{rv}(g_U), d)$ 
```

▷ g normalised

```
     $eu \leftarrow \text{LVE}(\{g_U, g\}, \emptyset, d)$ 
```

```
    if  $eu > eu_{max}$  then
```

```
       $d^* \leftarrow d$ 
```

```
       $eu_{max} \leftarrow eu$ 
```

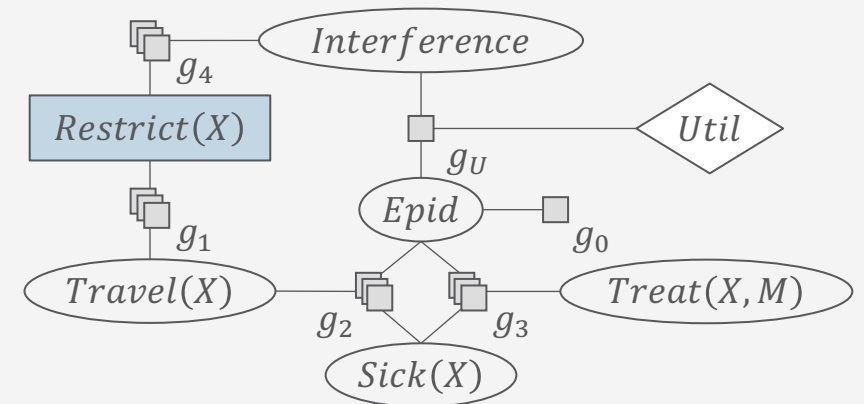
```
  return  $d^*$ 
```

LVE-MEU

- Modify to save all assignments that lie within ε -margin

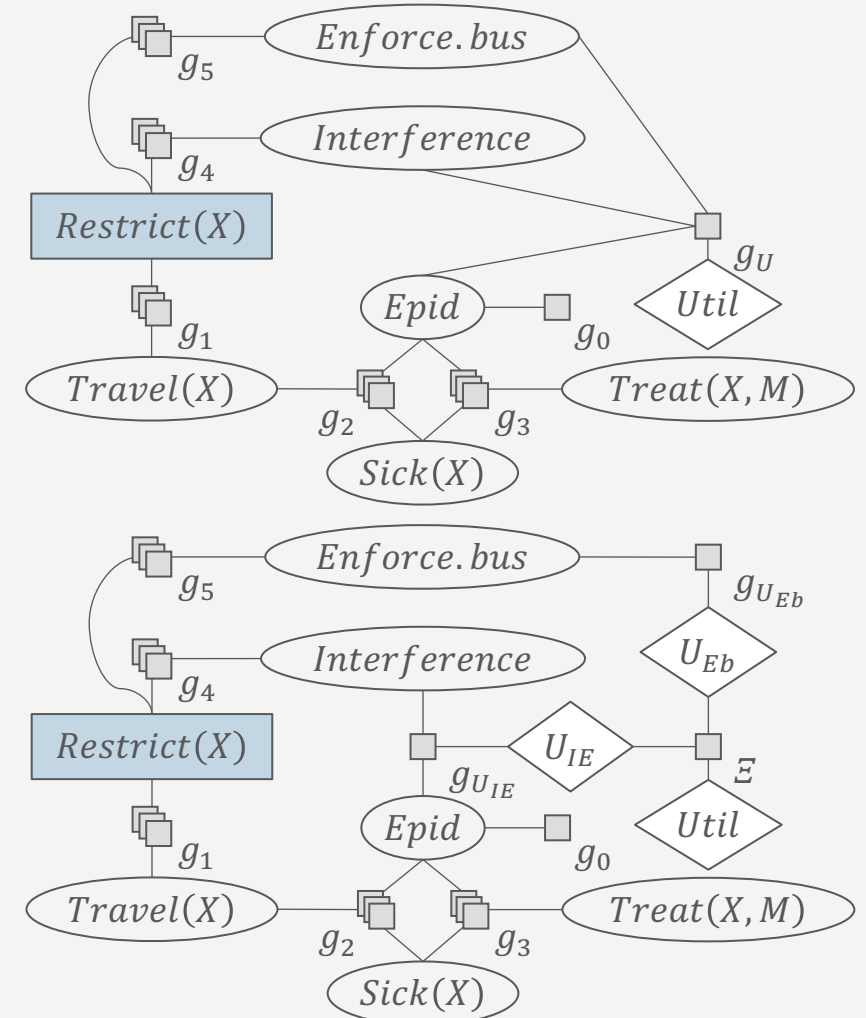
Structure in Multi-attribute Settings

- So far: Set of attributes without structure
 - Single utility functions mapping to one utility
 - Example: $\phi_U(Interference, Epid)$
- Cases with structure:
 - 1. Set of (distinguishable) attributes with structure**
 - Set of utility functions, mapping to interim utilities, combined into one overall utility
 - 2. Set of indistinguishable attributes**
 - Utility parfactor mapping to an interim utility PRV, which is combined into one utility
 - 3. Sets of distinguishable and indistinguishable attributes**
 - Set of utility parfactors and utility factors, combined into one utility
 - Considering structure requires a combination function $\mathcal{E}$



1. Set of Attributes with Structure: Example

- Example:
 - $\phi_{U_{IE}}(Interference, Epid)$
utility factor over *Interference*, *Epid*
 - $\phi_{U_{Eb}}(Enforce.bus)$
utility factor over *Enforce.bus*
 - (Effort it takes to enforce travel restriction on busses)
 - $\mathcal{E}$ a combination function, combining U_1, U_2 into $Util$
 - Could rewrite model using $\mathcal{E}$ into a model containing only one utility factor g_U (shown above)
 - $\phi_U(Interference, Epid, Enforce.bus)$
 $= \mathcal{E}[\phi_{U_{IE}}(Interference, Epid), \phi_{U_{Eb}}(Enforce.bus)]$



1. Set of Attributes with Structure

- Set of attributes that show MPI $\rightarrow$ Utility function "factorises" into sets of functions over attributes, combined with a combination function $\mathcal{E}$, i.e.,

$$U(r_1, \dots, r_M) = \mathcal{E}[\phi_1(r_1), \dots, \phi_M(r_M)]$$

- I.e., each $\phi_i(r_i)$ maps to its own interim utility, U_i , combined into an overall utility U through $\mathcal{E}$
- More general: Each f_i has a set of random variables $\mathbf{r}_i$ as input with $\mathbf{r} = \{r_1, \dots, r_M\} = \bigcup_{i=1}^m \mathbf{r}_i$
- Extended syntax: Decision model

$$G = \{g_i\}_{i=1}^n \cup \{g_u\}_{u=1}^m \cup \{\mathcal{E}\}$$

- Refer to submodel of potential parafactors by G_P and to submodel of utility factors by G_U
- $g_i = \phi_i(\mathcal{A}_i)_{|C_i}$ parafactors with (decision) PRVs as arguments
- $g_u = \phi_{U_u}(\mathcal{R}_u)$ utility *factors*, each mapping to a utility variable U_u
- $\mathcal{E}$ a combination function, combining all U_u into one U , i.e.,

$$\phi_U(r_1, \dots, r_M) = \mathcal{E} \left[\phi_{U_1} \left(\pi_{\mathcal{R}_1}(r_1, \dots, r_M) \right), \dots, \phi_{U_m} \left(\pi_{\mathcal{R}_m}(r_1, \dots, r_M) \right) \right]$$

1. Set of Attributes with Structure: EU Query & MEU Problem

- Given a decision model $G = G_P \cup G_U \cup \{E\} = \{g_i\}_{i=1}^n \cup \{g_u\}_{u=1}^m \cup \{E\}$
 - Query for an **expected utility (EU)**: change in sum over $rv(G_U)$ instead of $rv(g_U)$

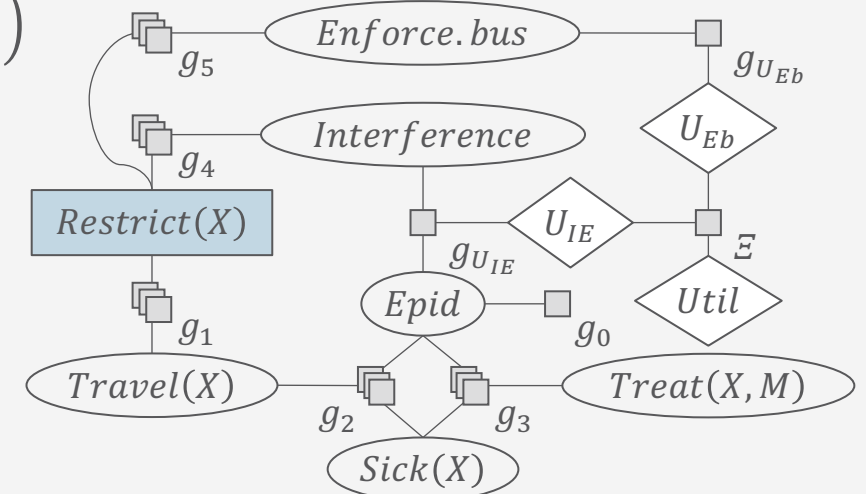
$$eu(e, d) = \sum_{v \in \text{ran}(rv(G_U) \setminus E \setminus D)} P(v|e, d) \cdot E \left[\phi_{U_1} \left(\pi_{R_1}(v, e, d) \right), \dots, \phi_{U_m} \left(\pi_{R_m}(v, e, d) \right) \right]$$

- If E addition, then

$$eu(e, d) = \sum_{v \in \text{ran}(gr(rv(G_U) \setminus E \setminus D))} P(v|e, d) \cdot \sum_{g_u \in G_U} \phi_{U_u} \left(\pi_{R_u}(v, e, d) \right)$$

- Works like MULTIPLY, i.e., like a join, but with *summing* of utilities instead of multiplying of potentials
- MEU problem: *no changes*

$$\begin{aligned}
 \text{meu}(G|e) &= (d^*, eu(e, d^*)) \\
 d^* &= \underset{d \in \text{ran}(D)}{\text{argmax}} eu(e, d)
 \end{aligned}$$



1. Set of Attributes with Structure: Additive Join

- Operator:

Operator 1 Additive join of utility factors

Operator ADD

Inputs:

(1) Utility factor $f_{u'} = \phi_{u'}(\mathcal{R}_{u'})$

(2) Utility factor $f_{u''} = \phi_{u''}(\mathcal{R}_{u''})$

Output: Utility factor $\phi_u(\mathcal{R}_u)$ such that

(1) $\mathcal{R}_u = \mathcal{R}_{u'} \bowtie \mathcal{R}_{u''}$ and

(2) for each valuation $\mathbf{r} \in \text{ran}(\mathcal{R}_u)$ with $\mathbf{r}_{u'} = \pi_{\mathcal{R}_{u'}}(\mathbf{r})$ and $\mathbf{r}_{u''} = \pi_{\mathcal{R}_{u''}}(\mathbf{r})$

$$\phi_u(\mathbf{r}) = \phi_{u'}(\mathbf{r}_{u'}) + \phi_{u''}(\mathbf{r}_{u''})$$

Postcondition: $G_U \equiv G_U \setminus \{f_{u'}, f_{u''}\} \cup \text{ADD}(f_{u'}, f_{u''})$

- Example

$\phi_U(\text{Interference}, \text{Epid}, \text{Enforce. bus})$

$= \Xi[\phi_{U_{IE}}(\text{Interference}, \text{Epid}), \phi_{U_{Eb}}(\text{Enforce. bus})]$

$= \phi_{U_{IE}}(\text{Interference}, \text{Epid}) + \phi_{U_{Eb}}(\text{Enforce. bus})$

$= \text{add}(g_{U_{IE}}, g_{U_{Eb}})$

I	E	U_{IE}
false	false	10
false	true	-10
true	false	-20
true	true	2

Eb	U_{Eb}
false	0
true	-10

I	E	Eb	U_{IE}	
false	false	false	$10 + 0 = 10$	$g_{U_{Eb}}$
false	false	true	$-10 - 10 = 0$	
false	true	false	$-10 + 0 = -10$	Eb
false	true	true	$-10 - 10 = -20$	
true	false	false	$-20 + 0 = -20$	E
true	false	true	$-20 - 10 = -30$	
true	true	false	$2 + 0 = 2$	til
true	true	true	$2 - 10 = 8$	

(X, M)

1. Set of Attributes with Structure: MEU-LVE

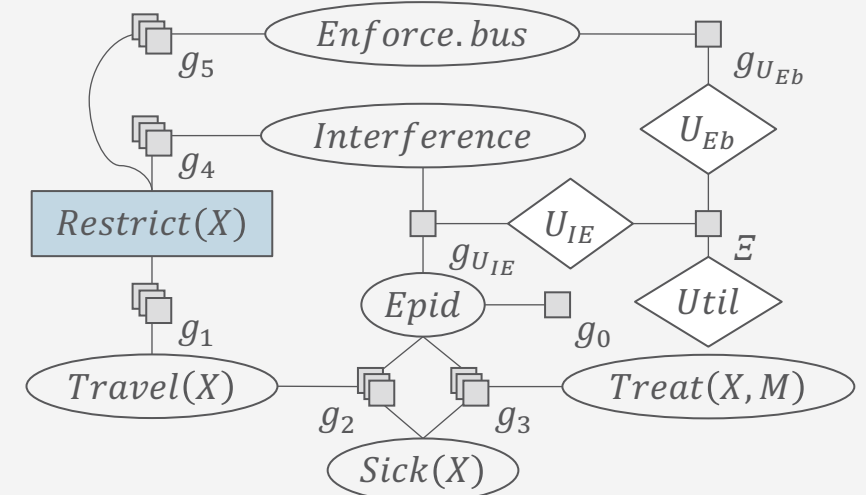
- Implement ADD operator
 - LVE with ADD operator referred to as LVE^{ADD}
- Changes in MEU-LVE
 - Input: decision model $G = G_P \cup G_U = \{g_i\}_{i=1}^n \cup \{g_u\}_{u=1}^m$
 - In for-loop:

$$g \leftarrow \text{LVE}(M \setminus G_U, \text{rv}(G_U), d) \quad \triangleright g \text{ normalised}$$

$$eu \leftarrow \text{LVE}^{\text{ADD}}(G_U \cup \{g\}, \emptyset, d)$$
 - If $\mathcal{E}$ not addition, need to implement (change LVE^{ADD} call)
 - Combines G_U into one g_U before multiplying with g and summing out the remaining variables

Splitting a single utility function into set of utility factors has upside of needing to learn / specify fewer entries **BUT:**

Complexity still exponential in M as combined into g



1. Set of Attributes with Structure: Simplification

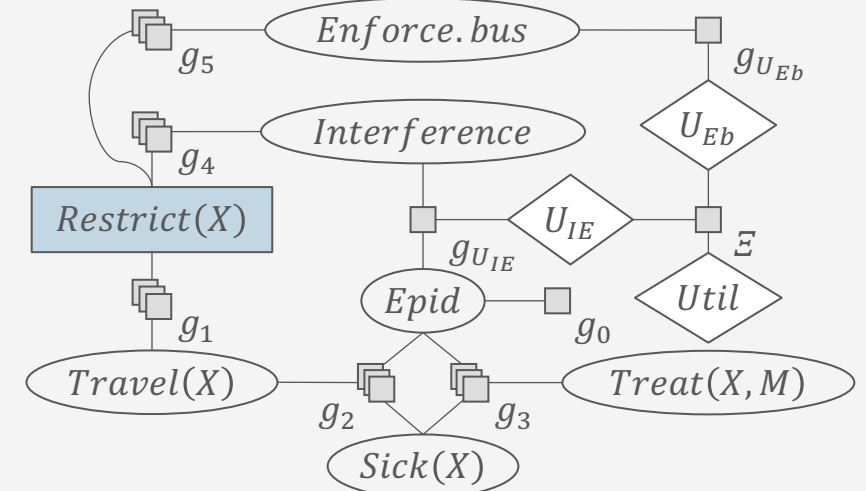
- Assume (conditional) independence between the different $rv(f_u)$ given $\mathbf{e}, \mathbf{d}$, i.e., $P(\mathbf{v}|\mathbf{e}, \mathbf{d}) = \prod_{u'=1}^m P(\mathbf{r}_{u'}|\mathbf{e}, \mathbf{d})$

$$\begin{aligned}
 eu(\mathbf{e}, \mathbf{d}) &= \sum_{\mathbf{v} \in \text{ran}(V)} P(\mathbf{v}|\mathbf{e}, \mathbf{d}) \cdot \mathbb{E}[\phi_{U_1}(\mathbf{r}_1), \dots, \phi_{U_m}(\mathbf{r}_m)] \\
 &= \sum_{\mathbf{v} \in \text{ran}(V)} P(\mathbf{v}|\mathbf{e}, \mathbf{d}) \cdot \sum_{u=1}^m \phi_u(\mathbf{r}_u) \\
 &= \sum_{u=1}^m \sum_{\mathbf{r}_u \in \text{ran}(rv(g_u))} P(\mathbf{r}_u|\mathbf{e}, \mathbf{d}) \cdot \phi_u(\mathbf{r}_u)
 \end{aligned}$$

Query on $rv(f_u)$ for each utility factor
 → Use multi-query algorithm like LJT

Only yields correct result under stochastic independence

- Idea similar to Boyen-Koller algorithm
- Preferential and stochastic independence do not follow from each other!



Derivation

$$\begin{aligned}
 eu(\mathbf{e}, \mathbf{d}) &= \sum_{v \in \text{ran}(V)} P(\mathbf{v}|\mathbf{e}, \mathbf{d}) \cdot \sum_{u=1}^m \phi_u(\mathbf{r}_u) = \sum_{v \in \text{ran}(V)} \sum_{u=1}^m P(\mathbf{v}|\mathbf{e}, \mathbf{d}) \cdot \phi_u(\mathbf{r}_u) = \sum_{u=1}^m \sum_{v \in \text{ran}(V)} P(\mathbf{v}|\mathbf{e}, \mathbf{d}) \cdot \phi_u(\mathbf{r}_u) \\
 &= \sum_{u=1}^m \sum_{v \in \text{ran}(V)} \prod_{u'=1}^m P(\mathbf{r}_{u'}|\mathbf{e}, \mathbf{d}) \cdot \phi_u(\mathbf{r}_u) \\
 &= \sum_{u=1}^m \sum_{\mathbf{r}_1 \in \text{ran}(R_1)} \dots \sum_{\mathbf{r}_m \in \text{ran}(R_m)} P(\mathbf{r}_1|\mathbf{e}, \mathbf{d}) \cdot \dots \cdot P(\mathbf{r}_m|\mathbf{e}, \mathbf{d}) \cdot \phi_u(\mathbf{r}_u) \\
 &= \sum_{u=1}^m \sum_{\mathbf{r}_1 \in \text{ran}(R_1)} P(\mathbf{r}_1|\mathbf{e}, \mathbf{d}) \cdot \dots \cdot \sum_{\mathbf{r}_m \in \text{ran}(R_m)} P(\mathbf{r}_m|\mathbf{e}, \mathbf{d}) \cdot \phi_u(\mathbf{r}_u) \\
 &= \sum_{u=1}^m \sum_{\mathbf{r}_u \in \text{ran}(R_u)} P(\mathbf{r}_u|\mathbf{e}, \mathbf{d}) \cdot \phi_u(\mathbf{r}_u) \cdot \underbrace{\sum_{u'=1, u' \neq u}^m P(\mathbf{r}_{u'}|\mathbf{e}, \mathbf{d})}_{= 1} \\
 &= \sum_{u=1}^m \sum_{\mathbf{r}_u \in \text{ran}(R_u)} P(\mathbf{r}_u|\mathbf{e}, \mathbf{d}) \cdot \phi_u(\mathbf{r}_u) \quad \begin{array}{l} = 1 \\ \text{(probability distributions} \\ \rightarrow \text{sums to 1)} \end{array}
 \end{aligned}$$

1. Set of Attributes with Structure: Simplification – Example

- Example: $\mathbf{d}_1 = \{ban\}$

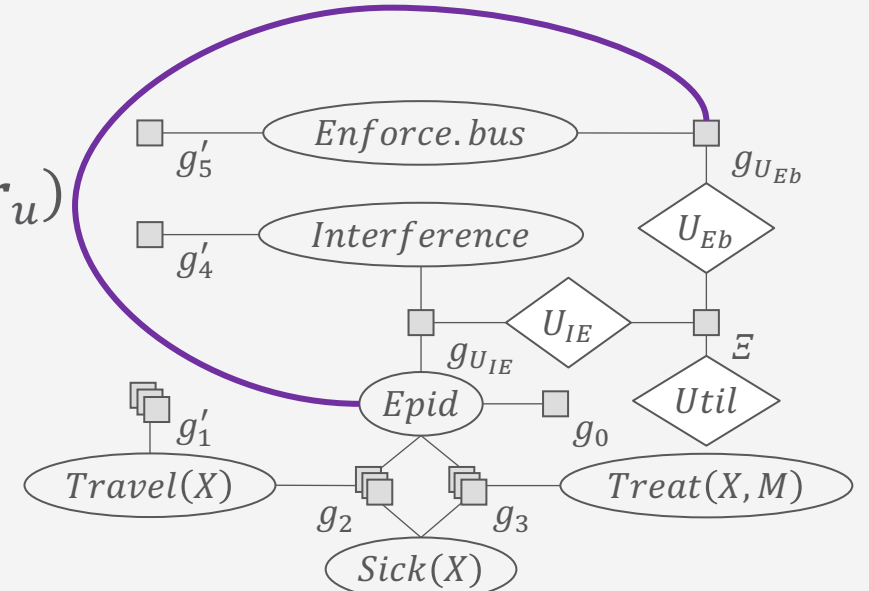
- $P(E, I, Eb | \mathbf{d}) = P(E | \mathbf{d}) \cdot P(I | \mathbf{d}) \cdot P(Eb | \mathbf{d})$
 $eu(\mathbf{d}_1)$

$$= \sum_{eb \in \text{ran}(Eb)} \sum_{i \in \text{ran}(I)} \sum_{e \in \text{ran}(E)} P(eb, i, e | \mathbf{d}) \cdot \sum_{u=1}^2 \phi_u(\mathbf{r}_u)$$

$$= \sum_{eb \in \text{ran}(Eb)} \sum_{i \in \text{ran}(I)} \sum_{e \in \text{ran}(E)} P(eb | \mathbf{d}) \cdot P(i | \mathbf{d}) \cdot P(e | \mathbf{d}) \cdot \sum_{u=1}^2 \phi_u(\mathbf{r}_u)$$

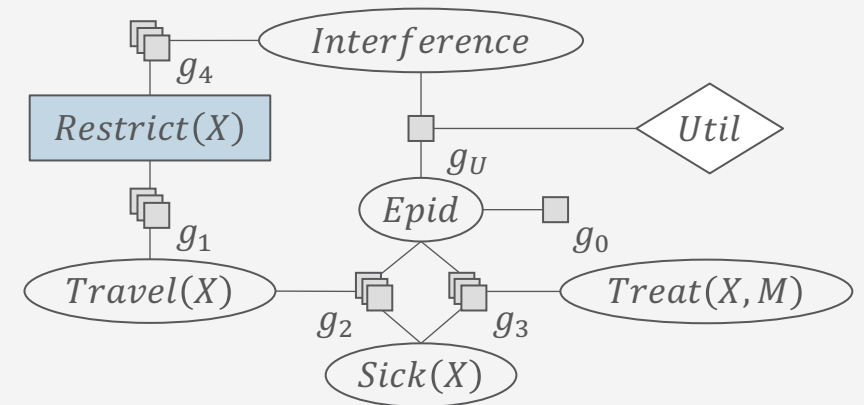
$$= \sum_{eb \in \text{ran}(Eb)} P(eb | \mathbf{d}) \cdot \phi_{Eb}(eb) + \sum_{i \in \text{ran}(I)} P(i | \mathbf{d}) \cdot \sum_{e \in \text{ran}(E)} P(e | \mathbf{d}) \cdot \phi_{IE}(i, e)$$

If adding *Epid* as an input to g_{UEb} ,
 $P(E, I, Eb | \mathbf{d}) \neq P(E, Eb | \mathbf{d}) \cdot P(E, I | \mathbf{d})$



Structure in Multi-attribute Settings

- So far: Set of attributes without structure
 - Single utility functions mapping to one utility
 - Example: $\phi_U(\text{Interference}, \text{Epid})$
- Cases with structure:
 1. **Set of (distinguishable) attributes with structure**
 - Set of utility functions, mapping to interim utilities, combined into one overall utility
 2. **Set of indistinguishable attributes**
 - Utility parfactor mapping to an interim utility PRV, which is combined into one utility
 3. **Sets of distinguishable and indistinguishable attributes**
 - Set of utility parfactors and utility factors, combined into one utility
 - Considering structure requires a combination function $\mathcal{E}$



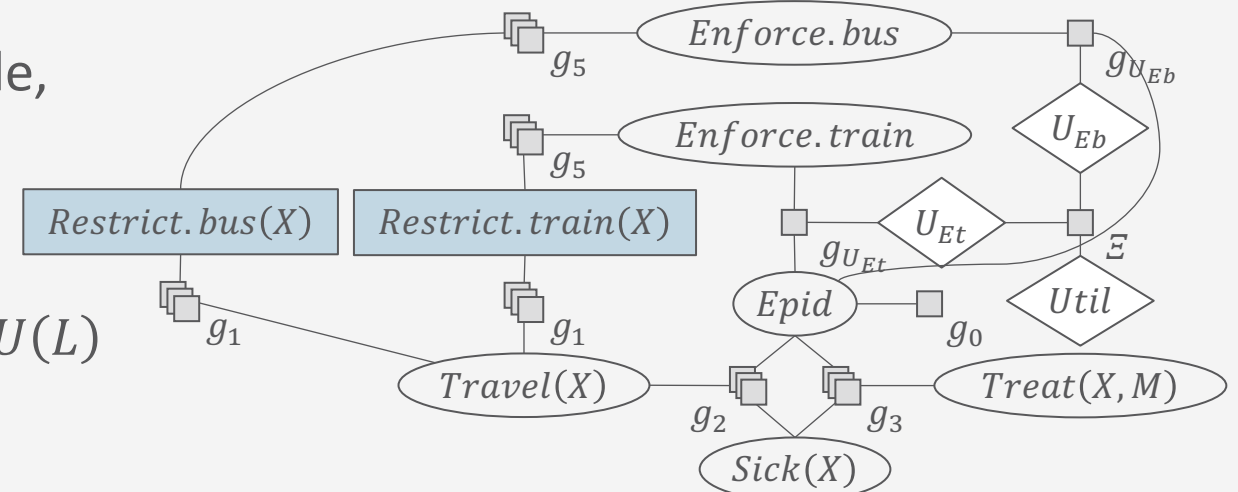
2. Set of Indistinguishable Attributes

- Indistinguishable attributes $R_1, \dots, R_M$ that show MPI $\rightarrow$ Utility function “factorises” into a set of *indistinguishable* functions f_i over indistinguishable attributes

- Utility function:

$$U(r_1, \dots, r_M) = \mathbb{E}[\phi_1(r_1), \dots, \phi_M(r_M)] = \mathbb{E}[\phi_u(r_1), \dots, \phi_u(r_M)]$$

- All ϕ_i are ϕ_u , mapping to an interim utility variable U_i
- If $\mathbb{E}$ addition, then $U(r_1, \dots, r_M) = M \cdot \phi_u(r_u)$
- Precondition:* For the f_i to be indistinguishable, the R_i need to be indistinguishable
- Encode indistinguishable attributes as PRV $R(L)$, $|\text{dom}(L)| = M$
- Then, encode interim utilities U_i as utility PRV $U(L)$
- Logical variables of utility PRV always follow logical variables in PRVs of utility function

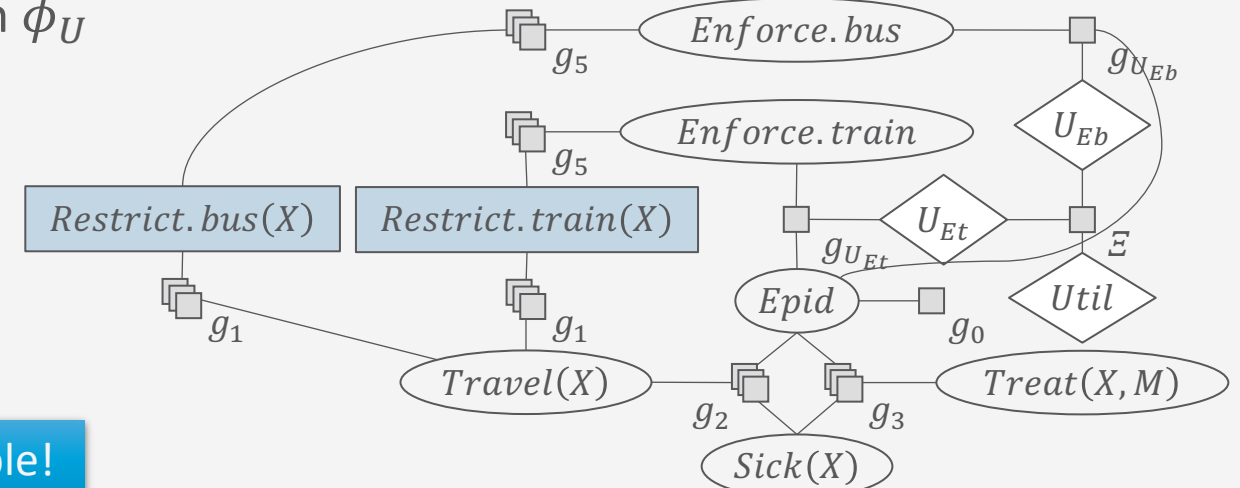
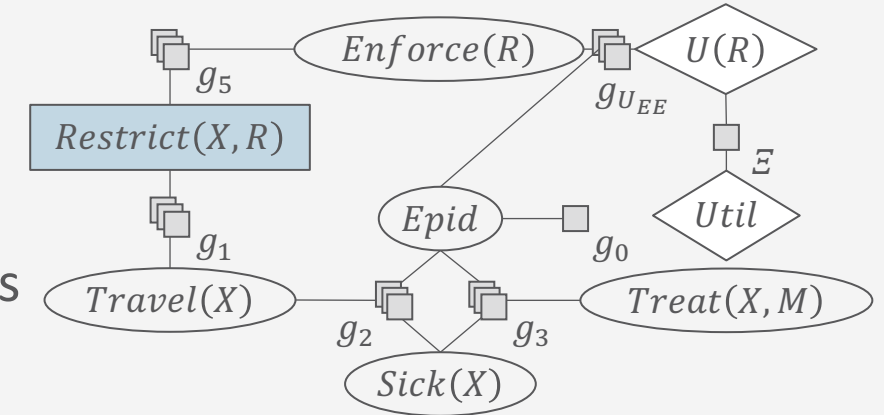


2. Set of Indistinguishable Attributes

- Extended syntax: Decision model

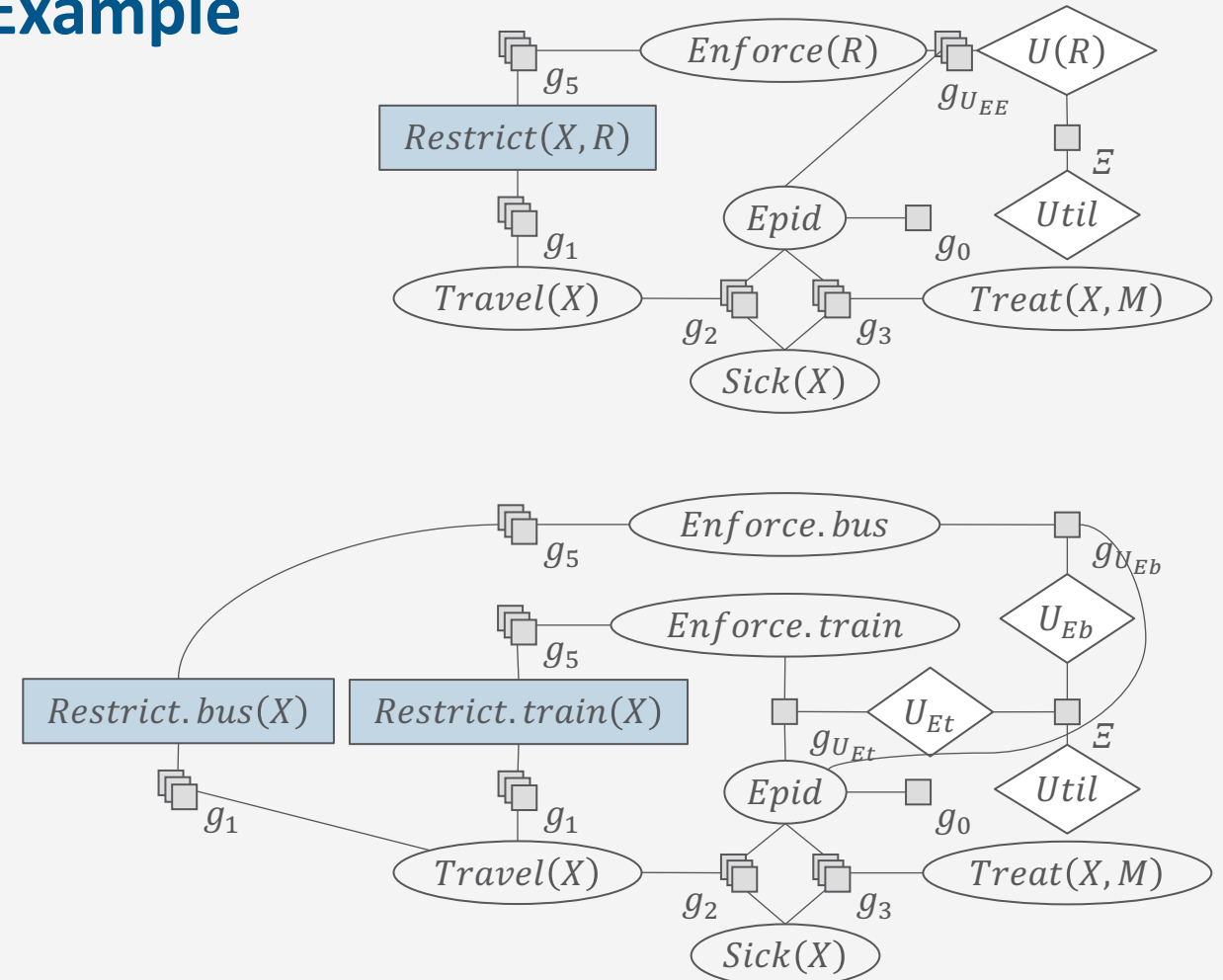
$$G = \{g_i\}_{i=1}^n \cup \{g_U\} \cup \{E\}$$
 - $g_i = \phi_i(\mathcal{A}_i)_{|C_i}$ parfactors with (decision) PRVs as arguments
 - $g_U = \phi_{U(\mathcal{L})}(\mathcal{A})$ a utility *parfactor* and $U(\mathcal{L})$ a utility PRV
 - $\mathcal{L} = \text{lv}(\mathcal{A})$ holds
 - $\text{gr}(g_U) = \{f_1, \dots, f_m\}$, all f_i with utility function ϕ_U
 - E a combination function, combining $U(\mathcal{L})$ into one U in lifted way (for liftability)
 - Addition yields a multiplication
 - Compare multiplication leading to an exponentiation in multiplicative semantics

As of now, logical variables in utility model possible!



2. Set of Indistinguishable Attributes: Example

- Example:
 - Assume that effort for enforcing travel restrictions on busses and trains is identical
 - Ground:
 - Utility factor $\phi_{U_{Eb}}(Epid, Enforce.bus)$
 - Utility factor $\phi_{U_{Et}}(Epid, Enforce.train)$
 - Lifted:
 - Utility parfactor $\phi_{U(R)}(Epid, Enforce(R))$
 - T constraint with $\text{dom}(R) = \{\text{train}, \text{bus}\}$
 - Combination function: addition
 - Lifted: multiplication with $|\text{dom}(R)|$



2. Set of Indistinguishable Attributes: EU Query & MEU Problem

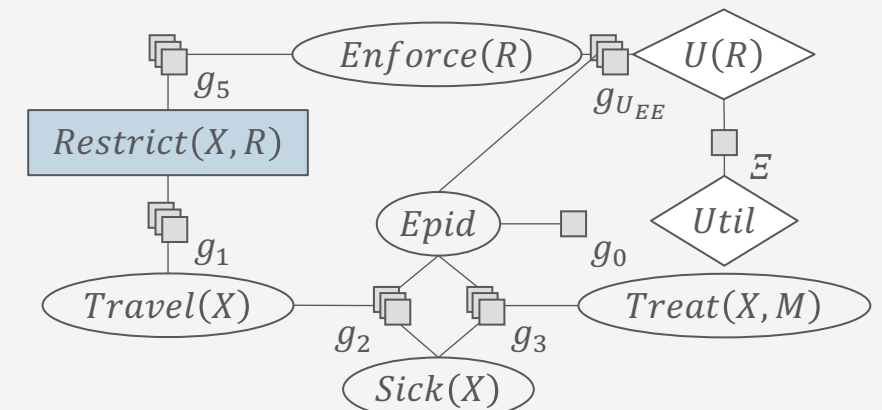
- Given a decision model $G = G_P \cup G_U \cup \{E\} = \{g_i\}_{i=1}^n \cup \{g_U\} \cup \{E\}$
 - Query for an **expected utility (EU)**: sum over $\text{gr}(\text{rv}(g_U))$

$$eu(e, d) = \sum_{v \in \text{ran}(\text{gr}(\text{rv}(G_U) \setminus E \setminus D))} P(v|e, d) \cdot E \left[\phi_{U_1} \left(\pi_{\mathcal{R}_1}(v, e, d) \right), \dots, \phi_{U_m} \left(\pi_{\mathcal{R}_m}(v, e, d) \right) \right]$$

- MEU problem: *no changes*

$$\begin{aligned} \text{meu}(G|e) &= (d^*, eu(e, d^*)) \\ d^* &= \underset{d \in \text{ran}(D)}{\text{argmax}} eu(e, d) \end{aligned}$$

- But: Given semantics, EU query calculation **not lifted!**
→ Can we avoid grounding?

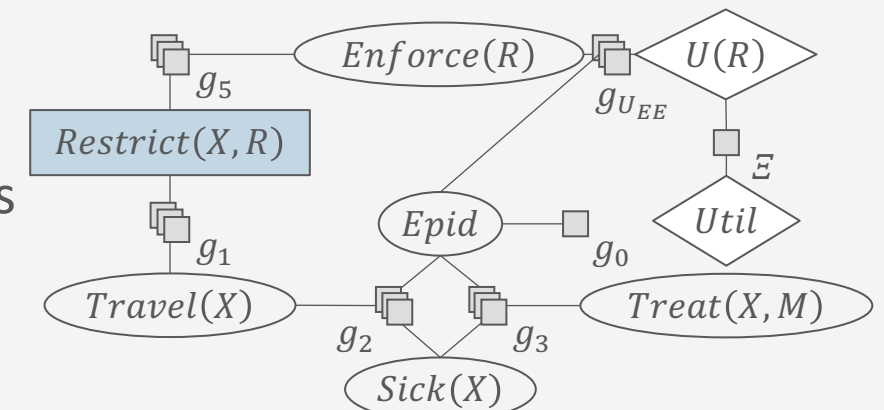


2. Set of Indistinguishable Attributes: Liftability

- Given a decision model $G = G_P \cup G_U \cup \{E\} = \{g_i\}_{i=1}^n \cup \{g_U\} \cup \{E\}$
 - Query for an **expected utility (EU)**: sum over $\text{gr}(\text{rv}(g_U))$

$$eu(e, d) = \sum_{v \in \text{ran}(\text{gr}(\text{rv}(G_U) \setminus E \setminus D))} P(v|e, d) \cdot E \left[\phi_{U_1} \left(\pi_{\mathcal{R}_1}(v, e, d) \right), \dots, \phi_{U_m} \left(\pi_{\mathcal{R}_m}(v, e, d) \right) \right]$$

- Changes in calculations for $eu(e, d)$ with $\text{rv}(G_U)$ now containing logical variables
 - $P(v|e, d)$ a parameterised query with $V = \text{rv}(G_U)$
 - If query liftable, then V as CRVs in answer → liftable
- But: logical variables in g_U not counted
 - If E addition: additive count-conversion for utility parfactors
 - Sum then over range of CRVs (include $\text{Mul}(h)$!)
 - Lifted calculations: Sum polynomial in domain sizes



2. Set of Indistinguishable Attributes: Additive Count-Conversion

- Operator:

Operator 2 Count-conversion for utility parafactors

Operator COUNT-CONVERT

Inputs:

- (1) Utility parafactor $g_u = \phi_{U(\mathbf{X})}(\mathcal{A})|_C$
- (2) logical variable $X \in lv(\mathcal{A})$ and $X \in \mathbf{X}$, to count in g_u

Preconditions:

- (1) There is exactly one atom $A_i \in \mathcal{A}$ with $X \in lv(A_i)$.
- (2) X is count-normalised w.r.t. $\mathbf{Z} = lv(\mathcal{A}) \setminus \{X\}$ in C .
- (3) For all counted logical variables $X^\#$ in g : $\pi_{X, X^\#}(C) = \pi_X(\pi_{\mathbf{X}}(C)) \times \pi_{X^\#}(\pi_{\mathbf{X}}(C))$.

Output: utility parafactor $\phi'_{U'}(\mathcal{A}')|_C$ such that

- (1) $U' = \#_X[U(\mathbf{X})]$,
- (2) $\mathcal{A}' = (A_1, \dots, A_{i-1}) \circ A'_i \circ (A_{i+1}, \dots, A_n)$, $A'_i = \#_X[A_i]$, and
- (3) for each valuation $\mathbf{a}'$ to $\mathcal{A}'$ with $a'_i = h$,

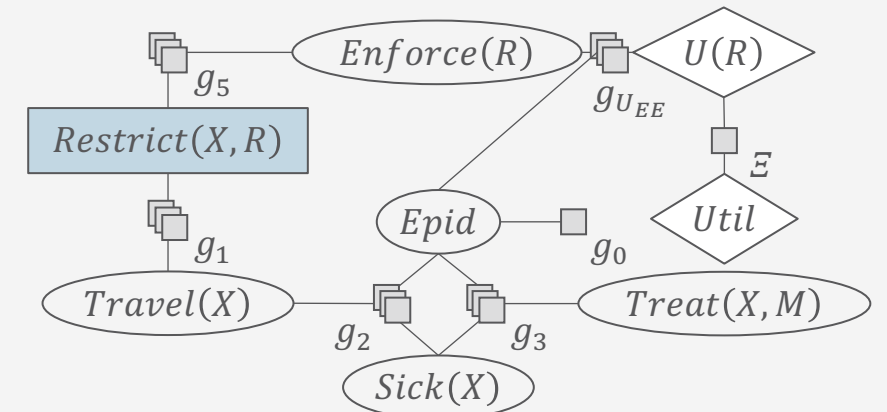
$$\phi'_{U'}(\mathbf{X})(\dots, a_{i-1}, h, a_{i+1}, \dots) = \sum_{a \in \text{ran}(A_i)} h(a_i) \phi_{U(\mathbf{X})}(\dots, a_{i-1}, a_i, a_{i+1}, \dots)$$

where h is a histogram $\{(a_i, n_i)\}_{i=1}^m$ with $m = |\text{ran}(A_i)|$, $a_i \in \text{ran}(A_i)$, $n_i \in \mathbb{N}$, and $\sum_{a_i \in \text{ran}(A_i)} h(a_i) = \text{ncount}_{X|\mathbf{Z}}(C)$, and $h(a_i) = n_i$.

Postcondition: $G_U \equiv G_U \setminus \{g_u\} \cup \text{COUNT-CONVERT}(g_u, X)$

Compare multiply count-conversion:

$$\begin{aligned} & \phi'(\dots, a_{i-1}, h, a_{i+1}, \dots) \\ &= \prod_{a_i \in \text{ran}(A_i)} \phi(\dots, a_{i-1}, a_i, a_{i+1}, \dots)^{h(a_i)} \end{aligned}$$

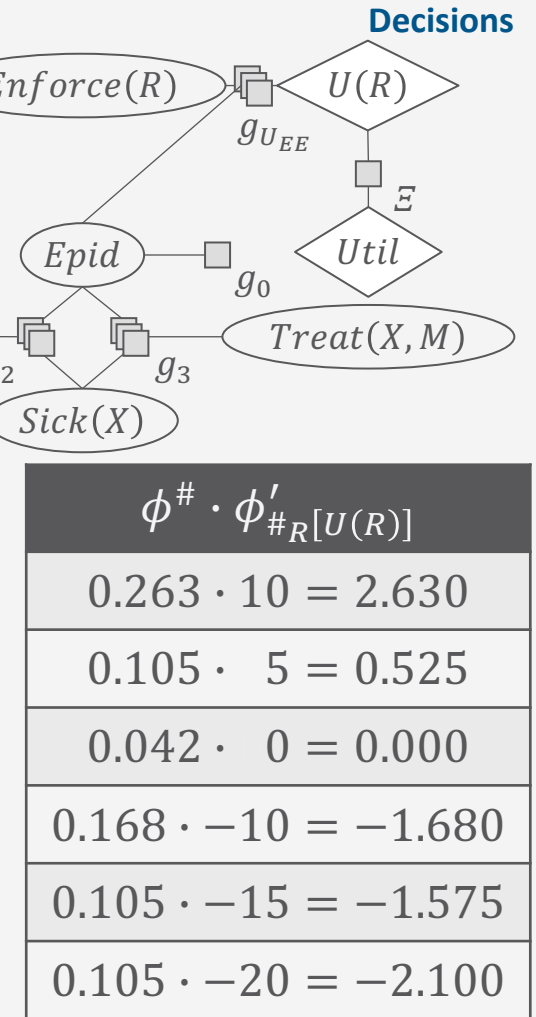


E	$E(R)$	ϕ
false	false	10
false	true	4
true	false	8
true	true	5

E	$\#_R[E(R)]$	$\phi^\#$	ϕ^n
false	[0,2]	$4^0 \cdot 10^2 = 100$	0.263
false	[1,1]	$4^1 \cdot 10^1 = 40$	0.105
false	[2,0]	$4^2 \cdot 10^0 = 16$	0.042
true	[0,2]	$5^0 \cdot 8^2 = 64$	0.168
true	[1,1]	$5^1 \cdot 8^1 = 40$	0.105
true	[2,0]	$5^2 \cdot 8^0 = 40$	0.105

E	$E(R)$	$\phi_{U(R)}$
false	false	5
false	true	0
true	false	-5
true	true	-10

E	$\#_R[E(R)]$	$\phi'_{\#_R[U(R)]}$
false	[0,2]	$0 \cdot 0 + 2 \cdot 5 = 10$
false	[1,1]	$1 \cdot 0 + 1 \cdot 5 = 5$
false	[2,0]	$2 \cdot 0 + 0 \cdot 5 = 0$
true	[0,2]	$0 \cdot -10 + 2 \cdot -5 = -10$
true	[1,1]	$1 \cdot -10 + 1 \cdot -5 = -15$
true	[2,0]	$2 \cdot -10 + 0 \cdot -5 = -20$



$$eu = 1 \cdot 2.630 + 2 \cdot 0.525 + 1 \cdot 0 + 1 \cdot -1.68 + 2 \cdot -1.575 + 1 \cdot -2.1 = -3.25$$

2. Set of Indistinguishable Attributes: Simplification

- Assume all groundings are independent
 - $\forall R(\mathbf{x}), R(\mathbf{y}) \in \text{gr}(\text{rv}(g_U)) : (R(\mathbf{x}) \perp R(\mathbf{y}) | \mathbf{e}, \mathbf{d})$
- Then,

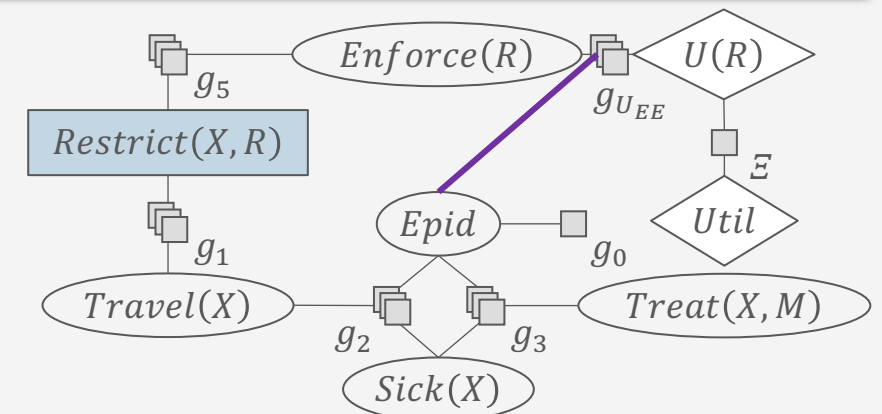
$$\begin{aligned} eu(\mathbf{e}, \mathbf{d}) &= \sum_{u=1}^m \sum_{\mathbf{a}_u \in \text{ran}(\text{rv}(g_U))} P(\mathbf{a}_u | \mathbf{e}, \mathbf{d}) \cdot \phi_u(\mathbf{a}_u) \\ &= m \cdot \sum_{\mathbf{a}_u \in \text{ran}(\text{rv}(g_U))} P(\mathbf{a}_u | \mathbf{e}, \mathbf{d}) \cdot \phi_u(\mathbf{a}_u) \end{aligned}$$

- $P(\mathbf{a}_u | \mathbf{e}, \mathbf{d})$ a representative query, i.e., a query over $\mathbf{A}_u = \text{rv}(g_U)$ with a representative grounding $\mathbf{x}$ of its logical variables $\mathbf{X} = \text{lv}(\mathbf{A}_u)$
- $m = |\text{gr}(g_U)|$

Lifted calculation:

- Sum *independent* of domain sizes m
- Multiplication with domain size in $O(\log m)$

Here, groundings are not independent because of *Epid*; without *Epid*, the groundings would be independent (of each other and anything else in the model)



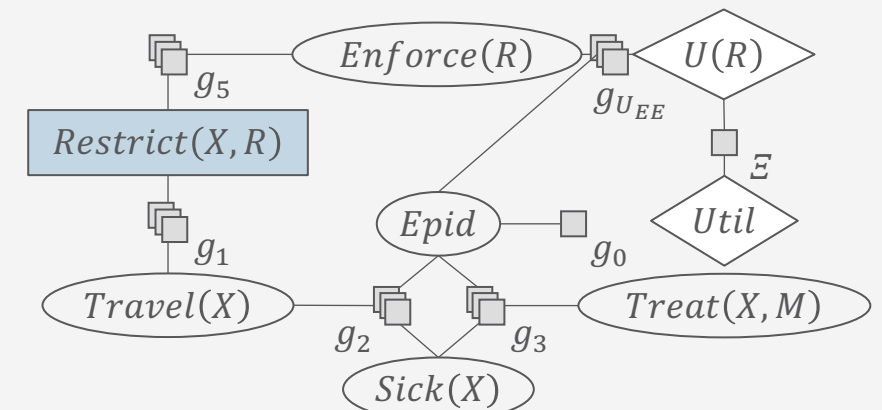
2. Set of Indistinguishable Attributes: MEU-LVE

- Implement ADD-COUNT-CONVERT operator
 - LVE with ADD operator and ADD-COUNT-CONVERT operator referred to as $\text{LVE}^{\text{addCC}}$
- Changes in MEU-LVE
 - Input: decision model $G = G_P \cup G_U = \{g_i\}_{i=1}^n \cup \{g_U\}$
 - In for-loop:

$g \leftarrow \text{LVE}(G \setminus G_U, \text{rv}(G_U), d)$ ▷ g normalised
 $eu \leftarrow \text{LVE}^{\text{addCC}}(G_U \cup \{g\}, \emptyset, d)$

- If $\mathcal{E}$ not addition, need to implement (change $\text{LVE}^{\text{addCC}}$ call)
- Count-converts the PRVs in g_U before multiplying with g and summing out the remaining variables
 - If PRVs in g_U not count-convertible → Ground logical variable and join partially grounded utility parfactors using ADD operator

If parameterised query *liftable*, then:
Complexity polynomial in M



2. Set of Indistinguishable Attributes: Logical Variables in Utility PRVs

- Definition says $\mathcal{L} = \text{lv}(\mathcal{A})$ holds for a utility parfactor $\phi_{U(\mathcal{L})}(\mathcal{A})$ a utility *parfactor* and $U(\mathcal{L})$ a utility PRV
- What about $\mathcal{L} \subset \text{lv}(\mathcal{A})$?
 ✗ Given grounding semantics, *not valid* as combination not defined
 - Example: $\phi_{Util}(\text{Restrict}(X), \text{Epid})$
 - Groundings: $\phi_{Util}(\text{Restrict}(\text{alice}), \text{Epid}), \phi_{Util}(\text{Restrict}(\text{eve}), \text{Epid}), \phi_{Util}(\text{Restrict}(\text{bob}), \text{Epid})$
- What about $\mathcal{L} \supset \text{lv}(\mathcal{A})$?
 ✓ Given grounding semantics, *valid* as only more utility factors occur
 - Example: $\phi_{U(R, \mathcal{C})}(\text{Enforce}(R), \text{Epid}), |\text{dom}(\mathcal{C})| = 3$
 - Groundings: $\phi_{U(b, c_1)}(\text{Enforce}(b), \text{Epid}), \phi_{U(b, c_2)}(\text{Enforce}(b), \text{Epid}), \phi_{U(b, c_3)}(\text{Enforce}(b), \text{Epid}),$
 $\phi_{U(t, c_1)}(\text{Enforce}(t), \text{Epid}), \phi_{U(t, c_2)}(\text{Enforce}(t), \text{Epid}), \phi_{U(t, c_3)}(\text{Enforce}(t), \text{Epid}),$

2. Set of Indistinguishable Attributes: Eliminating Logical Variables

- Grounding a utility parfactor with additional logical variables in its utility PRV leads to copies of utility factors over the same inputs that can be combined based on $\bar{E}$
 - Eliminate beforehand as a first step to simplify a model
- Operator for eliminating additional logical variables in a utility PRV of a utility parfactor

Operator 4 Logical variable elimination in utility PRVs

Operator ELIM-LOG-VARS

Inputs:

- (1) Utility parfactor $g_u = \phi_{U(\mathbf{X})}(\mathcal{A})|_C$
- (2) Logical variables $\mathbf{Y} \subseteq \mathbf{X}$

Preconditions:

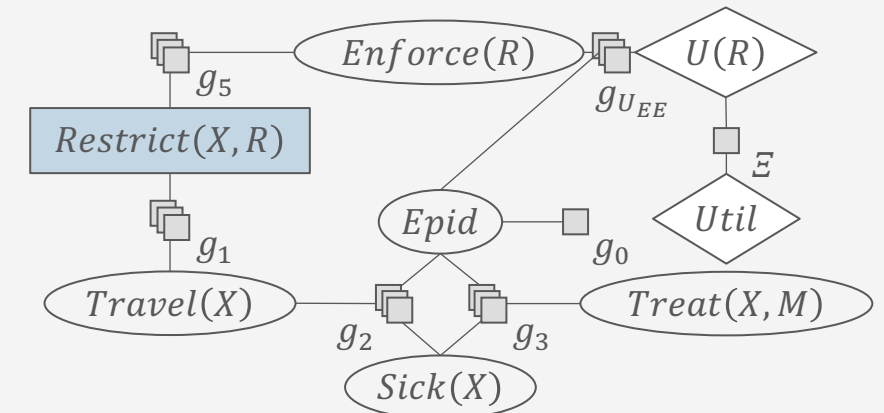
- (1) $\mathbf{Y}$ do not occur in $\mathcal{A}$, i.e., $\mathbf{Y} \cap lv(\mathcal{A}) = \emptyset$.
- (2) $\mathbf{Y}$ are count-normalised w.r.t. $lv(\mathcal{A})$ in C .

Output: utility parfactor $\phi'_{U(\mathbf{Z})}(\mathcal{A})|_{C'}$ such that

- (1) $\mathbf{Z} = \mathbf{X} \setminus \mathbf{Y}$,
- (2) $C' = C \setminus \mathbf{Y}$ (remove $\mathbf{Y}$ and its constants from C), and
- (3) for each valuation $\mathbf{a}$ to $\mathcal{A}$,

$$\phi'_{U(\mathbf{Z})}(\mathbf{a}) = ncount_{\mathbf{Y}|\mathbf{Z}}(C) \cdot \phi_{U(\mathbf{X})}(\mathbf{a})$$

Postcondition: $G_U \equiv G_U \setminus \{g_u\} \cup \text{ELIM-LOG-VARS}(g_u, \mathbf{Y})$



2. Set of Indistinguishable Attributes: Eliminating Logical Variables

- Example: $\phi_{U(R,C)}(Enforce(R), Epid), |\text{dom}(C)| = 3$
- New utility parfactor: $\phi_{U(R)}(Enforce(R), Epid)$
 - For all $en \in \text{ran}(Enforce(R)), ep \in \text{ran}(Epid)$:

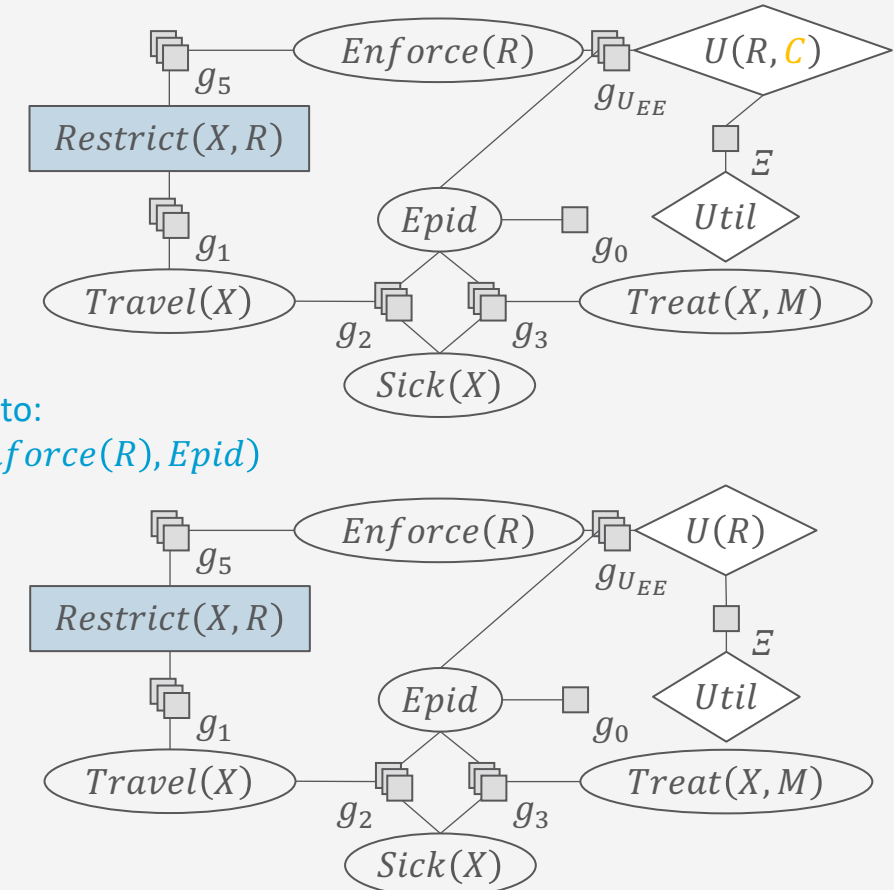
$$\phi_{U(R)}(en, ep) = 3 \cdot \phi_{U(R,C)}(en, ep)$$
- Ground comparison:
 - For all $en \in \text{ran}(Enforce(b)), ep \in \text{ran}(Epid)$:

$$\phi_{U(b,c_1)}(en, ep) + \phi_{U(b,c_2)}(en, ep) + \phi_{U(b,c_3)}(en, ep) = 3 \cdot \phi_{U(R,C)}(en, ep)$$
 - For all $en \in \text{ran}(Enforce(t)), ep \in \text{ran}(Epid)$:

$$\phi_{U(t,c_1)}(en, ep) + \phi_{U(t,c_2)}(en, ep) + \phi_{U(t,c_3)}(en, ep) = 3 \cdot \phi_{U(R,C)}(en, ep)$$

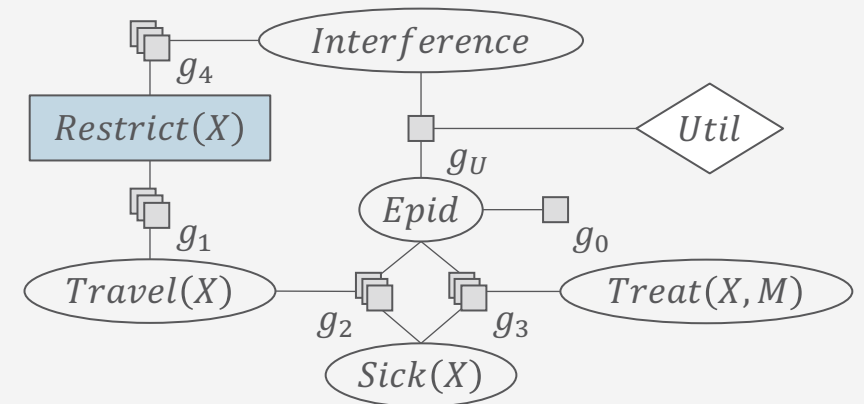
Combine into:

$\phi_{U(R)}(Enforce(R), Epid)$



Structure in Multi-attribute Settings

- So far: Set of attributes without structure
 - Single utility functions mapping to one utility
 - Example: $\phi_U(\text{Interference}, \text{Epid})$
- Cases with structure:
 1. **Set of (distinguishable) attributes with structure**
 - Set of utility functions, mapping to interim utilities, combined into one overall utility
 2. **Set of indistinguishable attributes**
 - Utility parfactor mapping to an interim utility PRV, which is combined into one utility
 3. **Sets of distinguishable and indistinguishable attributes**
 - Set of utility parfactors and utility factors, combined into one utility
 - Considering structure requires a combination function $\mathcal{E}$



3. Sets of Distinguishable & Indistinguishable Attributes

- Full expressiveness in terms of syntax: Allows for a set of utility parfactors as utility model
- Full decision model:

- Syntax

$$G = \{g_i\}_{i=1}^n \cup \{g_u\}_{u=1}^m \cup \{E\}$$

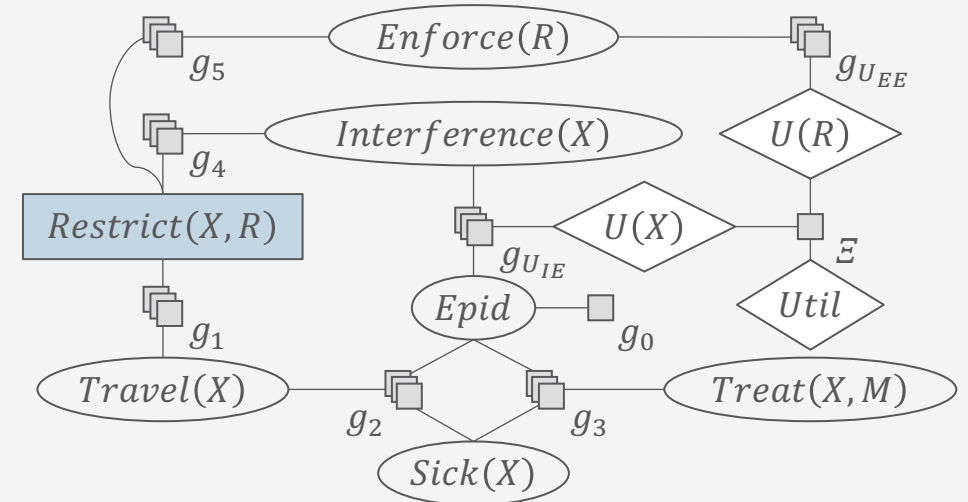
- $g_i = \phi_i(\mathcal{A}_i)_{|C_i}$ parfactor with (decision) PRVs as arguments
- $g_u = \phi_{U_u(\mathcal{L}_u)}(\mathcal{A}_u)_{|C_u}$ utility parfactor, mapping to a utility PRV $U_u(\mathcal{L}_u)$ with $\mathcal{L}_u = \text{lv}(\mathcal{A}_u)$
- E a combination function, combining all $U_u(\mathcal{L}_u)$ into one U
- Semantics: grounding semantics
 - Given an action assignment $\mathbf{d}$, full joint $P_{G_P}[\mathbf{d}]$ over grounding, multiplying, and normalising
 - EU queries sum over $\text{gr}(\text{rv}(G_U)) \rightarrow$ Lifiable parameterised query or simplification for liftability
 - MEU problem: With $\mathbf{D}$ the decision PRVs in G

$$\text{meu}(G|\mathbf{e}) = (\mathbf{d}^*, eu(\mathbf{e}, \mathbf{d}^*)) \quad \mathbf{d}^* = \underset{\mathbf{d} \in \text{ran}(\mathbf{D})}{\text{argmax}} eu(\mathbf{e}, \mathbf{d})$$

3. Sets of Distinguishable & Indistinguishable Attributes: Example

- Decision model $G = G_P \cup G_U \cup \{E\}$
 - $G_P = \{g_i\}_{i=0}^5$
 - $g_0 = \phi_0(Epid)$
 - $g_1 = \phi_1(Restrict(X, R), Travel(X))$
 - $g_2 = \phi_2(Epid, Sick(X), Travel(X))$
 - $g_3 = \phi_3(Epid, Sick(X), Treat(X, M))$
 - $g_4 = \phi_4(Restrict(X, R), Enforce(R))$
 - $g_5 = \phi_5(Restrict(X, R), Interf.(X))$
 - $G_U = \{g_u\}_{u=0}^5$
 - $g_{UEE} = \phi_{U(R)}(Enforce(R))$
 - $g_{UIE} = \phi_{U(X)}(Interference(X), Epid)$
 - E addition with additive operators for LVE

- In EU query
 - Independences given $Restrict(X, R)$
 - Between utility parfactor PRVs ✓
 - Between groundings of $Enforce(R)$ ✓



MEU Problems: Alternative Solution Approach

- Solving an MEU problem in decision model G with $\mathcal{E}(\mathbf{v})$ as short form for utility model:

$$\text{meu}(G|\mathbf{e}) = (\mathbf{d}^*, eu(\mathbf{e}, \mathbf{d}^*)), \mathbf{d}^* = \underset{\mathbf{d} \in \text{ran}(\mathbf{D})}{\text{argmax}} eu(\mathbf{e}, \mathbf{d}) = \underset{\mathbf{d} \in \text{ran}(\mathbf{D})}{\text{argmax}} \sum_{\mathbf{v} \in \text{ran}(\text{gr}(\text{rv}(G_U) \setminus E \setminus \mathbf{D}))} P(\mathbf{v}|\mathbf{e}, \mathbf{d}) \cdot \mathcal{E}(\mathbf{v})$$
- So far: for each $\mathbf{d}$, set $\mathbf{d}$, eliminate all PRVs not in G_U , eliminate remaining PRVs
 - Advantage: Reduced model by setting $\mathbf{d}$ (possible independences)
 - Disadvantage: possibly large $P(\mathbf{v}|\mathbf{e}, \mathbf{d})$ has to be computed
- Alternative: Compute a **maximum-a-posteriori (MAP) assignment** for the decision PRVs
 - Eliminate all non-decision PRVs in G_P by summing out, eliminate the decision PRVs by *maxing* out (replace sum operation by max-out operation)
 - Max-out: for each remaining world, pick the assignment with maximum value and store
 - Advantage: Does not require computing $P(\mathbf{v}|\mathbf{e}, \mathbf{d})$, easier to exploit factorisation
 - Disadvantage: Only a ranking (no true expected utility), no further independences through $\mathbf{d}$

Some References

- MEU in parfactor-based decision models
 - Warning: not as detailed as in these slides

Version using an early version of LVE, mashing early parfactor graphs and MLNs:

Udi Apsel and Ronan I. Brafman. Extended Lifted Inference with Joint Formulas. In: *UAI-11 Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, 2011.

MEU-LVE: Marcel Gehrke, Tanya Braun, Ralf Möller, Alexander Waschkau, Christoph Strumann, and Jost Steinhäuser. Towards Lifted Maximum Expected Utility. In: *Proceedings of the First Joint Workshop on Artificial Intelligence in Health in Conjunction with the 27th IJCAI, the 23rd ECAI, the 17th AAMAS, and the 35th ICML*, 2018.

Marcel Gehrke, Tanya Braun, Ralf Möller, Alexander Waschkau, Christoph Strumann, and Jost Steinhäuser. Lifted Maximum Expected Utility. In: *Artificial Intelligence in Health*, 2019.

- Markov logic decision networks (MLDNs)
 - MLN + parameterised decisions + utility weights
 - Probability + utility weights per first-order formula
 - Use weighted model counting to solve MEU problem

MLDNs: Aniruddh Nath and Pedro Domingos. A Language for Relational Decision Theory. In: *Proceedings of the International Workshop on Statistical Relational Learning*, 2009.

MLDNs + WMC: Udi Apsel and Ronan I. Brafman. Lifted MEU by Weighted Model Counting. In: *AAAI-12 Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 2012.

- Decision-theoretic Probabilistic Prolog (DTProbLog)
 - Utilities of DTProbLog programs combined into EU over theory defined by programs

DTProbLog: Guy Van den Broeck, Ingo Thon, Martijn van Otterlo, and Luc De Raedt. DTProbLog: A Decision-Theoretic Probabilistic Prolog. In: *AAAI-10 Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.

Interim Summary

- Decision models
 - Probabilistic graphical model extended with decision and utility variables
- Parfactor-based version
 - Decision PRVs, utility PRVs, utility parfactors, combination function
 - Collective decisions for groups of indistinguishable constants
- EU queries, MEU problem
 - Find set of actions (decisions) that lead to maximum expect utility
 - MEU-LVE using calls to LVE and LVE operators to answer EU queries
 - Combination function addition $\rightarrow$ additive join + count-conversion

Outline: 6. Decision-theoretic Models and Inference

a. Utility Theory

- Preferences, maximum expected utility (MEU) principle
- Utility function, multi-attribute utility theory

b. Static Decision Making

- Modelling, semantics, inference tasks
- Inference algorithm: LVE for MEU as an example

c. Sequential Decision Making

- Modelling, semantics, sequential MEU problem
- Inference algorithm: LDJT for MEU as an example
- Acting

Decision Making over Time

So far:

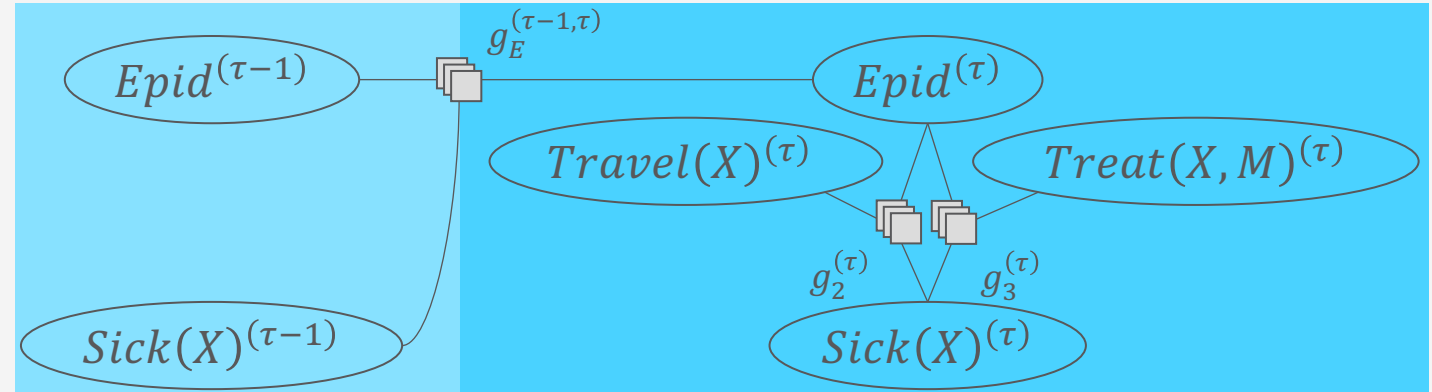
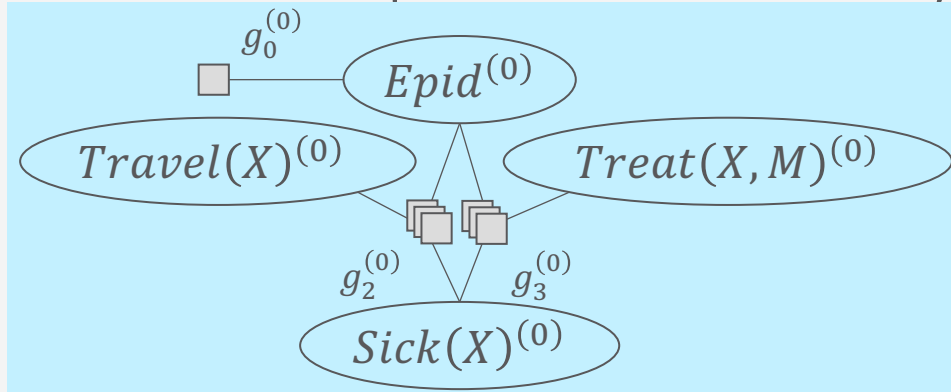
- Calculate the expected utility of a decision and its effect on the (current) state

With time

- Decisions need to be made at each time step
 - Each time step yields a utility
 - Decisions/actions have an effect not only on the current state/utility but also on the future and therefore, future decisions
- Need to consider a temporal sequence of decisions and project its effect into the future
- Requires calculating the expected utility over a sequence

Sequential Parfactor-based Model

- Sequential parfactor-based model $(G^0, G^\rightarrow)$ with
 - $G^0 = \{g_i^{(0)}\}_{i=1}^{n_0}$, $g_i^{(0)} = \phi_i^{(0)}(A_1^{(0)}, \dots, A_{k_i}^{(0)})$
 - $G^\rightarrow$ a 2-slice model, i.e., $G^\rightarrow = \{g_i^{(\tau)}\}_{i=1}^n \cup \{g_j^{(\tau-1, \tau)}\}_{j=1}^m$
 - Example: $G^0 = \{g_2^{(0)}, g_3^{(0)}\} \cup \{g_0^{(0)}\}$, $G^\rightarrow = \{g_2^{(\tau)}, g_3^{(\tau)}\} \cup \{g_E^{(\tau-1, \tau)}\}$
 - No assumptions about observability



Semantics: Unroll model M for T steps, ground, build full joint.

$$M^{(0:T)} = M^0 \cup \bigcup_{t=1}^T M^{\rightarrow|\tau=t}$$

$$P_M^T = \frac{1}{Z} \prod_{\zeta \in gr(M^{(0:T)})} \zeta$$

Query Answering Problem & Query Types

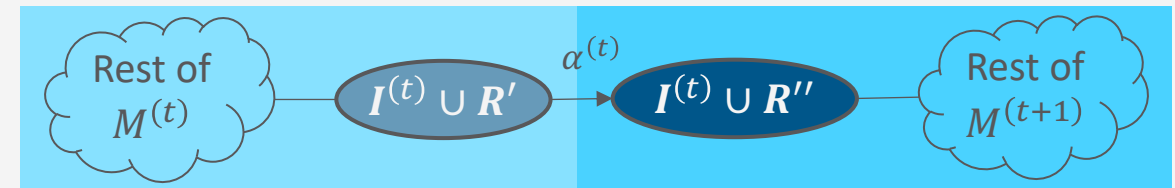
Inference Tasks

- Formally, query on a dynamic model M over template random variables R :

$$P(\mathcal{S}^{(\pi)} | e^{(0:t)})$$
 - where t the current step, $\mathcal{S} \subseteq R$ query terms, $E \subseteq R$ evidence terms, and usually $\mathcal{S} \cap E = \emptyset$
 - Filtering: $\pi = t$
 - Prediction: $\pi > t$
 - Hindsight: $\pi < t$

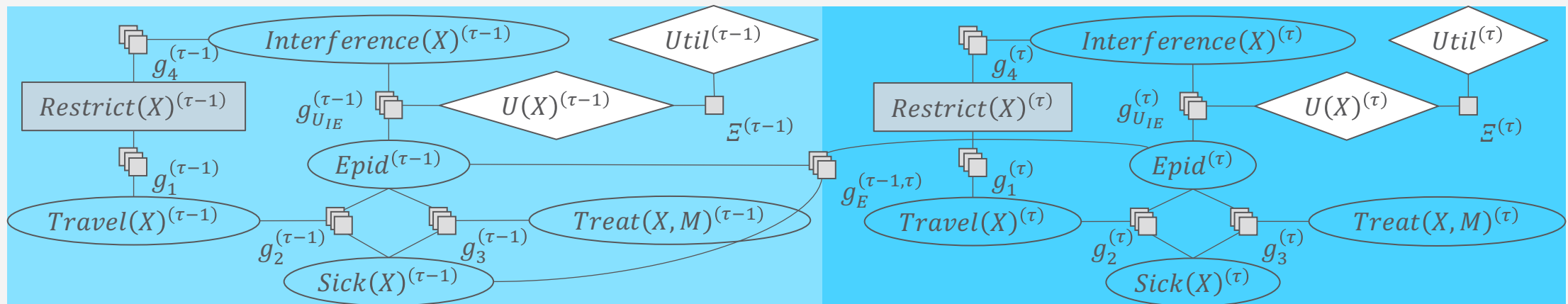
Inference Algorithm: LDJT

- Two FO jtrees $(J^0, J^\rightarrow)$ with interfaces for separation between past and present
- LJT as a subroutine for query answering
- Forward/backward messages to move in time



Decision Making over Time

- Basis: a sequential model ($G^0, G^\rightarrow$)
 - Describe behaviour over time using interslice parafactors
 - Within a slice, describe intra-slice (episodic) behaviour
- Extend intra-slice parts with decision + utility PRVs
 - Intra-slice behaviour described using a decision model
 - Inter-slice behaviour allows for predicting effect of decision on next step



A First Version of a Sequential Decision Model

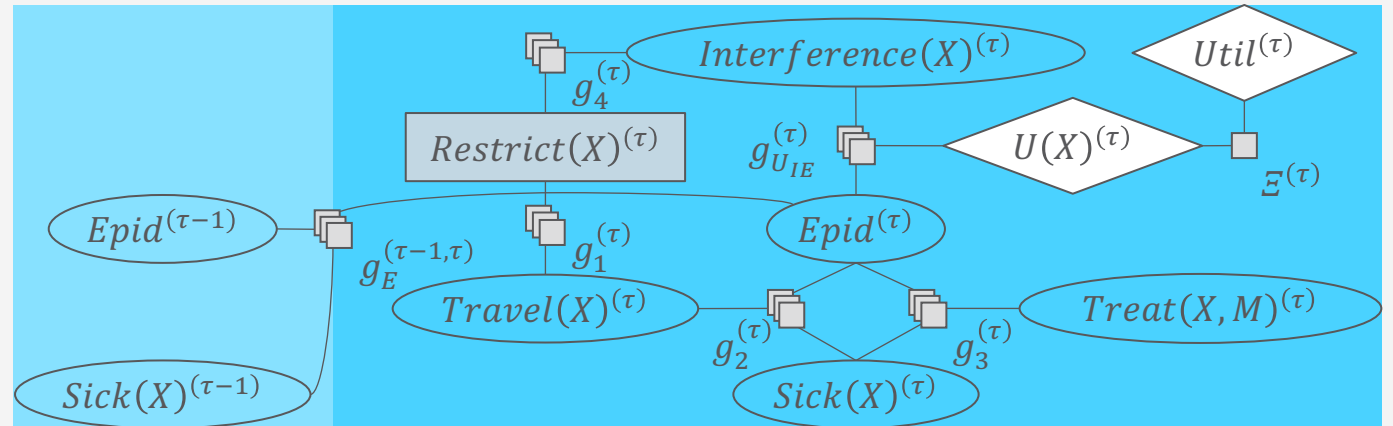
- $G = (G^0, G^\rightarrow)$ is a sequential model where $G^0, G^\rightarrow$ are decision models, i.e.,
 - G^0 is a decision model describing the intra-time slice behaviour for $t = 0$

$$G^0 = \left\{ g_i^{(0)} \right\}_{i=1}^{n_0} \cup \left\{ g_u^{(0)} \right\}_{u=1}^{m_0} \cup \{ E^{(0)} \}$$

- $G^\rightarrow$ is a decision model describing the intra- and inter-slice behaviour for $t > 0$

$$G^\rightarrow = \left\{ g_i^{(\tau)} \right\}_{i=1}^n \cup \left\{ g_u^{(\tau)} \right\}_{u=1}^m \cup \left\{ g_j^{(\tau-1,\tau)} \right\}_{j=1}^k \cup \{ E^{(\tau)} \}$$

- with
 - $g_i^{(t)}, g_j^{(\tau-1,\tau)}$ (potential) parfactors
 - $g_u^{(t)}$ utility parfactors
 - $E^{(t)}$ a combination function
 - $t \in \{0, \tau\}$



Assumptions about Actions

- Assumptions in terms of time / sequential behaviour...
 - Discrete steps
 - Markov-1
 - Stationary process
- ... lead to assumptions/ constraints on decisions
 - Actions can be carried out from one step to the next (no duration)
 - Effect/outcome of actions immediately captured in the next step
 - Actions do not affect the stationary process

Actions over Time

- Given a set of decision PRVs
 - $\mathbf{D}^{(\tau)} = \{D_1^{(\tau)}, \dots, D_n^{(\tau)}\}$
- Sequential sequence of decisions = **sequential action assignment**
 - Compound event for a sequence of steps from t_1 to t_2 , written as $\mathbf{d}^{(t_1:t_2)}$, i.e.,
$$\mathbf{d}^{(t_1:t_2)} = (\mathbf{d}^{(t_1)}, \mathbf{d}^{(t_1+1)}, \dots, \mathbf{d}^{(t_2-1)}, \mathbf{d}^{(t_2)})$$
 - $\mathbf{d}^{(t_i)} = \{D_1^{(t_i)} = d_1^{(t_i)}, \dots, D_n^{(t_i)} = d_n^{(t_i)}\}, t_i \in \{t_1, \dots, t_2\}$
- Decision making over time usually involves calculating the best sequence starting at current step t , making decisions for κ steps, given evidence up until t and previous decisions
 - I.e., finding the best $\mathbf{d}^{(t:t+\kappa)}$ given $\mathbf{e}^{(1:t)}$ and $\mathbf{d}^{(1:t-1)}$

So far, we used utilities to determine “the best”. How can we use them with time?

Utilities over Time

- One has to iterate over sequential action assignments and pick the one that yields the maximum expected utility
 - Each individual slice has a utility (or reward) → Multi-attribute utility theory
 - All individual utilities need to be combined into one utility of the complete sequence using a **sequential combination function Υ**
$$U(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots) = \Upsilon[U(\mathbf{r}^{(0)}), U(\mathbf{r}^{(1)}), U(\mathbf{r}^{(2)}), \dots]$$
- Assumption: Preference of one sequence over the other does not depend on time
 - **Preferences are stationary**
 - Formally, if two state sequences $(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots)$ and $(\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \dots)$ have the same starting state ($\mathbf{r}^{(0)} = \mathbf{s}^{(0)}$), then the two sequences $(\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots)$ and $(\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \dots)$ should be preference-ordered in the same way as $(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots)$ and $(\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \dots)$
 - Preference independence between the different slices
 - Use additive combination function (Υ)

Sequential Combination Functions

- General sequential combination function γ

$$U(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots) = \gamma[U(\mathbf{r}^{(0)}), U(\mathbf{r}^{(1)}), U(\mathbf{r}^{(2)}), \dots]$$

- Additive

- Sum over individual utilities

$$U(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots) = U(\mathbf{r}^{(0)}) + U(\mathbf{r}^{(1)}) + U(\mathbf{r}^{(2)}) + \dots$$

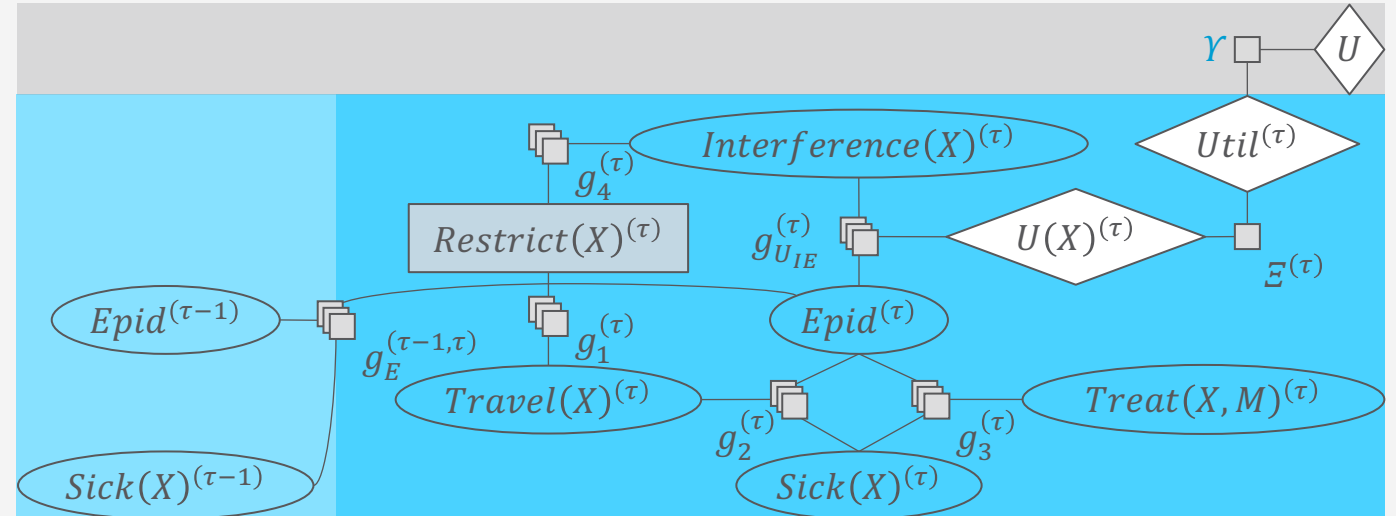
- Discounted

- Using a discount factor $\gamma \in [0,1]$
 - Reward now may be more important than one in t steps
 - If $\gamma = 1$: additive
 - Utility of a sequence = sum over discounted individual utilities

$$U(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots) = \gamma^0 \cdot U(\mathbf{r}^{(0)}) + \gamma^1 \cdot U(\mathbf{r}^{(1)}) + \gamma^2 \cdot U(\mathbf{r}^{(2)}) + \dots = \sum_{t=0} \gamma^t U(\mathbf{r}^{(t)})$$

Sequential Decision Model

- Sequential decision model $G = (G^0, G^{\rightarrow}, \gamma)$ is given by
 - Decision model G^0 describing the intra-time slice behaviour for $t = 0$ (as before)
 - Decision model $G^{\rightarrow}$ describing the intra- and inter-slice behaviour for $t > 0$ (as before)
 - Sequential combination function γ such as (discounted) addition
- Semantics given by unrolling G for T steps to form an episodic decision model as defined before
 - Ground to get a propositional model

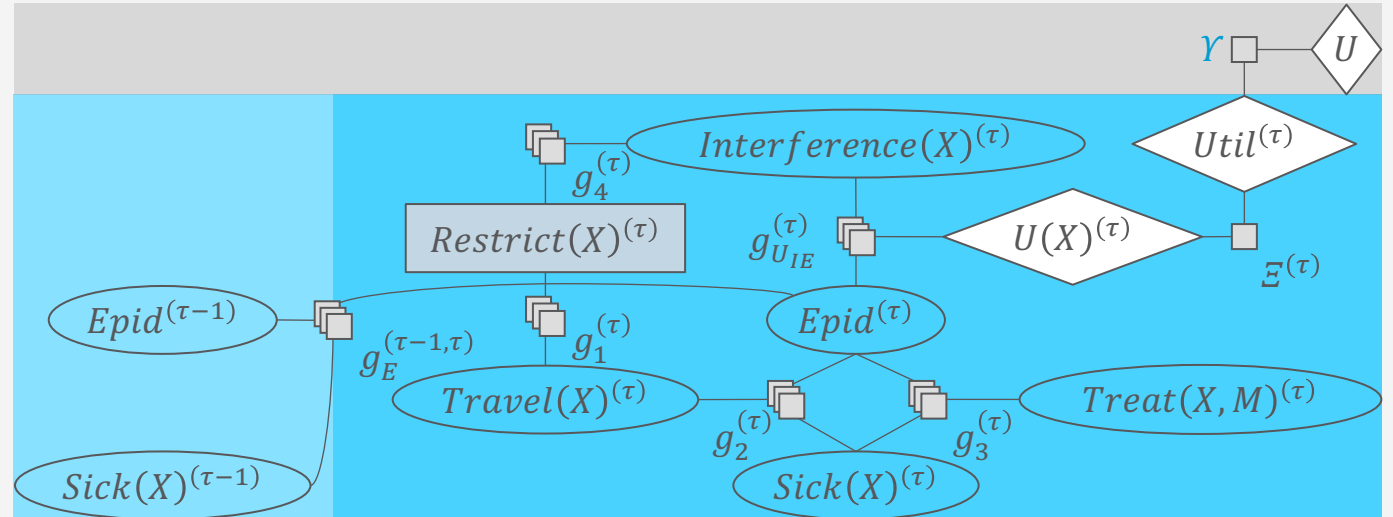


Sequential EU Query

- Sequential EU query at a current step t with a horizon κ in a sequential decision model G
 $eu(e^{(1:t)}, d^{(1:t-1)}, d^{(t:t+\kappa)})$

$$= \sum_{v^{(t:t+\kappa)} \in \text{ran}(rv(G_U^{(t:t+\kappa)}))} P(v^{(t:t+\kappa)} | e^{(1:t)}, d^{(1:t-1)}, d^{(t:t+\kappa)}) \sum_{k=0}^{\kappa} \gamma^k \Xi^{(t+k)}(G_U^{(t+k)})$$

- Additive/discounted combination (γ)
- In terms of semantics, unroll G for $t + \kappa$ steps and answer an “episodic” EU query over “future” decisions $d^{(t:t+\kappa)}$ given observations $e^{(1:t)}$ and “past” decisions $d^{(1:t-1)}$ as evidence
 - Observation: $G_U^{(1:t-1)}$ irrelevant at t

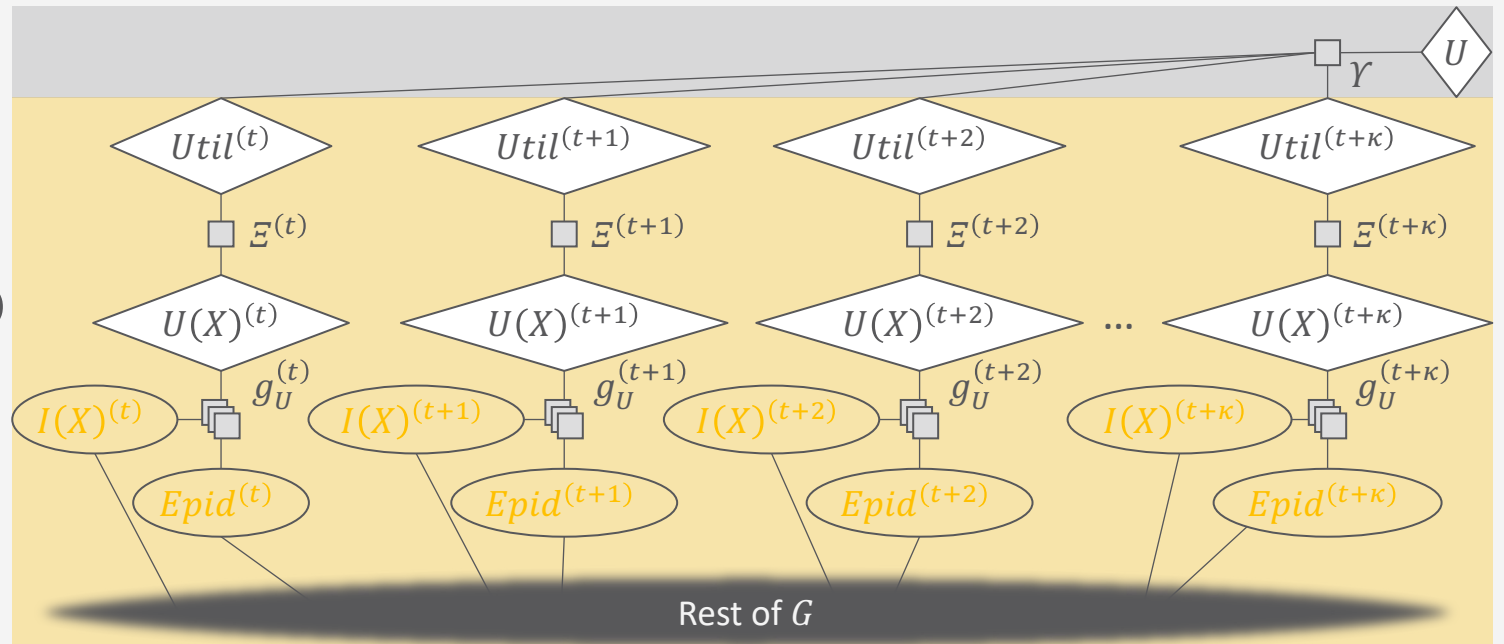


EU Query over Time: Exact

$$eu(e^{(1:t)}, d^{(1:t-1)}, d^{(t:t+\kappa)}) = \sum_{v^{(t:t+\kappa)} \in \text{ran}(rv(G_U^{(t:t+\kappa)}))} P(v^{(t:t+\kappa)} | e^{(1:t)}, d^{(1:t-1)}, d^{(t:t+\kappa)}) \sum_{k=0}^{\kappa} \gamma^k \Xi^{(t+k)}(G_U^{(t+k)})$$

- Problem with answering an EU query exactly: Query terms from all slices $t : t + \kappa$ involved
 - Unlikely that query terms are independent
 - Even assuming current model $G^{(t)}$ independent from the past with $e^{(1:t-1)}, d^{(1:t-1)}$ set, still need to unroll model for κ steps

Realistically not computable!



EU Query over Time: Approximate

- In episodic model, assume independence between PRVs of utility parfactored given $\mathbf{e}, \mathbf{d}$ to factorise query
 - Set of smaller queries instead of one large
 - Exact answer if independences hold
- Does that help with sequential models?
 - Yes and no.
 - Assuming independence allows for factorising query into queries for each slice
 - BUT: Query answer will be approximate as independence between slices not reasonable in a sequential model just given $\mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t-1)}$
 - Otherwise, a set of episodic models suffice

EU Query over Time: Approximate

- $eu(\mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t-1)}, \mathbf{d}^{(t:t+\kappa)})$ when assuming the inter-slice independence:

$$\begin{aligned}
 & eu(\mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t-1)}, \mathbf{d}^{(t:t+\kappa)}) \\
 &= \sum_{\mathbf{v}^{(t:t+\kappa)} \in \text{ran}(rv(G_U^{(t:t+\kappa)}))} P(\mathbf{v}^{(t:t+\kappa)} | \mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t-1)}, \mathbf{d}^{(t:t+\kappa)}) \sum_{k=0}^{\kappa} \gamma^k \cdot \overbrace{\mathbb{E}^{(t+k)} \left[\phi_{U_1}^{(t+k)}(\mathbf{r}_1^{(t+k)}), \dots, \phi_{U_m}^{(t+k)}(\mathbf{r}_m^{(t+k)}) \right]}^{\mathbf{r}_u^{(t+k)} = \pi_{R_u}(\mathbf{v}^{(t:t+\kappa)})} \\
 &= \sum_{\mathbf{v}^{(t:t+\kappa)} \in \text{ran}(rv(G_U^{(t:t+\kappa)}))} \prod_{k'=0}^{\kappa} P(\mathbf{v}^{(t+k')} | \mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t-1)}, \mathbf{d}^{(t:t+k')}) \sum_{k=0}^{\kappa} \gamma^k \cdot \mathbb{E}^{(t+k)} \left[\phi_{U_1}^{(t+k)}(\mathbf{r}_1^{(t+k)}), \dots, \phi_{U_m}^{(t+k)}(\mathbf{r}_m^{(t+k)}) \right] \\
 &= \sum_{k=0}^{\kappa} \gamma^k \sum_{\mathbf{v}^{(t+k)} \in \text{ran}(rv(G_U^{(t+k)}))} P(\mathbf{v}^{(t+k)} | \mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t-1)}, \mathbf{d}^{(t:t+k)}) \cdot \mathbb{E} \left[\phi_{U_1}^{(t+k)}(\mathbf{r}_1^{(t+k)}), \dots, \phi_{U_m}^{(t+k)}(\mathbf{r}_m^{(t+k)}) \right] \\
 &= \sum_{k=0}^{\kappa} \gamma^k \cdot eu(\mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t+k-1)}, \mathbf{d}^{(t+k)}) \quad \text{One EU query per step}
 \end{aligned}$$

EU Query over Time: Approximate

- Solving a sequential EU query reduces to solving individual EU queries per step

$$eu(\mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t-1)}, \mathbf{d}^{(t:t+\kappa)}) = \sum_{k=0}^{\kappa} \gamma^k \cdot eu(\mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t+k-1)}, \mathbf{d}^{(t+k)})$$

- Solve individual EU queries based on intra-slice assumptions
 - E.g., assuming additive combination function or independence between intra-slice utility parfactors

$$\begin{aligned} & eu(\mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t-1)}, \mathbf{d}^{(t:t+\kappa)}) \\ &= \sum_{k=0}^{\kappa} \gamma^k \sum_{\mathbf{v}^{(t+k)} \in \text{ran}(rv(G_U^{(t+k)}))} P(\mathbf{v}^{(t+k)} | \mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t-1)}, \mathbf{d}^{(t:t+\kappa)}) \sum_{u=1}^m \phi_u^{(t+k)}(\mathbf{r}_u^{(t+k)}) \\ &= \sum_{k=0}^{\kappa} \gamma^k \sum_{u=1}^m \sum_{\mathbf{r}_u^{(t+k)} \in \text{ran}(rv(g_u^{(t+k)}))} \phi_u^{(t+k)}(\mathbf{r}_u^{(t+k)}) P(\mathbf{r}_u^{(t+k)} | \mathbf{e}^{(1:t)}, \mathbf{d}^{(1:t-1)}, \mathbf{d}^{(t:t+\kappa)}) \end{aligned}$$

Sequential MEU Problem

- Given a sequential decision model G , evidence $\mathbf{e}^{(0:t)}$, previous decisions $\mathbf{d}^{(0:t-1)}$, and a horizon (integer) κ
- Sequential MEU problem
 - Find the sequential action assignment that yields the highest expected utility in G
 - Formally,

$$MEU(\mathbf{e}^{(0:t)}, \mathbf{d}^{(0:t-1)}, \kappa) = \left(\mathbf{d}^*, eu(\mathbf{e}^{(0:t)}, \mathbf{d}^{(0:t-1)}, \mathbf{d}^*) \right)$$

$$\mathbf{d}^* = \arg \max_{\mathbf{d}^{(t:t+\kappa)} \in \text{ran}(\mathbf{D}^{(t:t+\kappa)})} eu(\mathbf{e}^{(0:t)}, \mathbf{d}^{(0:t-1)}, \mathbf{d}^{(t:t+\kappa)})$$

- Size of $\text{ran}(\mathbf{D}^{(t:t+\kappa)})$ exponential in number of groups (as with episodic decision making) and κ

Solving a Sequential MEU Problem: MEU-LDJT

- Given a sequential decision model G , evidence $\mathbf{e}^{(0:t)}$, previous decisions $\mathbf{d}^{(0:t-1)}$, a horizon (integer) κ , and incoming filtering, prediction, hindsight queries Q
- Procedure:
 - Construct FO jtrees $J^0, J^\rightarrow$
 - For $t = 0, \dots$,
 - Instantiate $J^{(t)}$, add $\alpha^{(t-1)}$, enter $\mathbf{e}^{(t)}$, pass messages, answer Q
 - MEU loop: For each possible sequential action assignment $\mathbf{d}^{(t:t+\kappa)}$
 - $eu \leftarrow$ Calculate EU query, exactly or approximately
 - if $eu > eu^*$, then store $\mathbf{d}^{(t:t+\kappa)}$ in $\mathbf{d}^*$, setting $eu^* \leftarrow eu$
 - Output $\mathbf{d}^*$ or act out $\mathbf{d}^*$: Set $\mathbf{d}^{(t)}$ (and send $\mathbf{d}^{(t)}$ to an execution platform)
 - Calculate $\alpha^{(t)}$

Calculates MEU problem anew in each step; adapt such that $\mathbf{d}^*$ is acted out for κ steps, or until following $\mathbf{d}^*$ no longer appears safe

- Store $\mathbf{d}^*$ and, instead of going into the MEU loop, set the next $\mathbf{d}^{(t)}$ from $\mathbf{d}^*$

Solving a Sequential MEU Problem: MEU-LDJT

- Calculating a sequential EU query for $\mathbf{d}^{(t:t+\kappa)}$ in a sequential FO jtree
 - Exact:
 - Unroll FO jtree for κ steps, set $\mathbf{d}^{(t:t+\kappa)}$, compute probability query over $\text{rv}\left(G_U^{(t:t+\kappa)}\right)$, calculate expected utility
 - Approximate:
 - Overall expected utility $U \leftarrow 0$
 - For $k = 0, \dots, \kappa$
 - Instantiate FO jtree $J^{(t+k)}$, add $\alpha^{(t+k-1)}$ (if not already done)
 - Set $\mathbf{d}^{(t+k)}$, pass (or update) messages, compute probability query over $\text{rv}\left(G_U^{(t+k)}\right)$, calculate expected utility for $t+k$ (set of EU queries if assuming intra-slice independence), add result to U
 - Calculate $\alpha^{(t+k)}$

Sequential MEU Problem: Example

- Given no evidence, current time step $t = 2$, and $\kappa = 2$

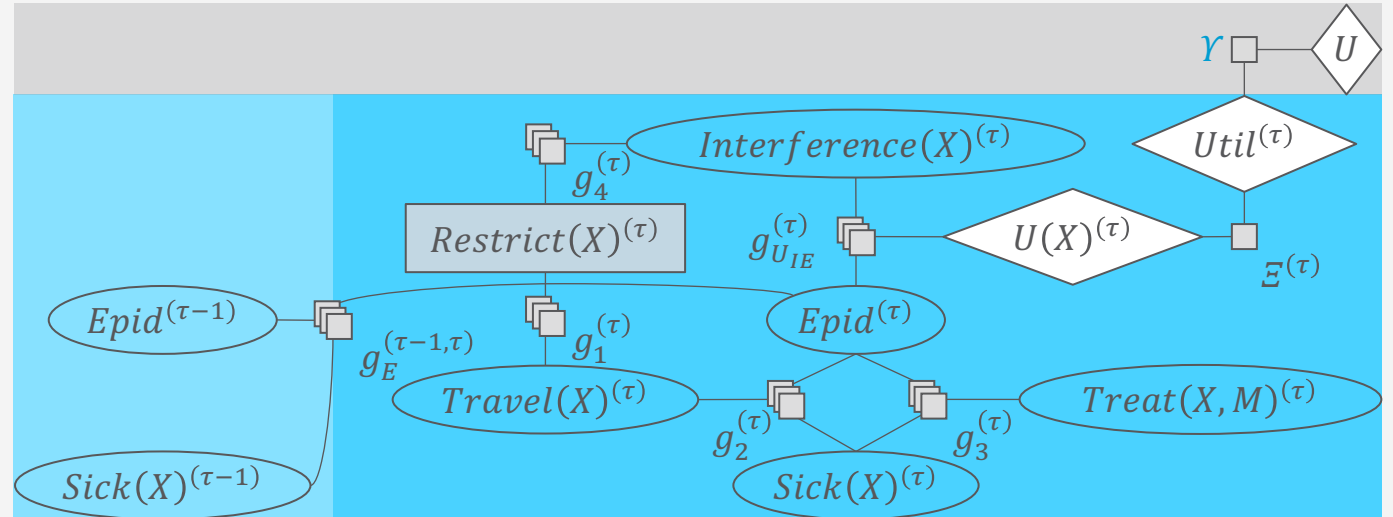
$$d^* = \arg \max_{i \in \{1, \dots, 8\}} eu(d_i^{(2:4)})$$

- Test **eight** sequential action assignments

- For each $d_i^{(2:4)}$
 - Set $d_i^{(2:4)}$ in unrolled FO jtree $J^{(2:4)}$ and answer EU query $eu(d_i^{(2:4)})$ or
 - Set $d_i^{(2:4)}$ incrementally in $J^{(t+k)}$ and add up EU queries $eu(d_i^{(t+k)})$
- Output $\arg \max d_i^{(2:4)}$
 - Or act out $d_i^{(2)}$

	$d^{(2)}$	$d^{(3)}$	$d^{(4)}$
$d_1^{(2:4)}$	ban	ban	ban
$d_2^{(2:4)}$	ban	ban	free
$d_3^{(2:4)}$	ban	free	ban
$d_4^{(2:4)}$	ban	free	free
$d_5^{(2:4)}$	free	ban	ban
$d_6^{(2:4)}$	free	ban	free
$d_7^{(2:4)}$	free	free	ban
$d_8^{(2:4)}$	free	free	free

Decisions



Acting

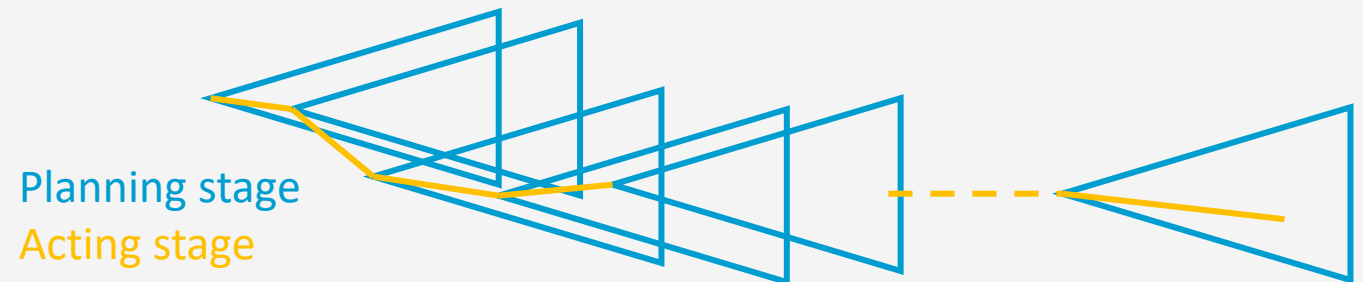
- After computing a sequential action assignment, assignment is acted out, including:
 - Actuators get commands to carry out the action behind a current assignment,
 - Internal state is updated: Decision PRVs are set, (messages are passed,) and then time moves on
- Agent can then continue to act according to the sequential assignment or recalculate as new evidence comes in
 - Recalculate once every $k \leq \kappa$ steps

Using Decision Making in Acting

- Receding horizon:
 - Call meu, obtain a sequential action assignment π , perform 1st action, call meu again ...
 - meu refers to an implementation solving an MEU problem
 - Like game-tree search (chess, checkers, etc.)
- Useful when unpredictable things are likely to happen
 - Re-plans immediately
- Potential problem:
 - May pause repeatedly while waiting for meu to return

Run-MEU (G, κ)

```
while  $e \leftarrow$  new evidence do  
  absorb  $e$  in  $G$   
   $\pi \leftarrow$  meu( $G, \kappa$ )  
   $d \leftarrow$  pop-first( $\pi$ )  
  perform  $d$   
  update  $G$  with  $d$ 
```

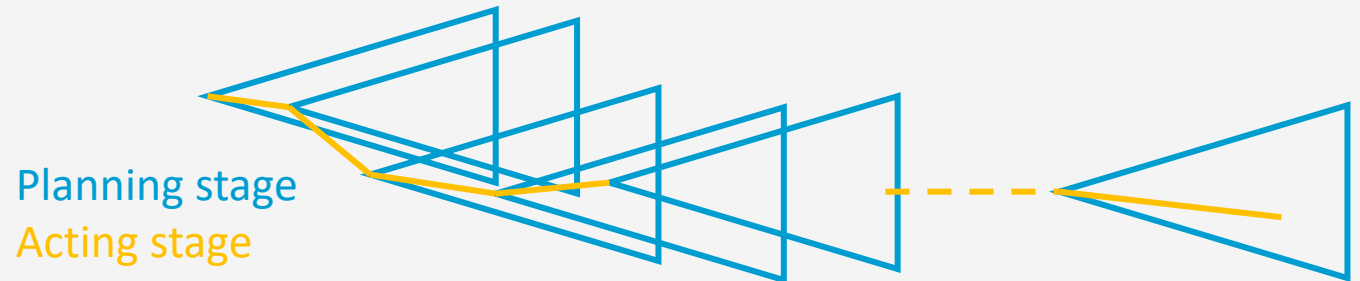


Using Decision Making in Acting

- Lazy look-ahead:
 - Call `meu`, execute the sequential action assignment as far as possible, do not call `meu` again unless necessary
 - Simulate tests whether the assignment will execute correctly
 - Lower-level refinement, physics-based simulation, prediction accuracy < some threshold
- Potential problems
 - May might miss opportunities to replace π with a better assignment
 - Without `Simulate`, may not detect problems until it is too late

```

Run-Lazy-MEU( $G, \kappa$ )
  while  $e \leftarrow$  new evidence do
    absorb  $e$  in  $G$ 
     $\pi \leftarrow \text{meu}(G, \kappa)$ 
    while  $\pi \neq ()$  and
      Simulate( $G, \pi$ )  $\neq$  failure do
       $d \leftarrow \text{pop-first}(\pi)$ 
      perform  $d$ 
      update  $G$  with  $d$ 
  
```



Using Decision Making in Acting

- May detect opportunities earlier than Run-Lazy-MEU
 - But may miss some that Run-MEU would find
- Without Simulate, may fail to detect problems until it is too late
 - Not as bad at this as Run-Lazy-MEU

```
Run-Concurrent-MEU ( $G, \kappa$ )  
   $\pi \leftarrow \langle \rangle$   
  // thread 1 + 2 run concurrently  
thread 1:  
  while  $e \leftarrow$  new evidence do  
    absorb  $e$  in  $G$   
     $\pi \leftarrow \text{meu}(G, \kappa)$   
thread 2:  
  while  $e \leftarrow$  new evidence do  
    absorb  $e$  in  $G$   
    if  $\pi \neq ()$  and  
      Simulate( $G, \pi$ )  $\neq$  failure then  
       $d \leftarrow \text{pop-first}(\pi)$   
      perform  $d$   
      update  $G$  with  $d$ 
```

Offline vs. Online Decision Making

- Online decision making: Depends on how far one looks into the future
 - Can extend the look-ahead further and further, propagating future effects back to present to see if the current best decision still holds
 - Extending the look-ahead to infinity, one will eventually reach a fix point: *offline decision making*
- Offline decision making
 - Solving a partially observable Markov decision process (POMDP) – Ch. 17 in Russell/Norvig's AIMA3
 - Basically find the steady state in terms of actions that leads to the maximum expected utility
 - Look-up table, fast during acting, huge overhead beforehand
 - Reacting to extreme situations/evidence no possible
 - In basic form, no longer factorised model but a transition function over complete state space
 - Part of the lecture: *Automated Planning and Acting* (next winter term)

Interim Summary

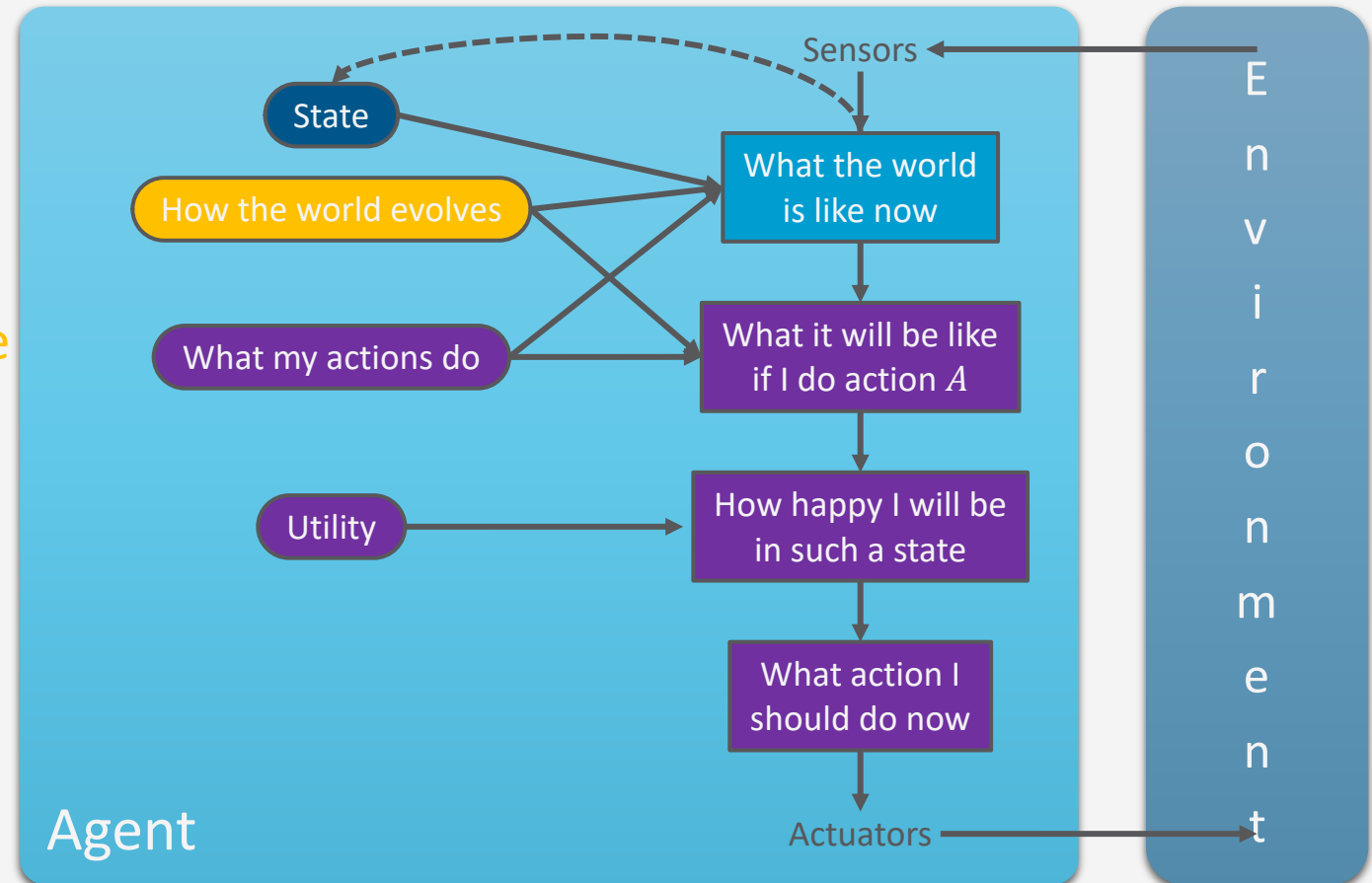
- Sequential decision models as a combination of
 - Sequential models for temporal / sequential aspect
 - Decision models for decision making
- Sequential expected utilities
 - Additive or discounted utilities
 - Exact solution virtually impossible to compute
 - Approximation by summing over EU query result for each step
- Inference based on LDJT
 - Proceed in time to answer EU queries part of a sequential EU query
- Acting
 - Immediate look-ahead, lazy run until failed simulation or sequence acted out, both concurrently

Contents in this Lecture Related to *Utility-based Agents*

- Further topics
 3. (Episodic) PRMs
 4. Lifted inference (in episodic PRMs)
 5. Lifted learning (of episodic PRMs)
 6. Lifted sequential PRMs and inference
 7. Lifted decision making

Sequential Decision Models

- Uncertainty modelled by probabilities
- Relational aspect using logical variables
- Temporal aspect by time indexing
- Decisions and effects by actions & utilities in a sequential model



Outline: 6. Decision-theoretic Models and Inference

a. Utility Theory

- Preferences, maximum expected utility (MEU) principle
- Utility function, multi-attribute utility theory

b. Static Decision Making

- Modelling, semantics, inference tasks
- Inference algorithm: LVE for MEU as an example

c. Sequential Decision Making

- Modelling, semantics, sequential MEU problem
- Inference algorithm: LDJT for MEU as an example
- Acting

The End