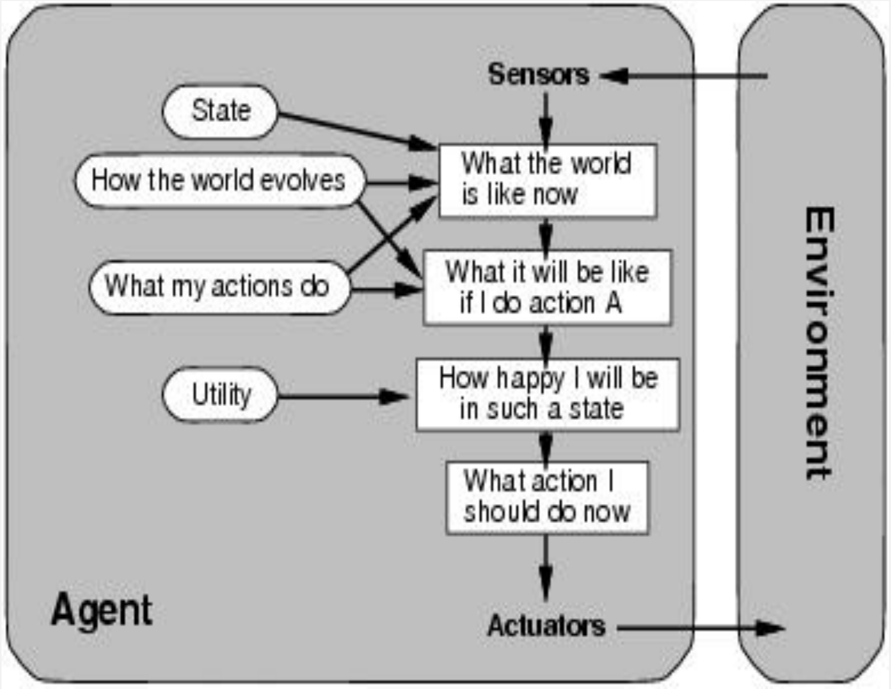
Algorithmic Abstractions of Rational Decision-Making

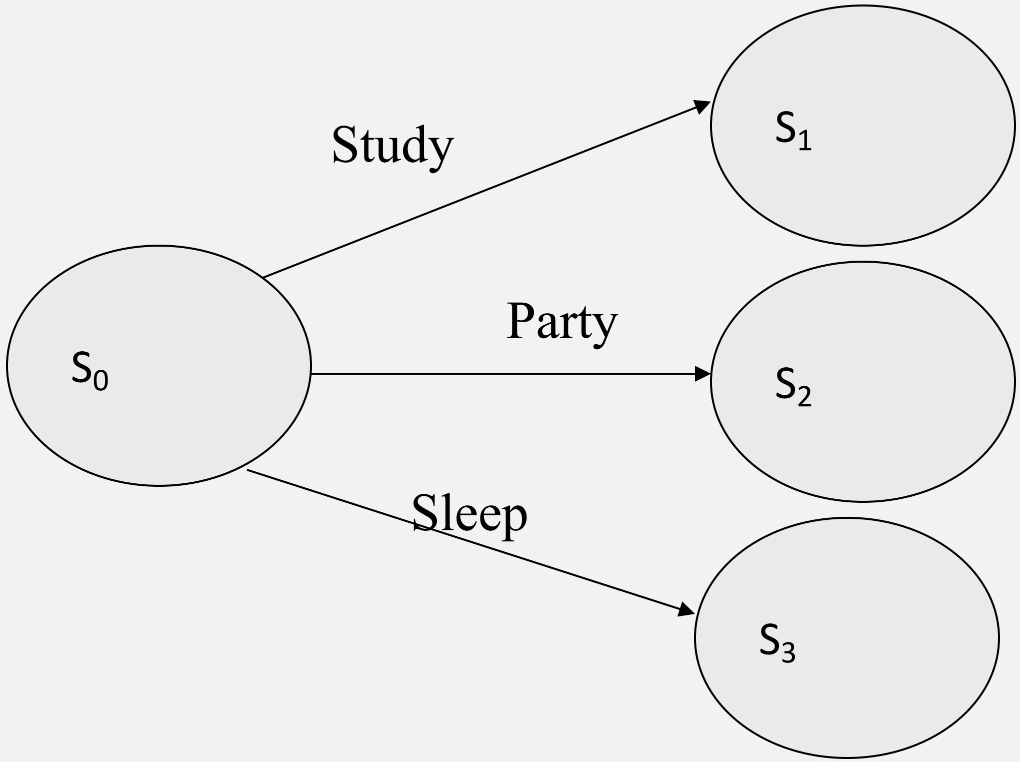
Decision-Theoretic Agent



- Different states have different utility to the agent
- Utility function of a simple decision theoretic agent maps each state onto a real number (utility)
- Actions are chosen based on the expected utility of the resulting state
- More general setting involves making complex decisions involving sequences of actions

Making Simple Decisions - Example

Decision time for Agent Joe Six Pack



Combining Beliefs and Desires

- Rational Behavior
  - Based on beliefs about the world, in particular, consequences of one's actions in a given state
  - Must cope with uncertainty
    - There is no way to know for sure the outcome of an action because of partial ignorance or inherently stochastic effects of actions
  - Must provide a means of comparing alternatives using a common currency
    - Partying on a Thursday night versus getting an A

Should Bob buy insurance?

- Bob is contemplating whether he should take insurance on a shipment from Amsterdam to St. Petersburg.
- If the ship does not encounter a storm, the shipment arrives on time in St. Petersburg, and Bob will earn 10,000 rubles
- If the ship encounters a storm, the shipment will be delayed and Bob will earn only 8000 rubles.
- The Amsterdam underwriters want Bob to pay 1000 rubles for a full coverage insurance policy
- Should Bob buy the policy?

Should Bob buy insurance?

- Potential loss to Bob in the event of delayed arrival of shipment = 2000 rubles
- Cost of insurance policy = 1000 rubles
- Should Bob buy the policy?

Should Bob buy insurance?

- Depends
- On what?
  - Probability of storm-related delay
  - If the storm is highly unlikely (say probability of storm $\approx$ 0.2) at that time of the year, perhaps not
  - If the storm is likely (say probability of storm $\approx$ 0.8) then perhaps
  - Can we translate this intuition into a precise prescription for decision making under uncertainty?

Should Bob buy insurance?

- Suppose Bob believes that the probability of a storm related delay $\approx$ 0.2
- Bob's expected earnings in the absence of insurance

  = (0.2)(8000) + 0.8 (10,000)

  = 1600 + 8000

  = 9600 rubles
- Bob's expected earnings if he purchases insurance

  = 10,000 – 1000

  = 9000 rubles
- Bob is perhaps better off without insurance than with it.

Should Bob buy insurance?

- Suppose Bob believes that the probability of a storm related delay $\approx$ 0.8
- Bob's expected earnings in the absence of insurance

  = (0.8)(8000) + 0.2 (10,000)

  = 6400 + 2000

  = 8400 rubles
- Bob's expected earnings if he purchases insurance

  = 10,000 – 1000

  = 9000 rubles
- Bob is better off with insurance than without it.

St. Petersburg Paradox

- From Nicolas Bernoulli's letter
- Consider the following game
  - Peter flips a fair coin repeatedly until a head shows up and will give Paul:
    - \$2 if the first *head* shows up on the 1st flip
    - $\$2^2$ if the first *head* shows up on the 2nd flip
    - ….
    - $\$2^k$ if the first head shows up on the k^{th} flip
  - How much should Paul pay Peter to play this game?

St. Petersburg Paradox (cont)

$$E(payoff) = \sum_{k=1}^{k=\infty} 2^k \left(\frac{1}{2}\right)^k = 1 + 1 + \cdots = \infty$$

- The expected payoff is infinite
- Does this mean it is rational for Paul to pay Peter any finite amount (say $1 million) to play this game?

St Petersburg Paradox

- How much should Paul be willing to pay Peter for a chance to play the game?
- Expected payoff = infinity
- But.. There is a risk of loss
  - Suppose Paul pays $1,000,000 to pay the game
  - Suppose the first head shows up on the 2nd toss
  - Paul will receive $4 and lose $999996
- Is the gamble worth the risk?
- Depends..

St Petersburg paradox

- Is Paul's gamble worth the risk of losing almost $1,000,000?
- Depends
- On what?
  - How much money Paul has to start with
  - The risk might be unacceptable if Paul's entire life savings is $1,000,000
  - The risk might be perfectly reasonable if Paul has billions in the bank
  - If Paul is poor, he may be justified in paying no more than two dollars, the minimum possible pay-off of the game

Making simple decisions

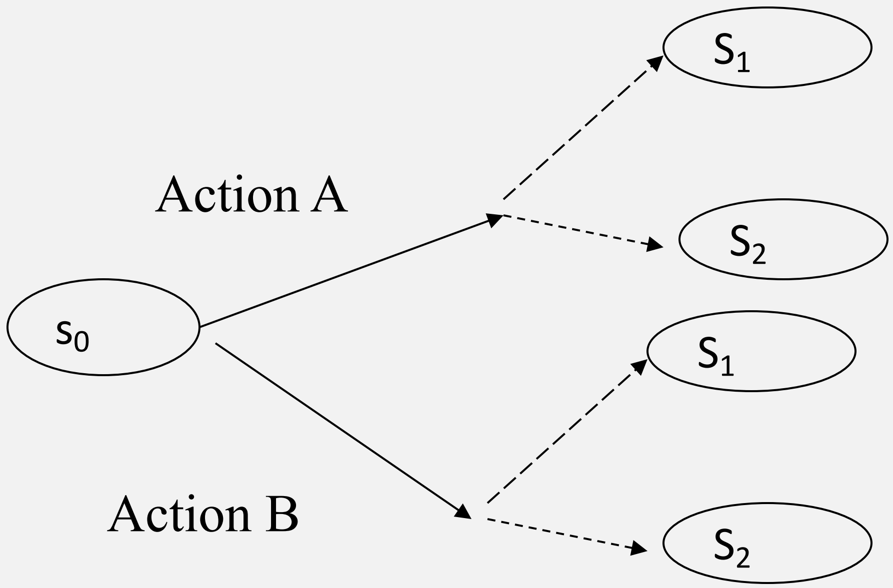
- Can we turn the previous intuition into a recipe for decision making under uncertainty?
- Von Neumann – Morgenstern solution
    - **M**aximum **E**xpected **U**tility (MEU) principle
        - Choose actions that maximize expected utility of outcome
    - As evident from the St. Petersburg paradox, for most people, the *utility* of money is not a linear function of the amount of money

Making simple decisions

- Notation
  - $U(S)$: utility of state S
  - S : snapshot of the world
  - A : action of the agent
  - $Result_i(A)$: ith outcome of (state resulting from doing) A
  - E : available evidence
  - $Do(A)$: executing A in current state S

Making simple decisions

- Utility function
  - assigns a single number to each outcome
  - models the desirability of the state to an agent
  - combined with probability of each outcome resulting from an action yields expected utility for action leading to each outcome



Making Decisions

- Expected Utility

$$EU(A \mid E) = \sum_i P(Result_i(A) \mid E, Do(A))\, U(Result_i(A))$$

- Maximum Expected Utility(MEU)
  - Choose an action which maximizes agent's expected utility
- Computing $P(Result_i(A) \mid E, Do(A))$ requires a probabilistic model of the world (Bayes Network)
- Computing the utility of a state $U(Result_i(A))$ may require search because it can be hard to tell how good a state is until we know where it would lead us

Decision Theoretic Agent

function DT-AGENT(*percept*) **returns** an *action*
  static: *belief_state*, probabilistic beliefs about the current state of the world
      action, the agent's action

  update *belief_state* based on *action* and *percept*
  calculate outcome probabilities for actions,
    given action descriptions and current *belief_state*
  select *action* with highest expected utility
    given probabilities of outcomes and utility information
  return *action*

Lotteries

Lotteries are used to model decision making scenarios

Lotteries have a finite set of possible mutually exclusive outcomes and probabilities associated with each outcome

Simple Lotteries : $L = [p, A; 1-p, B]$ (two outcomes)

$L = [p_1, A_1;\ p_2, A_2; \ldots\ p_n, A_n]$ (n outcomes)

$L = [1, A] = A$ (1 outcome)

Compound lotteries: outcomes are themselves lotteries

$$L_1 = [p, A; 1-p, L_2]$$

$$L_2 = [p_1, A; p_2, B; p_3, C]$$

Preferences

$A \succ B$: *A* is preferred to *B*

$A \sim B$: indifference between *A* & *B*

$A \succeq B$: *A* is preferred to *B* or there is indifference between *A* and *B*

In general, outcomes such as *A, B* can be simple outcomes or outcomes of lotteries $L = [p, A; 1-p, B]$

Constraints on Rational Preferences I

Rational preferences must satisfy "reasonable" constraints on rational behavior

- Orderability $(A \succ B) \oplus (B \succ A) \oplus (A \sim B)$

  $\oplus$ denotes exclusive OR

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

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

Constraints on Rational Preferences II

- Substitutability

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

- Monotonicity

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

$$A \sim B \Rightarrow (p = q \Leftrightarrow [p, A; 1-p, B] \sim [q, A; 1-q, B]$$

- Decomposability

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

Utility Function

- Utility Function models an agent's preferences
- Utility principle

$$U(A) > U(B) \Leftrightarrow A \succ B$$

$$U(A) = U(B) \Leftrightarrow A \sim B$$

- Expected Utility of a lottery

$$U([p_1, x_1; \ldots; p_n, x_n]) = \sum_i p_i U(x_i)$$

- Utility principle applies to lotteries

$$[p_1, x_1; \ldots; p_n, x_n] \succ [q_1, x_1; \ldots; q_n, x_n] \Leftrightarrow \sum_i p_i U(x_i) > \sum_i q_i U(x_i)$$

Existence of an utility function I

Consider a simple lottery $l = [p_1, x_1;\ p_2, x_2; ..\ p_r, x_r]$

where $x_1 \succ x_2 \succ \cdots x_{r-1} \succ x_r$

The continuity axiom guarantees the existence of u_i

that ensures indifference between each prize x_i

and a reference lottery $[u_i, x_1;\ (1 - u_i), x_r]$

Hence, we have

$$l = [p_1, x_1;\ p_2, x_2;\ ..\ p_r, x_r]$$

$$\sim [p_1, [u_1, x_1;\ (1 - u_1), x_r]; p_2, [u_2, x_1;\ (1 - u_2), x_r] .. p_r, [u_r, x_1;\ (1 - u_r), x_r]]$$

Existence of an utility function II

Note that each reference lottery $\left[u_i, x_1;\ \left(1-u_i\right), x_r\right]$

is equivalent to a simple lottery:

$$\left[u_i, x_1; 0, x_2; 0, x_3; ..0, x_{r-1}; \left(1-u_i\right), x_r\right]$$

Using the decomposability axiom, we have:

$$l = \left[p_1, x_1;\ p_2, x_2; ..\ p_r, x_r\right]$$

$$\sim \left[p_1, \left[u_1, x_1; \left(1\text{-}u_1\right), x_r\right]; p_2, \left[u_2, x_1; \left(1\text{-}u_2\right), x_r\right]..p_r, \left[u_r, x_1; \left(1\text{-}u_r\right), x_r\right]\right]$$

$$\sim \left[\left(p_1u_1 + p_2u_2 + .. + p_ru_r\right), x_1; \left(p_1\left(1-u_1\right) + p_2\left(1-u_2\right) + .. + p_r\left(1-u_r\right)\right), x_r\right]$$

$$\sim \left[\left(\sum_{i=1}^{r} p_iu_i\right), x_1; \left(\sum_{i=1}^{r} p_i\left(1-u_i\right)\right), x_r\right]$$

Existence of an utility function III

We showed

$$l = [p_1, x_1;\ p_2, x_2; \ldots p_r, x_r] \sim \left[\left(\sum_{i=1}^{r} p_i u_i\right), x_1; \left(\sum_{i=1}^{r} p_i (1-u_i)\right), x_r\right]$$

Similarly,

$$l' = [q_1, x_1;\ q_2, x_2; \ldots q_r, x_r] \sim \left[\left(\sum_{i=1}^{r} q_i u_i\right), x_1; \left(\sum_{i=1}^{r} q_i (1-u_i)\right), x_r\right]$$

Existence of an utility function IV

Orderability and monotonicity axioms imply

$$l \succ l' \Leftrightarrow \left[\left(\sum_{i=1}^{r} p_i u_i\right), x_1; \left(\sum_{i=1}^{r} p_i (1-u_i)\right), x_r\right] \succ \left[\left(\sum_{i=1}^{r} q_i u_i\right), x_1; \left(\sum_{i=1}^{r} q_i (1-u_i)\right), x_r\right]$$

$$\Leftrightarrow \sum_{i=1}^{r} p_i u_i > \sum_{i=1}^{r} q_i u_i$$

$$l \sim l' \Leftrightarrow \left[\left(\sum_{i=1}^{r} p_i u_i\right), x_1; \left(\sum_{i=1}^{r} p_i (1-u_i)\right), x_r\right] \sim \left[\left(\sum_{i=1}^{r} q_i u_i\right), x_1; \left(\sum_{i=1}^{r} q_i (1-u_i)\right), x_r\right]$$

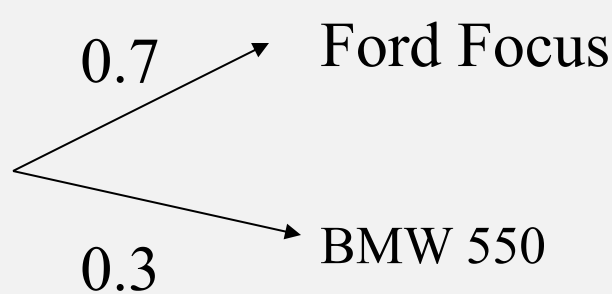
$$\Leftrightarrow \sum_{i=1}^{r} p_i u_i = \sum_{i=1}^{r} q_i u_i$$

Setting $u_i = U(x_i)$, we have established that the existence of a utility function follows from the constraints on rational preferences

Eliciting Utility Function

Need to map states or outcomes to real numbers

- Compare *A* to standard lottery L_p
  - utility of the best possible prize with probability *p*
    - $u_\top = 1$:
  - utility of the worst possible catastrophe with prob. *1-p*
    - $u_\bot = 0$
- Adjust *p* until $A \sim L_p$ $U(A) = p$
- Toyota Camry ~ *L*

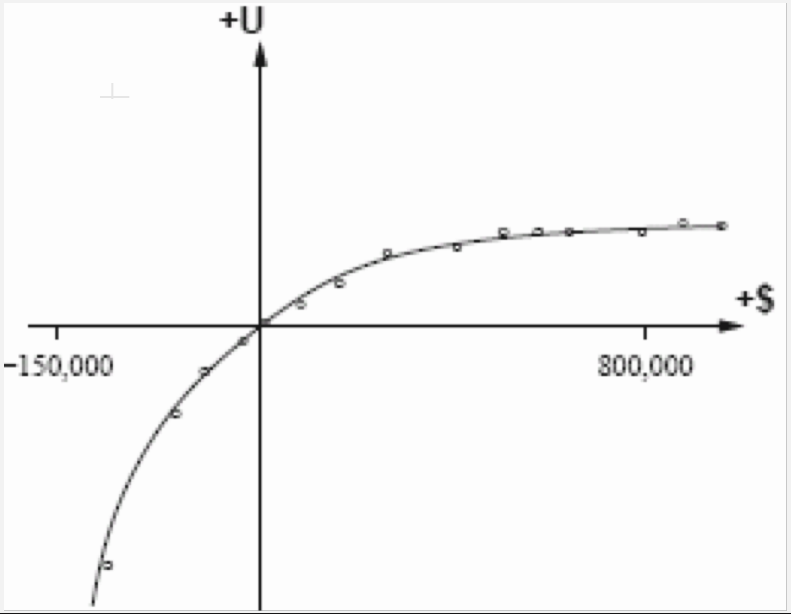


Some remarks on utility functions

- Constraints on rational preferences do not guarantee a unique utility function
  - positive linear transformation will not change preferences

$$U'(x) = k_1 U(x) + k_2 \quad where \quad k_1 > 0$$

- Utility of money is typically not a linear function of money



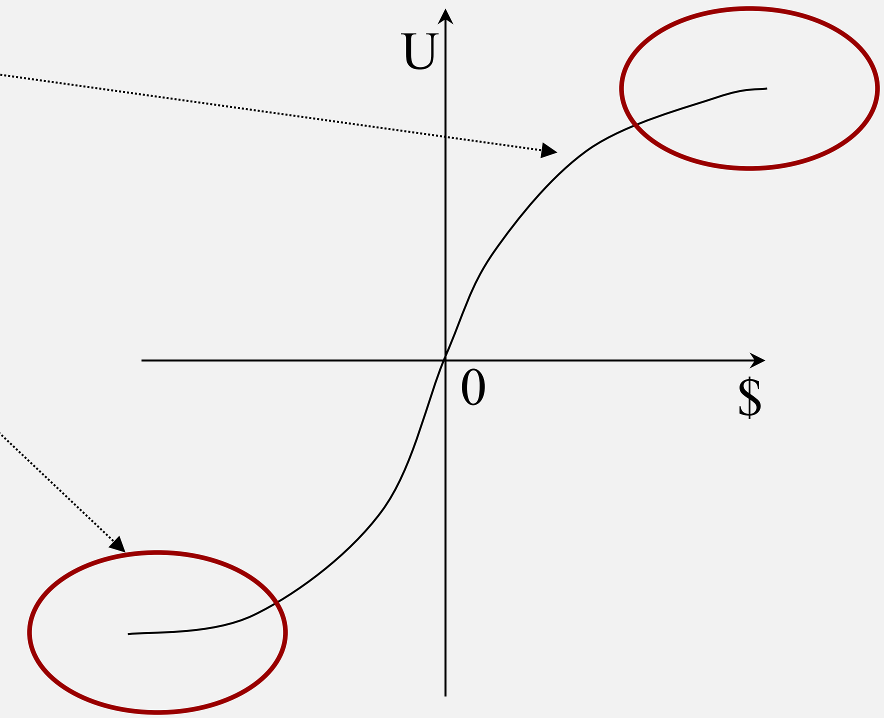
Utility Functions

- Given a lottery ***L***
- risk-averse

$$U(S_L) < U(S_{EMV(L)})$$

- risk-seeking

$$U(S_L) > U(S_{EMV(L)})$$



Multi-attribute Utility function

- Multi-Attribute Utility Theory (MAUT)
  - Outcomes are characterized by 2 or more attributes.
  - E.g., choice of an automobile might need to take into account
    - Price of the automobile
    - Fuel economy
    - Safety
    - ….
- Approach
  - Identify regularities in the preferences to simplify decision-making

Multi-attribute Utility Function

- Notation
  - Attributes $X_1, X_2, X_3, \ldots$
  - Attribute value vector $X = < x_1, x_2, \ldots. >$
  - Utility Function

$$U(x_1, \ldots, x_n)$$

Multi-attribute Utility Theory

- Dominance
  - Certain (strict dominance, Fig.1)
    - eg) airport site S_1 costs less, leads to less noise, safer than S_2
      : strict dominance of S_1 over S_2
  - Uncertain (Fig. 2)

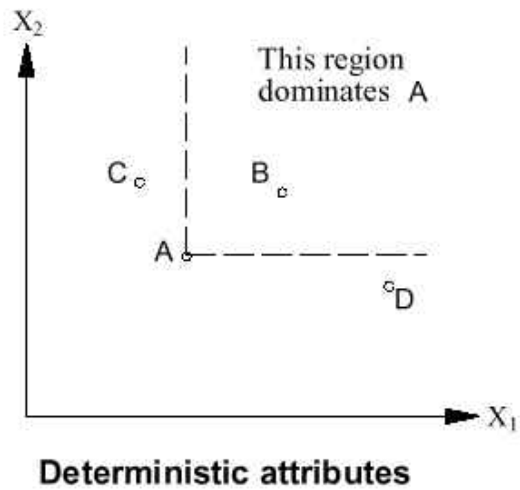


Fig. 1

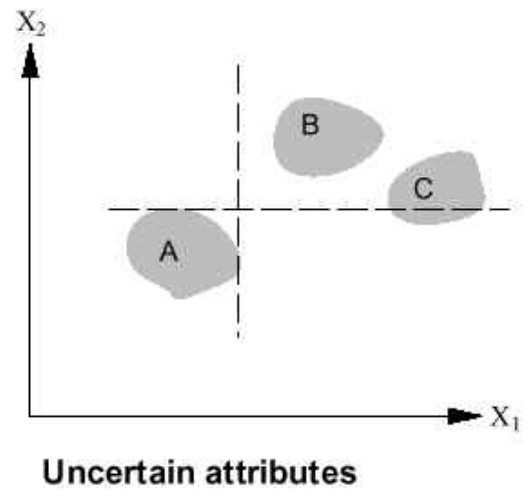


Fig.2

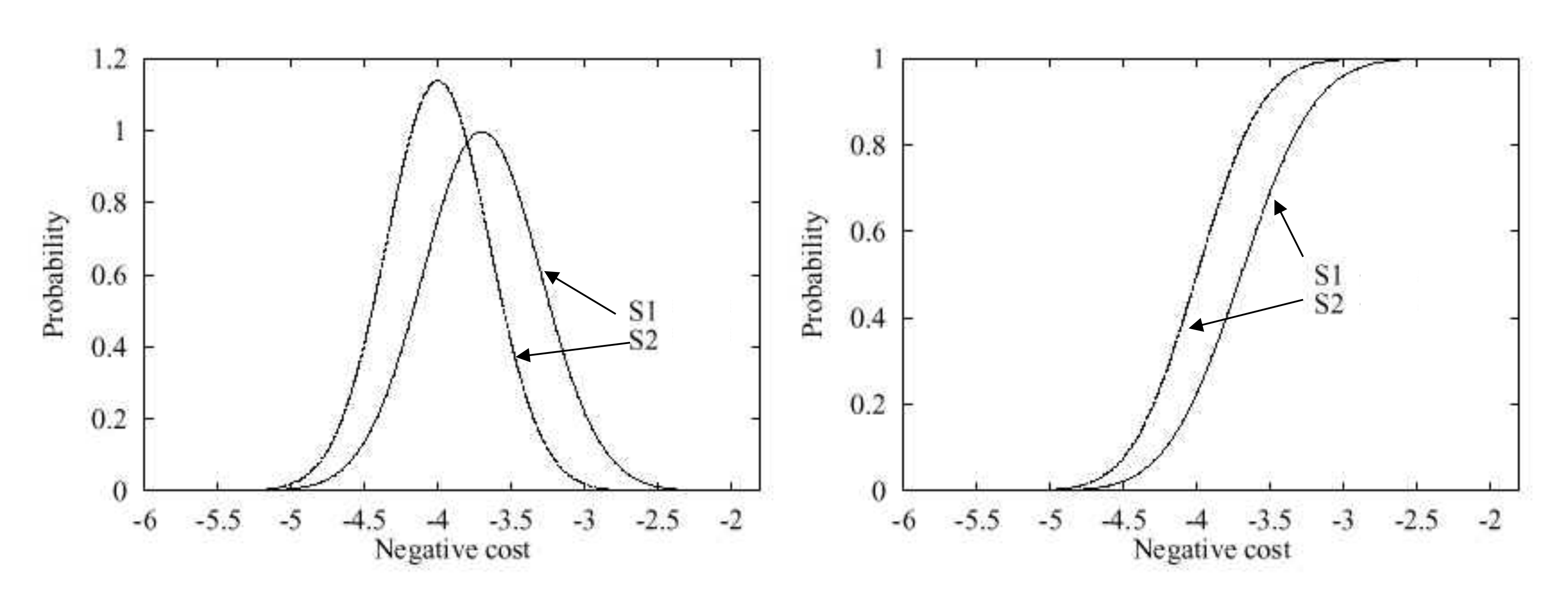
Multi-attribute Utility Theory

Stochastic dominance more common in real-world settings

- S_1 : avg \$3.7billion, standard deviation : \$0.4billion
- S_2 : avg \$4.0billion, standard deviation : \$0.35billion
    - S_1 stochastically dominates S_2

Multi-attribute Utility Theory

- Stochastic dominance



Multi-attribute Utility Function

Consider first preferences in the absence of uncertainty

- preferences between concrete outcome values.
- Preference structure
  - X_1 & X_2 *preferentially independent* of X_3 if preference between $< x_1, x_2, x_3 > \& < x'_1, x'_2, x_3 >$ does not depend on the particular choice of x_3

Example: Airport site location: <Noise,Cost,Safety>

- Suppose Noise and Cost are preferentially independent of Safety:
- If <20,000 suffer, $4.6billion, 0.06deaths/mpm is preferred over <70,000 suffer, $4.2billion, 0.06deaths/mpm> then <20,000 suffer, $4.6billion, 0.08deaths/mpm> is preferred over <70,000 suffer, $4.2billion, 0.08deaths/mpm>

Multi-attribute Utility Function

- Preferences without Uncertainty
- Mutual preferential independence (MPI)
  - Each pair of attributes is preferentially independent of the rest
  - eg) Airport site : <Noise, Cost, Safety>
    - Noise & Cost **P.I** Safety
    - Noise & Safety **P.I** Cost
    - Cost & Safety **P.I** Noise

      : <Noise,Cost,Safety> exhibits **MPI**
- Agent's preference behavior

$$\max \left[V(S) = \sum_{I} \left(V_i X_i(S) \right) \right]$$

Multi-attribute Utility Functions

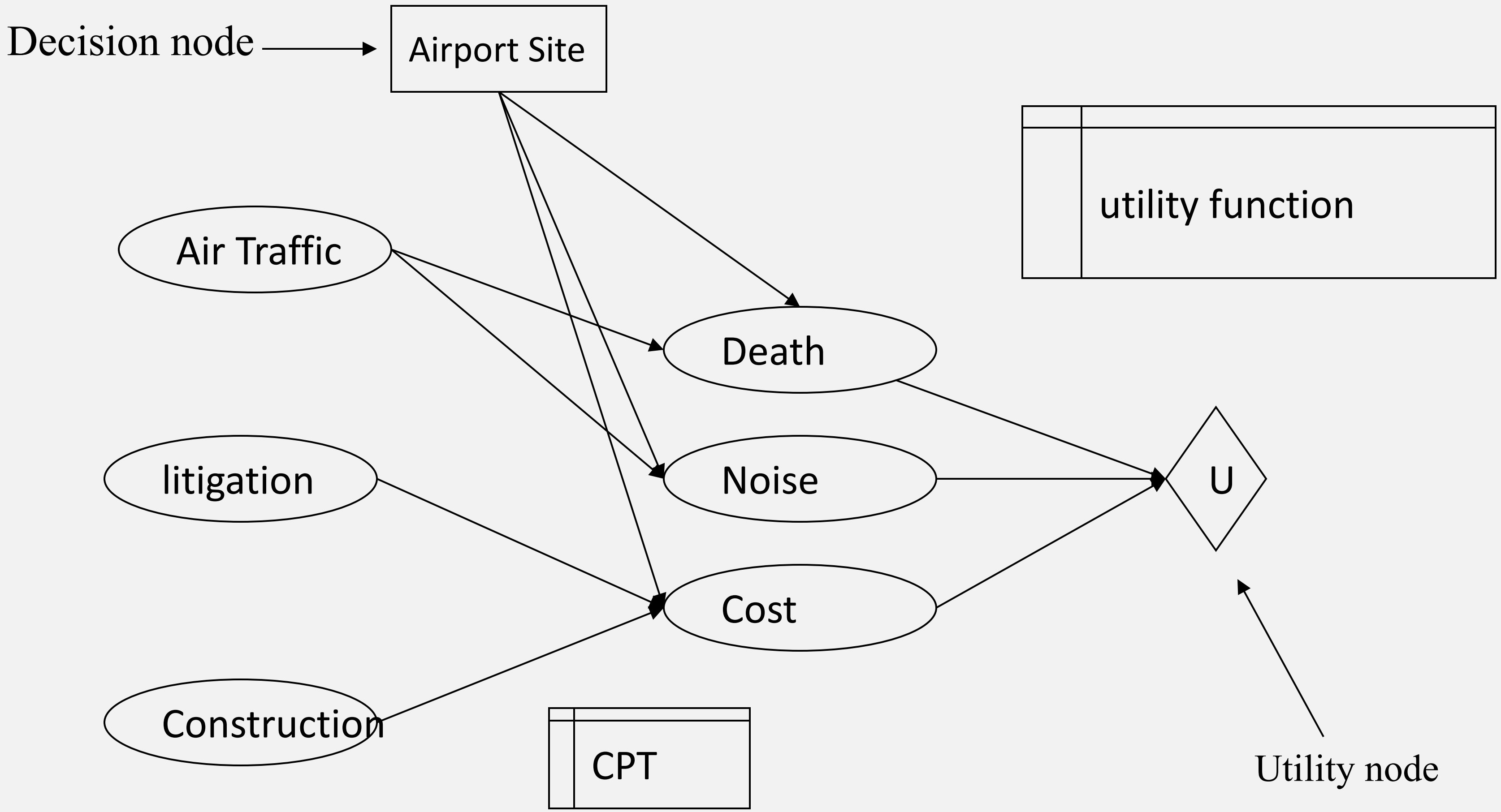
- Preferences in the presence of uncertainty
  - Preferences between Lotteries
  - Utility Independence (UI)
    - **X** is utility-independent of **Y** iff preferences over lotteries' attribute set **X** do not depend on particular values of a set of attribute **Y.**
  - Mutual Utility Independence (MUI)
    - Each subset of attributes is UI of the remaining attributes.
  - Mutual Utility Independence (MUI) implies a multiplicative utility function
  - Example (3 MUI attributes)

$$U = k_1U_1 + k_2U_2 + k_3U_3 + k_1k_2U_1U_2 + k_2k_3U_2U_3 + k_3k_1U_3U_1 + k_1k_2k_3U_1U_2U_3$$

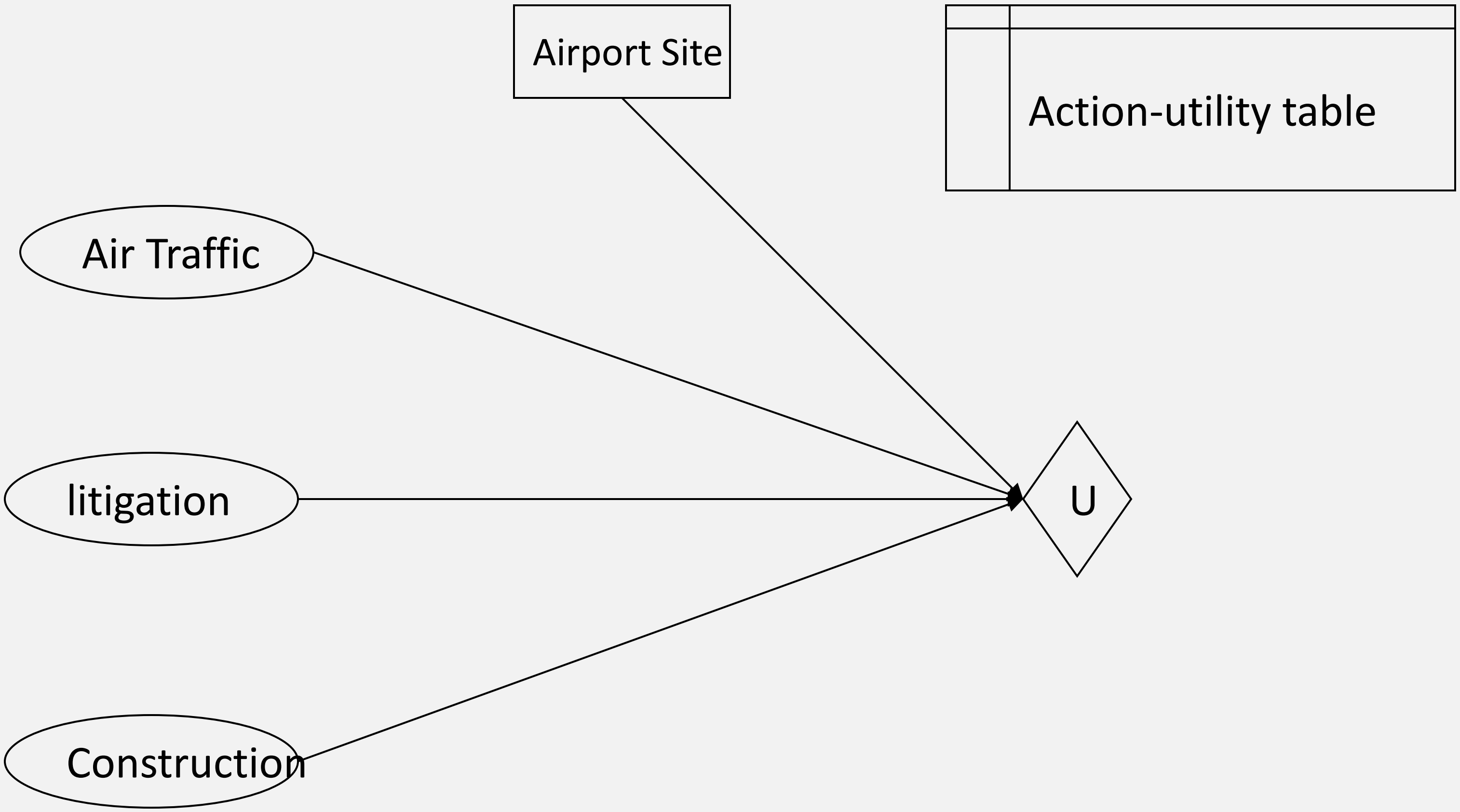
Decision Networks

- Simple formalism for expressing and solving decision problems
- Bayesian networks + decision & utility nodes
- Nodes
  - Chance nodes
  - Decision nodes
  - Utility nodes

A Simple Decision Network



A Simplified representation



Evaluating Decision Networks

Function DN-eval (*percept*) returns *action*

static *D*, a decision network

set evidence variables for the current state

for each possible value of decision node

set decision node to that value

calculate Posterior Prob. For parent nodes of the utility node

calculate resulting utility for the action

select the *action* with the highest utility

return *action*

Value of Information

- Information reduces uncertainty
- Information improves quality of decisions
- How to assess the value of information?
- Example: Buying rights for a diamond mine
  - Three blocks A, B and C, exactly one has diamonds, worth $\$K$
  - Prior probability that any one block has diamonds= 1/3
  - Current price of each block is $K/3$
  - Consultant offers results of a survey that definitively indicates whether or not block A contains diamonds
  - How much should you pay for the results of the survey?

What would you do if you had the results of the survey?

- With probability 1/3, the survey will indicate diamonds in block A
  - Buy block A
  - Profit = $K - (K/3) = (2K/3)$ \$
- With probability 2/3, the survey will indicate no diamonds in block A
  - Buy a block other than A
  - Probability of finding diamonds in one of the remaining blocks (B,C) = ½
  - Expected profit = $(K/2) - (K/3) = (K/6)$ \$

What would you do if you had the results of the survey?

Expected profit given the survey information

$$= \left(\frac{1}{3}\right)\left(\frac{2K}{3}\right) + \left(\frac{2}{3}\right)\left(\frac{K}{6}\right) = \left(\frac{K}{3}\right)\$$$

Expected profit in the absence of the survey information

$$= \left(\frac{1}{3}\right)K - \left(\frac{K}{3}\right) = 0$$

$\therefore$ The survey results for block A are worth at most $\left(\frac{K}{3}\right)\$$

Value of Information

- Expected value of Information

  = Expected value of best action given information

  – Expected value of best action without information

- Survey results "diamonds in A" or 'no diamonds in A"

$$= \frac{1}{3}\times\frac{k}{3}+\frac{2}{3}\times\left(\frac{1}{2}\times\frac{k}{3}+\frac{1}{2}\times\frac{k}{3}\right)-0=\frac{k}{3}$$

General recipe for assessing the value of information

- Current evidence E, Current best action α
- Possible action outcomes $\text{Result}_i(A)=S_i$
- Potential new evidence E_j
- Expected utility EU
- Value of perfect information

$$EU(\alpha \mid E) = \max_A \sum_i U(S_i) P(S_i \mid E, Do(A))$$

$$EU(\alpha_{E_j} \mid E, E_j) = \max_A \sum_i U(S_i) P(S_i \mid E, Do(A), E_j)$$

$$VPI_E(E_j) = \left(\sum_k P(E_j = e_{jk} \mid E) EU(\alpha_{e_{jk}} \mid E, E_j = e_{jk}) \right) - EU(\alpha \mid E)$$

Properties of Value of Perfect Information

- Nonnegative $\forall j, E \quad VPI_E(E_j) \geq 0$
- Nonadditive $VPI_E(E_j, E_k) \neq VPI_E(E_j) + VPI_{E_j}(E_k)$
- Order-Independent

$$VPI_E(E_j, E_k) = VPI_E(E_j) + VPI_{E,E_j}(E_k) = VPI_E(E_k) + VPI_{E,E_k}(E_j)$$

- Imperfect information about a variable X can be modeled by perfect information about a variable Y that is probabilistically related to variable X

Information Gathering Agent

function INFORMATION-GATHERING-AGENT(*percept*) **returns** an *action*
  static: D, a decision network

  integrate *percept* into D
  $j \leftarrow$ the value that maximizes $VPI(E_j) - Cost(E_j)$
  if $VPI(E_j) > Cost(E_j)$
    then return REQUEST(E_j)
  else return the best action from D

Making Simple Decisions: Summary

- Utility theory offers a prescriptive framework for rational decision-making under uncertainty
- Bounded rationality:
  - In the real world, decisions often have to be made with imperfect information, and under tight time and resource constraints
  - Resource bounded rationality often relies on simple heuristics that make us smart

Representing & Reasoning with Qualitative Preferences

Outline

I. Qualitative Preference Languages
- Representation : Syntax of languages CP-nets, TCP-nets, CI-nets, CP-Theories

II. Qualitative Preference Languages
- Ceteris Paribus semantics: the induced preference graph (IPG)
- Reasoning: Consistency, Dominance, Ordering, Equivalence & Subsumption
- Complexity of Reasoning

III. Practical aspects: Preference Reasoning via Model Checking
- From ceteris paribus semantics (IPG) to Kripke structures
- Specifying and verifying properties in temporal logic
- Translating Reasoning Tasks into Temporal Logic Properties

Decision Theory

What is a *decision*?

Choosing from a set of *alternatives A*

How are alternatives described?

What influences choice of an agent?

- preferences, uncertainty, risk

Can decisions be automated?

What happens if there are multiple agents?

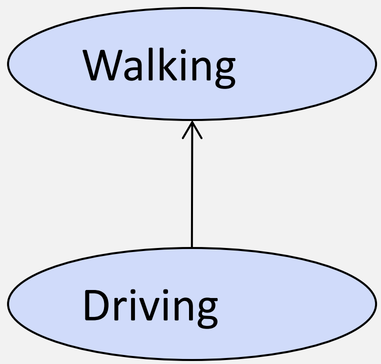
- conflicting preferences and choices

"I prefer walking over driving to work"

There is a 50% chance of snow. Walking may not be good after all.

Qualitative Preferences

Links denote preferences

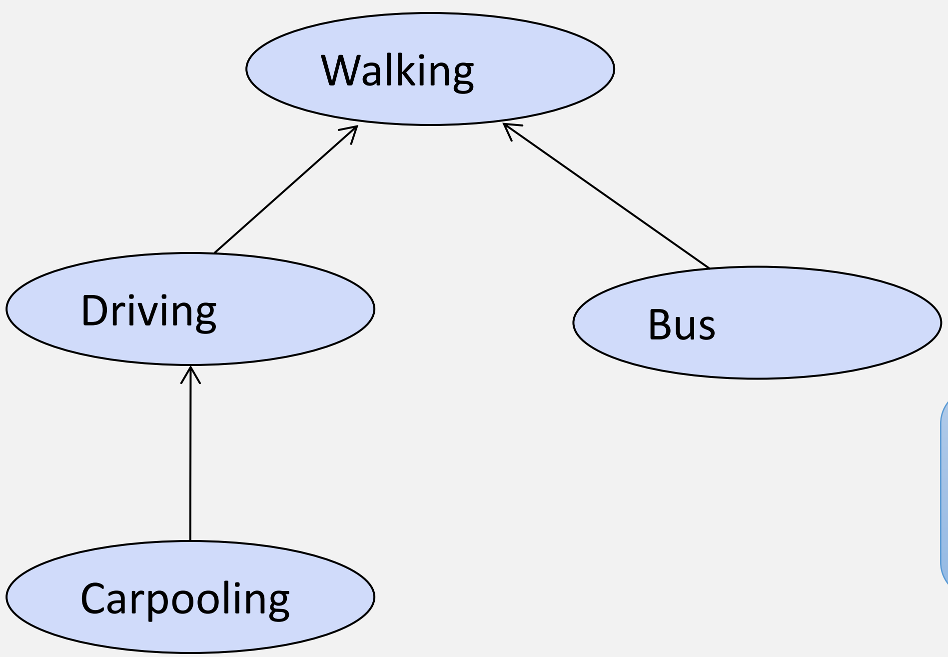


Qualitative

Walking = 0.7; Driving = 0.3

Walking = 0.6; Driving = 0.4

Quantitative



?

False sense of precision
False sense of completeness

Representation: Alternatives are Multi-attributed

Course selection – which course to take?

	497	402	430
Subject?	AI	SE	NW
Instructor?	Gopal	Tom	Bob
# Credits?	4	4	3

- Preference variables or attributes used to describe the domain
- Alternatives are assignments to preference variables
  - α = (instructor = Gopal, area = AI, credits = 3)
- $\alpha \succ \beta$ denotes that α is preferred to β

Qualitative Preference Languages

Qualitative preferences

- Unconditional Preferences
  - TUP-nets [Santhanam et al., 2010]
- Conditional Preferences
  - CP-nets [Boutilier et al. 1997,2002]
  - Models dependencies
- Relative Importance
  - TCP-nets [Brafman et al. 2006]
  - CI-nets [Bouveret et al. 2009]

AI $\succ_{area}$ SE

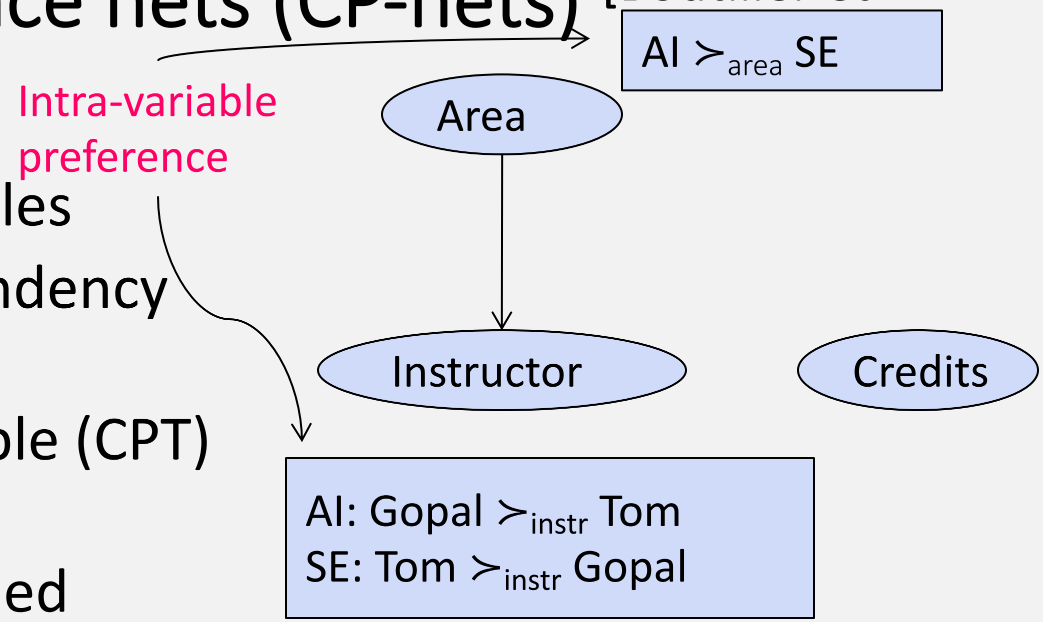
SE : Tom $\succ_{instructor}$ Gopal
AI : Gopal $\succ_{instructor}$ Tom

Instructor $\rhd$ Credits

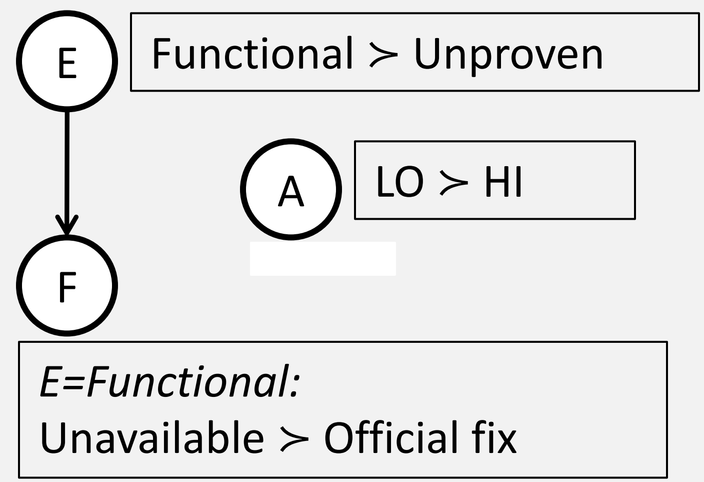
Conditional Preference nets (CP-nets) [Boutilier et al., 1997]

CP-nets

- Nodes – Preference Variables
- Edges – Preferential Dependency between variables
- Conditional Preference Table (CPT) annotates nodes
- CPT can be partially specified



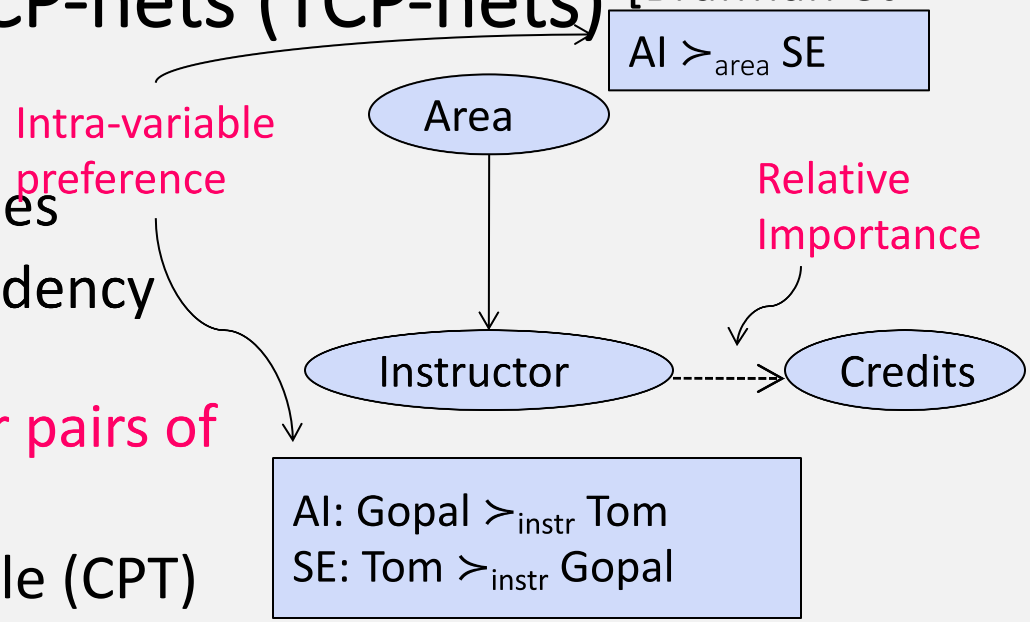
- Relative preferences over:
  - Pairs of values of an attribute

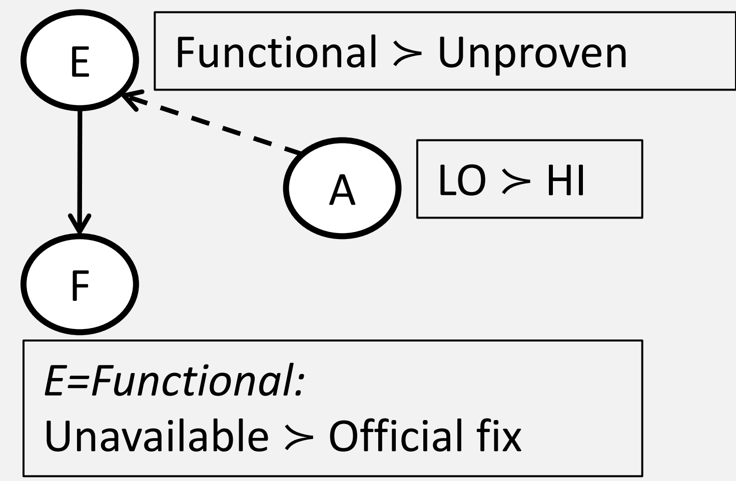


Trade-off enhanced CP-nets (TCP-nets) [Brafman et al., 2006]

TCP-nets

- Nodes – Preference Variables
- Edges – Preferential Dependency between variables
  & Relative Importance over pairs of variables
- Conditional Preference Table (CPT) annotates nodes
- CPT can be partially specified
- Comparative preferences over:
  - Pairs of values of an attribute
  - Pairs of attributes (importance)





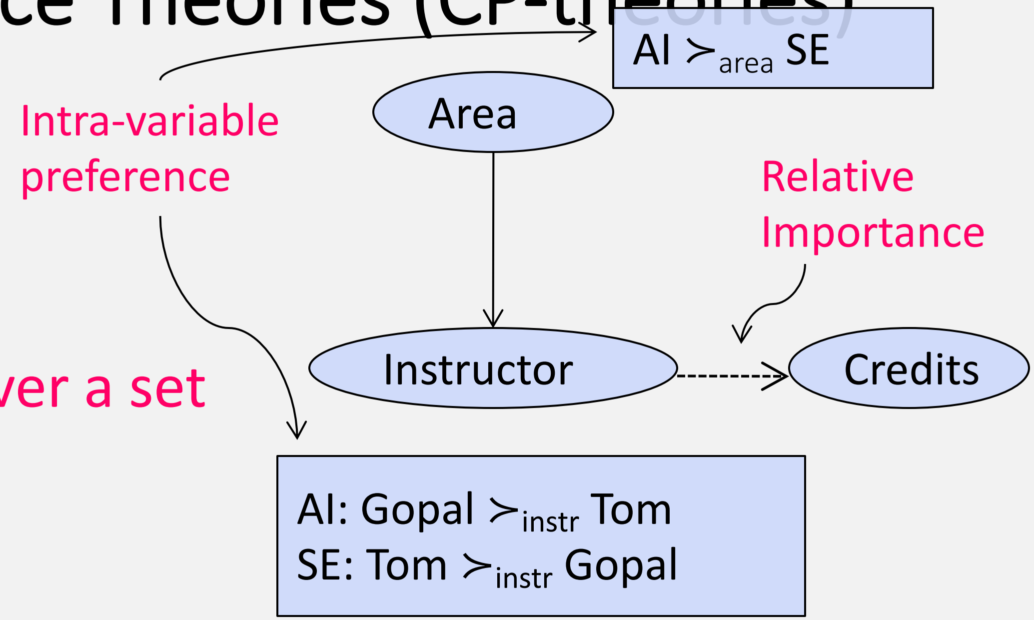
Conditional Preference Theories (CP-theories)

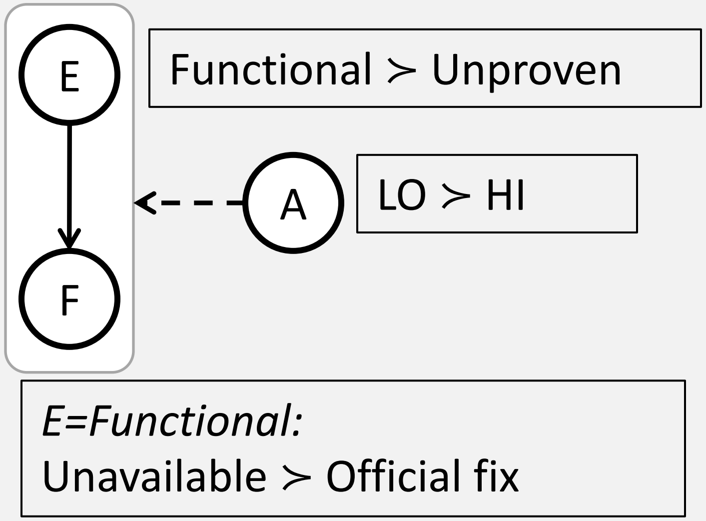
[Wilson 2004,2006]

CP-Theories

- Similar to TCP-nets but..

  Possible to express relative Importance of a variable over a set of variables





Conditional Importance Networks (CI-nets) [Bouveret 2009]

CI-nets (fair division of goods among agents)

- Preference *variables represent items* to be included in a deal
- Preference variables are Binary (presence/absence of an item)
- Intra-variable Preference is monotonic ($0 \succ 1$ or $1 \succ 0$)
  - Subsets preferred to supersets (or vice versa) by default
- CI-net Statements are of the form $S^{+}, S^{-} : S_1 \succ S_2$
  - Represents preference on the *presence of one set of items over another set under certain conditions*
  - If all propositions in S^{+} are true and all propositions in S^{-} are false, then the set of propositions S_1 is preferred to S_2

Conditional Importance Networks (CI-nets) [Bouveret 2009]

CI-nets (fair division of goods among agents)

a = Name
b = Address
c = Bank Routing Number
d = Bank Account Number

P1. $\{d\}, \{\} : \{b\} \succ \{c\}$
P2. $\{b\}, \{a\} : \{c\} \succ \{d\}$
P3. $\{\}, \{d\} : \{a, b\} \succ \{c\}$

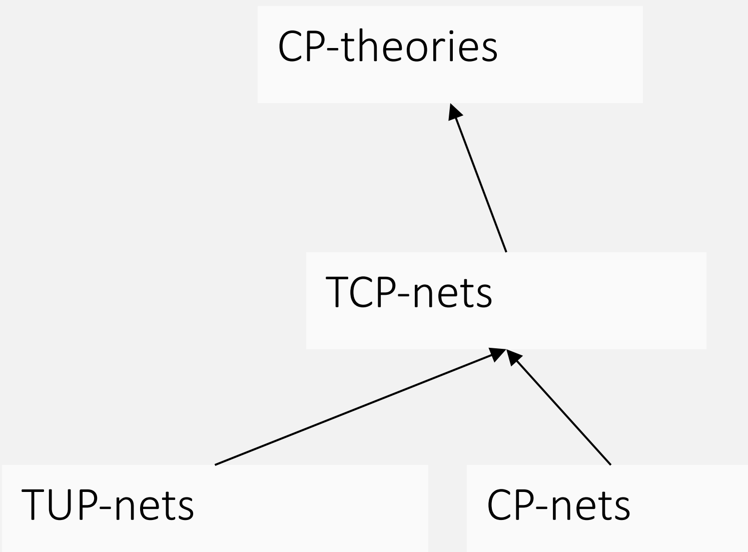
If I have to …
disclose my **address** without having to disclose my **name**,
then I would prefer …
giving my **bank routing number**
over …
my **bank account number**

Other Preference Languages

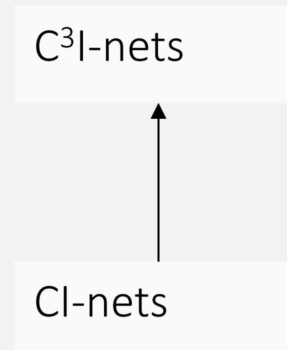
- Preference languages in Databases [Chomicki 2004]
- Preferences over Sets [Brafman et al. 2006]
- Preferences among sets (incremental improvement)[Brewka et al. 2010]
- Tradeoff-enhanced Unconditional Preferences (TUP-nets) [Santhanam et al. 2010]
- Cardinality-constrained CI-nets (C^3I-nets) [Santhanam et al. 2013]

Relative Expressivity of Preference Languages

Preferences over
Multi-domain Variables



Preferences over
(Sets of) Binary Variables



Preference Reasoning

- Exact Reasoning about Qualitative Preferences

Not covered :

- Uncertainty + Preferences
  - Cornelio et al. *Updates and Uncertainty in CP-Nets 2013*
  - Bigot et al. *Probabilistic CP-nets 2013*
- Applications
  - Rossi et al. *Preference Aggregation: Social Choice 2012*
  - Chomicki et al. *Skyline queries in Databases 2011*
  - Trabelsi et al. *Preference Induction Recommender systems 2013*
- Other Reasoning Approaches
  - Minyi et al. *Heuristic approach to dominance testing in CP-nets* 2011
  - Wilson *Upper Approximation for Conditional Preferences 2006*

Other Preference Languages

- Preference languages in Databases [Chomicki 2004]
- Preferences over Sets [Brafman et al. 2006]
- Preferences among sets (incremental improvement)[Brewka et al. 2010]
- Tradeoff-enhanced Unconditional Preferences (TUP-nets) [Santhanam et al. 2010]
- Cardinality-constrained CI-nets (C^3I-nets) [Santhanam et al. 2013]

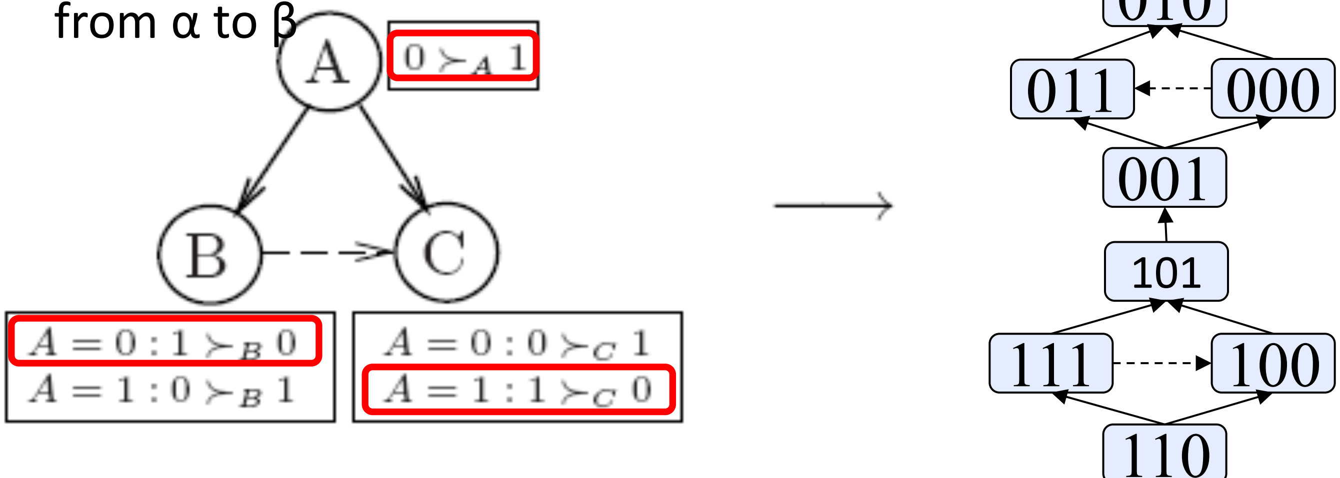
- We limit our discussion to CP-nets, TCP-nets and CI-nets
- Approach extensible to all other preference languages with *Ceteris paribus* semantics

Key concepts

- Induced Preference Graph (IPG)
- Semantics in terms of flips in the IPG
- Reasoning Tasks
  - Dominance over Alternatives
  - Equivalence & Subsumption of Preferences
  - Ordering of Alternatives
- Complexity of Reasoning

Induced Preference Graph (IPG) [Boutilier et al. 2001]

- *Induced preference graph* δ(P)= *G*(*V*,*E*) of preference spec P:
  - Nodes *V* : set of alternatives
  - Edges *E* : (α , β) ∈ *E* iff there is a *flip induced by some preference in P* from α to β



- δ(N) is acyclic (dominance is a strict partial order)
- α ≻ β iff there is a *path* in δ(N) from β to α (serves as the *proof*)

Santhanam et al. AAAI 2010

Preference Semantics in terms of IPG

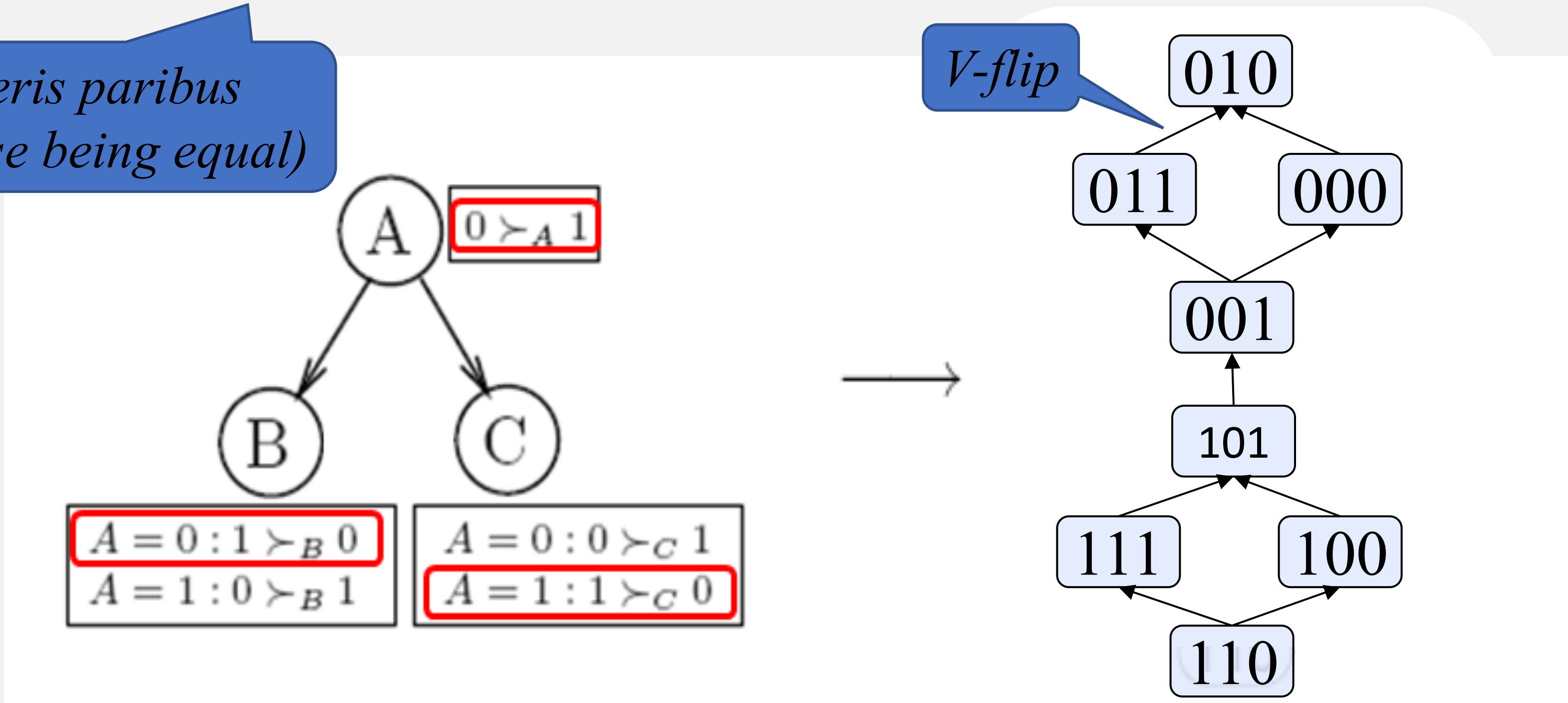
- $(\beta, \alpha) \in E$ iff there is a *flip* from α to β *"induced by some preference" in P*
- Types of flips
  - Ceteris Paribus flip – flip a variable, "all other variables equal"
  - Specialized flips
    - Relative Importance flip
    - Set based Importance flip
    - Cardinality based Importance flip
- Languages differ in the semantics depending on the specific types of flips they allow

… Next: examples

Flips for a CP-net [Boutilier et al. 2001]

- $(\beta, \alpha) \in E$ iff there is a statement in CP-net such that $x_1 \succ_1 x'_1$ (x_1 is preferred to x'_1) and …
  - ***V-flip*** : *all other variables being equal*, $\alpha(X_1)=x_1$ and $\beta(X_1)=x'_1$

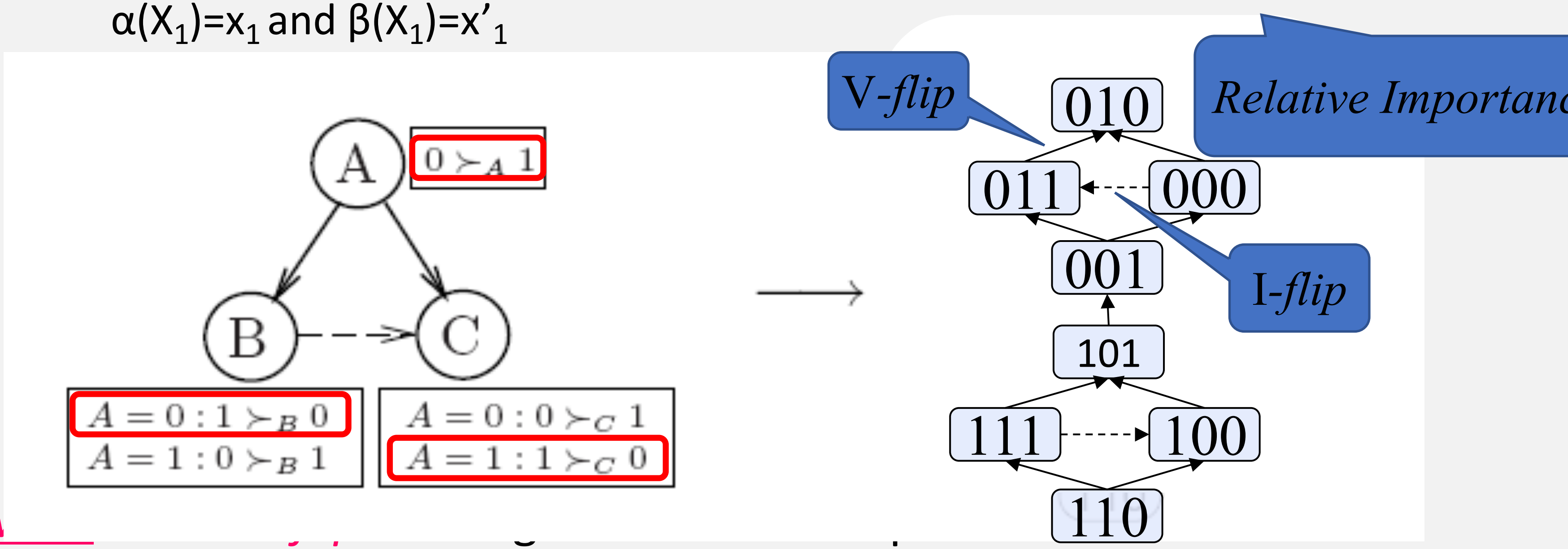
Ceteris paribus (all else being equal)



Single variable flip – change value of 1 variable at a time

Flips for TCP-nets & CP-theories [Brafman et al., Wilson 2004]

- (α , β) ∈ *E* iff there is a statement in TCP-net such that $x_1 \succ_1 x'_1$ (x_1 is preferred to x'_1) and …
  - ***V-flip*** : all other variables being equal, $\alpha(X_1)=x_1$ and $\beta(X_1)=x'_1$
  - ***I-flip*** : all variables *except **<u>those</u>** less important than X_1 being equal*, $\alpha(X_1)=x_1$ and $\beta(X_1)=x'_1$



Flips for a CI-net [Bouveret 2009]

- CI-nets express *preferences over subsets* of binary variables X.
  - Truth values of X_i tells its presence/absence in a set
  - Nodes in IPG correspond to subsets of X
  - Supersets are always preferred to Strict Subsets (convention)
  - $S^+, S^- : S_1 \succ S_2$ interpreted as …

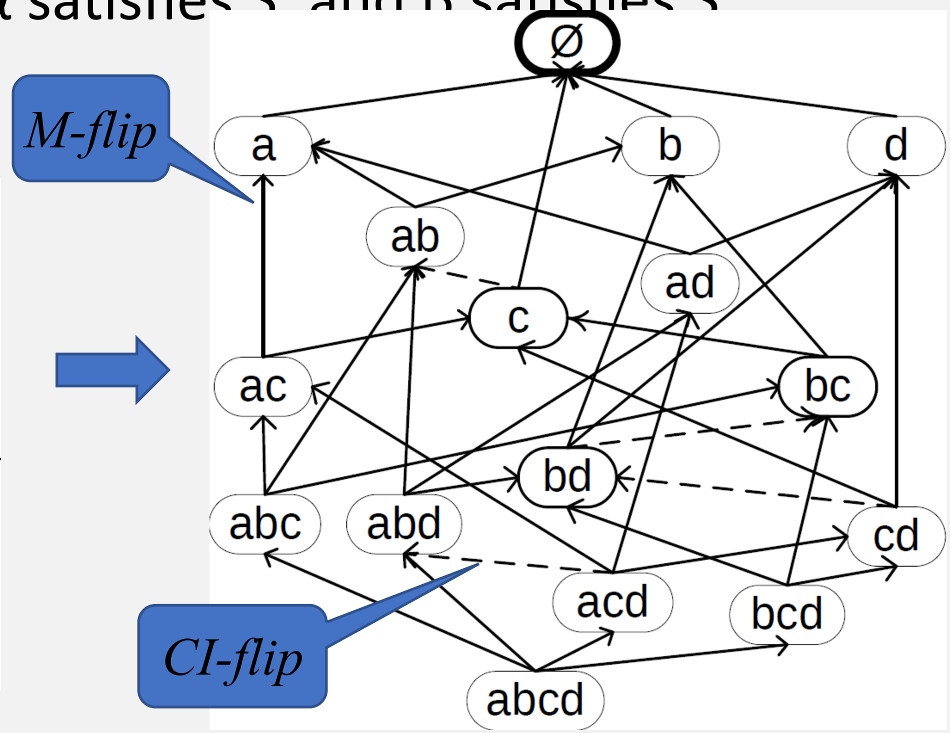
    If all propositions in S^+ are true and all propositions in S^- are false, then the set of propositions S_1 is preferred to S_2
- For α , $\beta \subseteq X$, $(\alpha, \beta) \in E$ (β preferred to α) iff
  - ***M-flip*** : all other variables being equal, $\alpha \subset \beta$
  - ***CI-flip*** : there is a CI-net statement s.t. $S^+, S^- : S_1 \succ S_2$ and α , β satisfy S^+, S^- and α satisfies S^+ and β satisfies S^-.

Flips for a CI-net [Bouveret 2009]

- For α , β ⊆ X, (α, β) ∈ *E* (β preferred to α) iff
  - ***M-flip*** : all other variables being equal, α ⊂ β
  - ***CI-flip*** : there is a CI-net statement $S^+, S^- : S_1 \succ S_2$ s.t. α , β satisfy S^+, S^- and α satisfies S^+ and β satisfies S^-
- Example:

a = Name
b = Address
c = Bank Routing Number
d = Bank Account Number

P1. $\{d\}, \{\} \ : \ \{b\} \succ \{c\}$
P2. $\{b\}, \{a\} \ : \ \{c\} \succ \{d\}$
P3. $\{\}, \{d\} \ : \ \{a, b\} \succ \{c\}$



Oster et al. FACS 2012

Flips for a C^3I-net [Santhanam et al. 2013]

- C^3I-nets express *preference over subsets* similar to CI-net
  - Truth values of X_i tells its presence/absence in a set
  - Nodes in IPG correspond to subsets of X
  - Sets with *higher cardinality* are preferred (conventional)
  - $S^+, S^- : S_1 \succ S_2$ interpreted as …

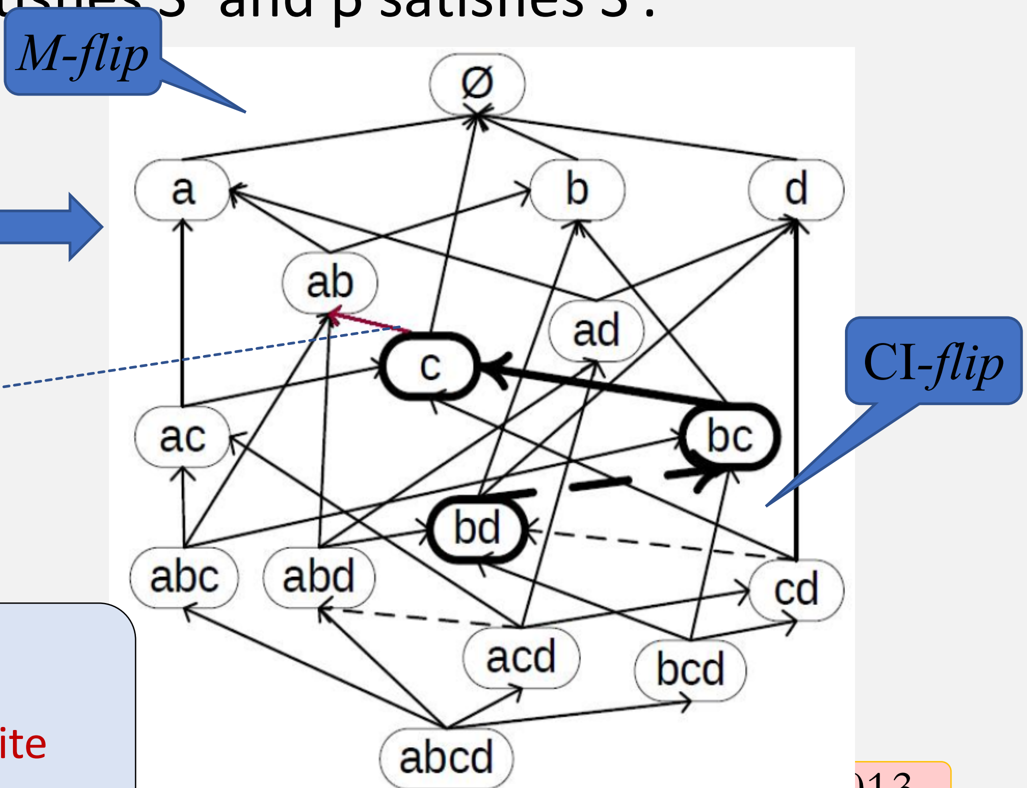
    If all propositions in S^+ are true and all propositions in S^- are false, then the set of propositions S_1 is preferred to S_2
- For α , $\beta \subseteq X$, $(\alpha, \beta) \in E$ (β preferred to α) iff
  - **_M-flip_** : all other variables being equal, $|\alpha| < |\beta|$
  - *CI-flip* : there is a CI-net statement s.t. $S^+, S^- : S_1 \succ S_2$ and α , β satisfy S^+, S^- and α satisfies S^+ and β satisfies S^-.
  - Extra cardinality constraint to enable dominance

Flips for a C³I-net [Santhanam et al. 2013]

- For α , β ⊆ X, (α, β) ∈ *E* (β preferred to α) iff
  - ***M-flip*** : α ⊂ β (all other variables being equal)
  - ***CI-flip*** : there is a CI-net statement $S^+, S^- : S_1 \succ S_2$ s.t. α , β satisfy S^+, S^- and α satisfies S^+ and β satisfies S^-.
  - ***C-flip*** : |α| < |β|

$$
\begin{array}{llcl}
\text{P1.} & \{d\},\{\} & : & \{b\} \succ \{c\} \\
\text{P2.} & \{b\},\{a\} & : & \{c\} \succ \{d\} \\
\text{P3.} & \{\},\{d\} & : & \{a,b\} \succ \{c\}
\end{array}
$$

C-flip - present in the CI-net, but **not** in the C³I-net



- {c} ≻ {bc} due to Monotonicity
- {bc} ≻ {bd} due to P2
- {ab} ⊁ {c} due to Cardinality despite P3

Santhanam et al. CSIIRW 2013

Reasoning Tasks

The semantics of any ceteris paribus language
can be represented in terms of properties of IPG

- Now we turn to the Reasoning Tasks:
  - Dominance & Consistency
  - Equivalence & Subsumption
  - Ordering
- Reasoning tasks reduce to verifying properties of IPG

Reasoning Tasks

Dominance relation:

- α ≻ β iff there exists a *sequence of flips* from β to α
- Property to verify: *Existence of path in IPG* from β to α

Consistency:

- A set of preferences is *consistent* if ≻ is a strict partial order
- Property to - verify: *IPG is acyclic*

Ordering: ?

semantics

- *Hint:* The non-dominated alternatives in the IPG are the best
- Strategy – Repeatedly Query IPG to get strata of alternatives

Equivalence (& Subsumption):

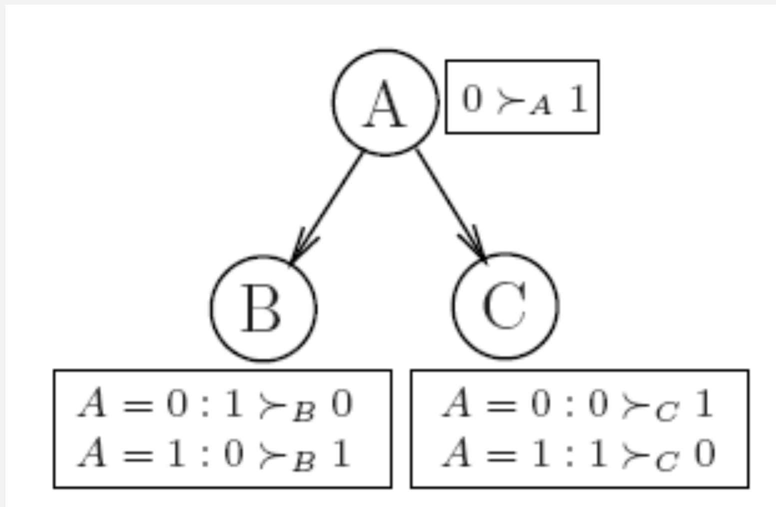
- A set P_1 of preferences is *equivalent* to another set P_2 if they induce the same dominance relation
- Property to verify: *IPGs are reachability equivalent*

Reasoning Tasks

Reasoning Task	Computation Strategy: Property of IPG to check	Remarks
Dominance: $\alpha \succ \beta$	Is β reachable from α ?	
Consistency of a set of preferences (P)	Is the IPG of P acyclic?	Satisfiability of the dominance relation; strict partial order
Equivalence of two sets of preferences P_1 and P_2	Are the IPGs of P_1 and P_2 reachability-equivalent?	
Subsumption of one set of preference (P_1) by another (P_2)	If β reachable from α in the IPG of P_1, does the same hold in the IPG of P_2?	
Ordering of alternatives	Iterative verification of the IPG for the non-existence of the non-dominated alternatives	Iterative modification of the IPG to obtain next set of non-dominated alternatives

Complexity of Dominance [Goldsmith et al. 2008]

Cast as a *search* for a flipping sequence, or *a path in IPG*



PSPACE-complete

- α = (A = 1, B = 0, C = 0)
- β = (A = 0, B = 1, C = 1)
- α ≻ β – Why?

Complexity of Reasoning Tasks

Reasoning Task	Complexity	Source
Dominance: $\alpha \succ \beta$	PSPACE-complete	Goldsmith et al. 2008
Consistency of a set of preferences (P)	PSPACE-complete	Goldsmith et al. 2008
Equivalence of two sets of preferences P_1 and P_2	PSPACE-complete	Santhanam et al. 2013
Subsumption of one set of preference (P_1) by another (P_2)	PSPACE-complete	Santhanam et al. 2013
Ordering of alternatives	NP-hard	Brafman et al. 2011

Practical Aspects

Part III – Outline

- Two Sound and Complete Reasoning Approaches:
  - Logic Programming
    - Answer Set Programming [Brewka et al.]
    - Constraint Programming [Brafman et al. & Rossi et al.]
  - Model Checking based
    - Preference reasoning can be reduced to verifying properties of the IPG [Santhanam et al. 2010]
    - Translate IPG into a Kripke Structure Model
    - Translate reasoning tasks into temporal logic properties over model
- Approximation & Heuristics
  - Wilson [Wilson 2006, 2011]

Preference Reasoning via Model Checking

- The *first practical solution to preference reasoning* in moderate sized CP-nets, TCP-nets, CI-nets, etc.
  - Casts dominance testing as reachability in an induced graph
  - Employs direct, succinct encoding of preferences using Kripke structures
  - Uses Temporal logic (CTL, LTL) for querying Kripke structures
  - Uses direct translation from reasoning tasks to CTL/LTL
    - Dominance Testing
    - Consistency checking (loop checking using LTL)
    - Equivalence and Subsumption Testing
    - Ordering (next-preferred) alternatives

Santhanam et al. (AAAI 2010, KR 2010, ADT 2013); Oster et al. (ASE 2011, FACS 2012)

Model Checking [Clark et al. 1986]

φ

- Model Checking: Given a desired property, (typically expressed as a temporal logic formula φ), and a (Kripke) structure *M* with initial state s, decide if *M, s* ⊨
- Active area of research in formal methods, AI (SAT solvers)
- Broad range of applications: hardware and software verification, security..
- Temporal logic languages : CTL, LTL, μ-calculus, etc.
- Many model checkers available : SMV, NuSMV, Spin, etc.

Advantages of Model Checking:

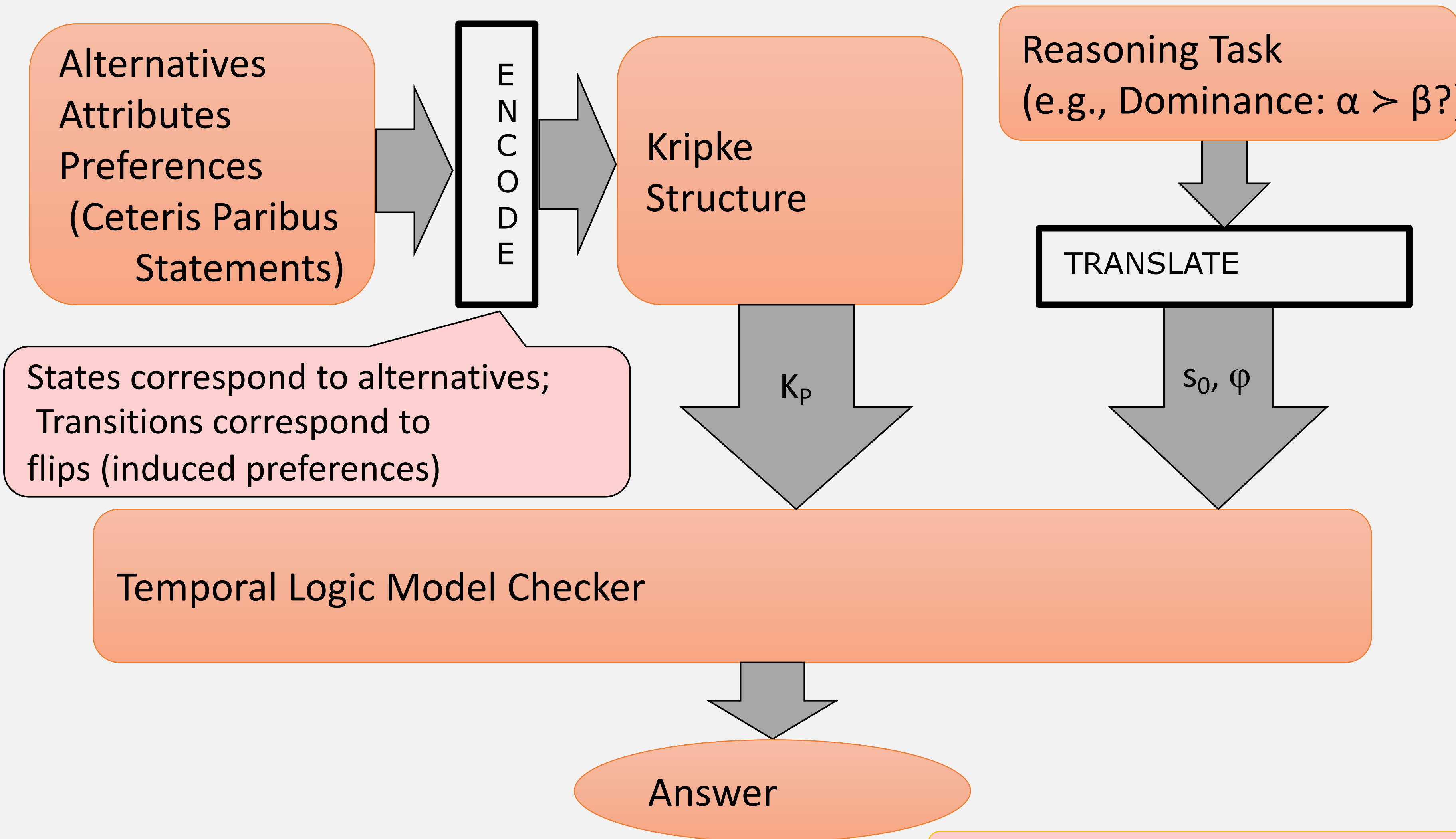
1. Formal Guarantees
2. Justification of Results

Preference Reasoning via Model Checking

Preference reasoning can be reduced to verifying properties of the Induced Preference Graph [Santhanam et al. 2010]

- Overview of Approach
  1. Translate IPG into a Kripke Structure Model
  2. Translate reasoning tasks into verification of temporal logic properties on the model

Overview: Preference Reasoning via Model Checking



Santhanam et al. AAAI 2010

Kripke Structure [Kripke, 1963]

A Kripke structure is a 4-tuple $K=(S, S_0, T, L)$ over variables V, where

- S represents the set of reachable states of the system
- S_0 is a set of initial states
- T represents the set of state transitions
- L is labeling (interpretation) function maps each node to a set of atomic propositions AP that hold in the corresponding state

Used to specify labeled transition systems describing states of the world w.r.t. flow of time

Computational tree temporal logic (CTL) is an extension of propositional logic

- Includes temporal connectives that allow specification of properties that hold over states and paths in K

Example

- $EF\varphi$ true in state s of K if φ holds in some state in some path beginning at s

Encoding Preference Semantics

Let P = {p_i} be a set of ceteris paribus preference statements on a set of preference variables X = {x_1, x_2, …}

Reasoning Strategy:

- Construct a Kripke model K_P = (S, S_0, T, L) using variables Z
  - Z = {z_i | $x_i \in X$}, with each variable z_i having same domain D_i as x_i
  - K_P must mimic the IPG in some sense
- The State-Space of K_P
  - S = $\Pi_i D_i$: states correspond to set of all alternatives
  - T : transitions correspond to allowed changes in valuations according to flip-semantics of the language
  - L : labeling (interpretation) function maps each node to a set of atomic propositions *AP* that hold in the corresponding state
  - S_0 : Initial states assigned according to the reasoning task at hand

From Syntax to Semantics

Encode K_P such that paths in IPG are enabled transitions, and no additional transitions are enabled

- Let p be a conditional preference statement in P
- p induces a flip between two nodes in the IPG iff
  1. "Condition" part in the preference statement is satisfied by both nodes
  2. "Preference" part (less & more preferred valuations) is satisfied by both
  3. "Ceteris Paribus" part that ensures apart from (1 & 2) that all variables other than those specified to change as per (2) are equal in both nodes
- Create transitions in K_P with guard conditions
  - "Condition" part of statement is translated to the guard condition
  - "Preference" part of statement is translated to assignments of variables in the target state
  - How to ensure ceteris paribus condition?

From Syntax to Semantics

Encode K_P such that paths in IPG are enabled transitions, and no additional transitions are enabled

- Let p be a conditional preference statement in P
- p induces a flip between two nodes in the IPG iff
  1. "Condition" part in the preference statement is satisfied by both nodes
  2. "Preference" part (less & more preferred valuations) in satisfied by both
  3. "Ceteris Paribus" part that ensures apart from (1 & 2) that all variables other than those specified to change as per (2) are equal in both nodes
- Create transitions in K_P with guard conditions
  - "Condition" part of statement is translated to the guard condition
  - "Preference" part of statement is translated to assignments of variables in the target state

How to encode ceteris paribus condition in the guards?

From Syntax to Semantics

Recall: In temporal logics, destination states represent "**future**" state of the world

- Equality of source and destination states forbidden as part of the guard condition specification!
- Workaround: Use auxiliary variables h_i to label edges

$$h_i = \begin{cases} 0 & \Rightarrow \quad \text{value of } z_i \textbf{ must not} \text{ change in a transition in the Kripke structure } K(P) \\ 1 & \Rightarrow \quad \text{otherwise} \end{cases} \quad (1)$$

- Auxiliary edge labels don't contribute to the state space

From Syntax to Semantics

Guard condition specification

- Recall: p induces a flip between two nodes in the IPG iff
  1. "Condition" part in the preference statement is satisfied by both nodes
  2. "Preference" part (less & more preferred valuations) in satisfied by both
  3. "Ceteris Paribus" part that ensures apart from (1 & 2) that all variables other than those specified to change as per (2) are equal in both nodes
- For each statement p of the form $\varrho : x_i = v_i \succ_{x_i} x_i = v_i'$ where ϱ is the "condition" part, guard condition is

$$\mathcal{G}(p) = Allow(p) \wedge Restrict(p) \text{ s.t.}$$

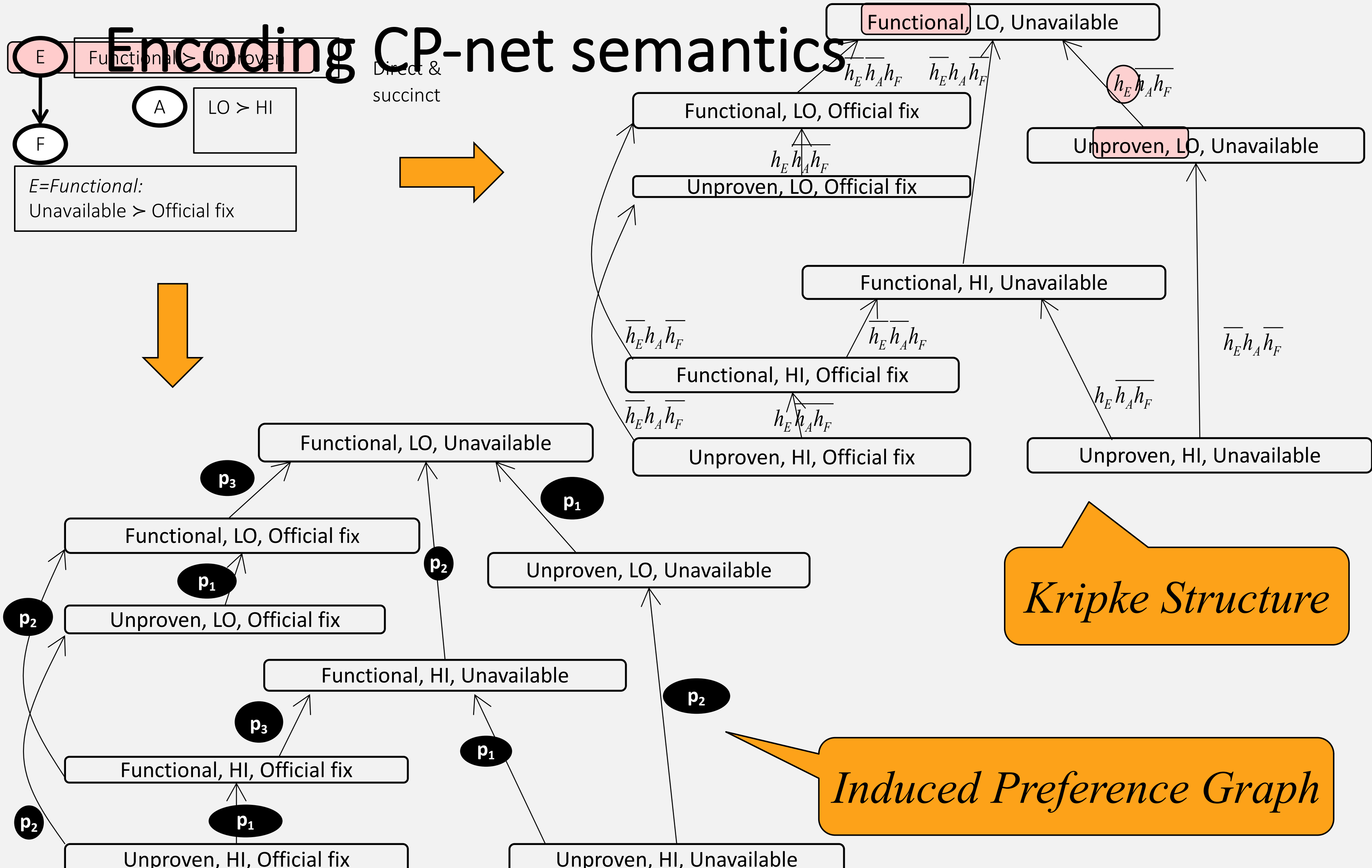
condition

preference

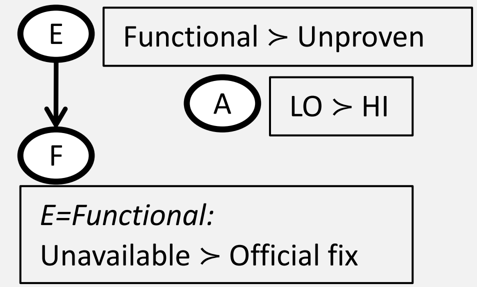
$$\begin{aligned} Allow(p) &:= \varrho \wedge z_i = v_i' \wedge h_i = 1 \\ Restrict(p) &:= \bigwedge_{x_j \in X \setminus \{x_i\}} h_j = 0 \end{aligned}$$

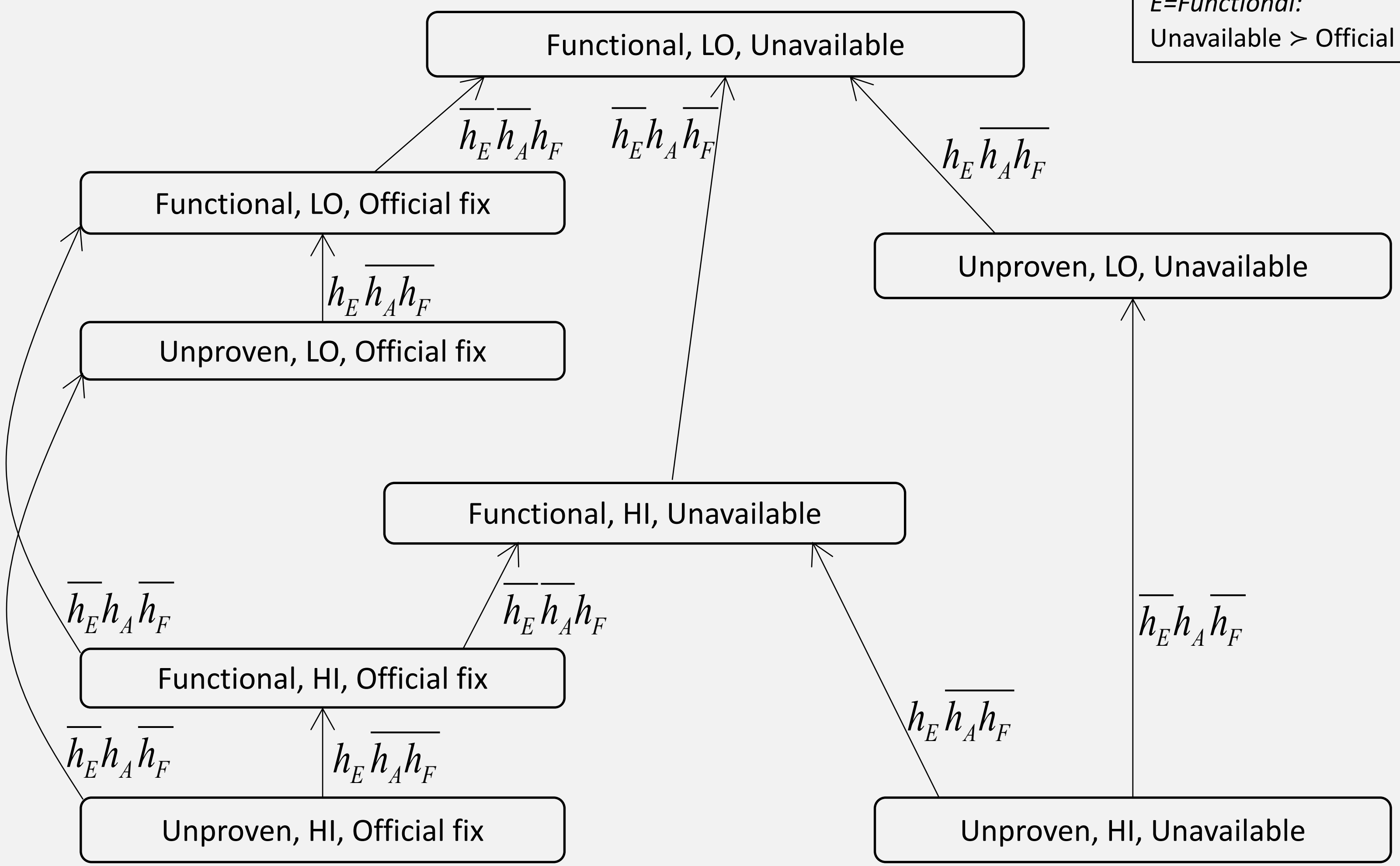
ceteris paribus

Encoding CP-net semantics



Encoding CP-net semantics

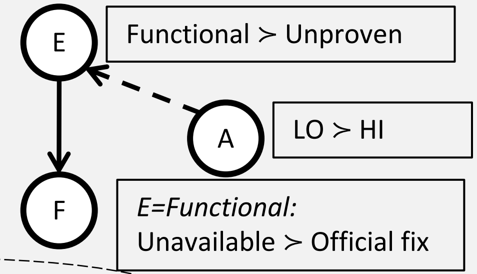


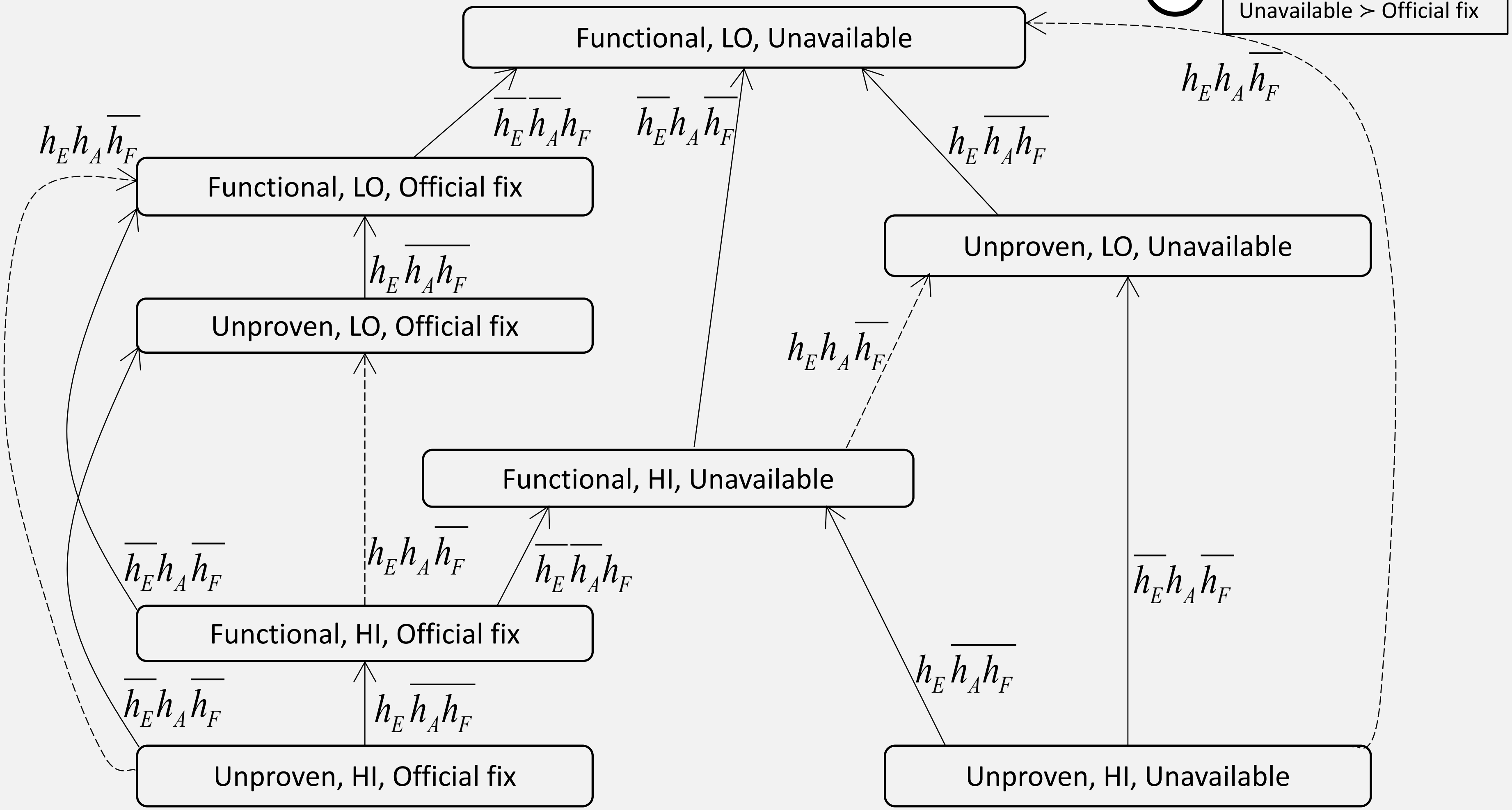


Encoding TCP-net Semantics

TCP-nets : Same overall idea as CP-nets

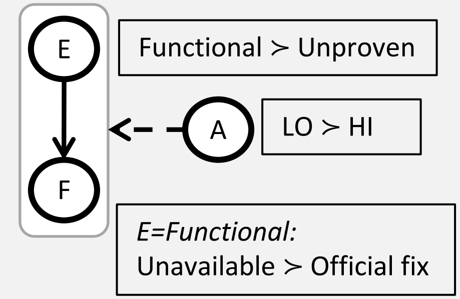
- Additional rule for encoding simple relative importance

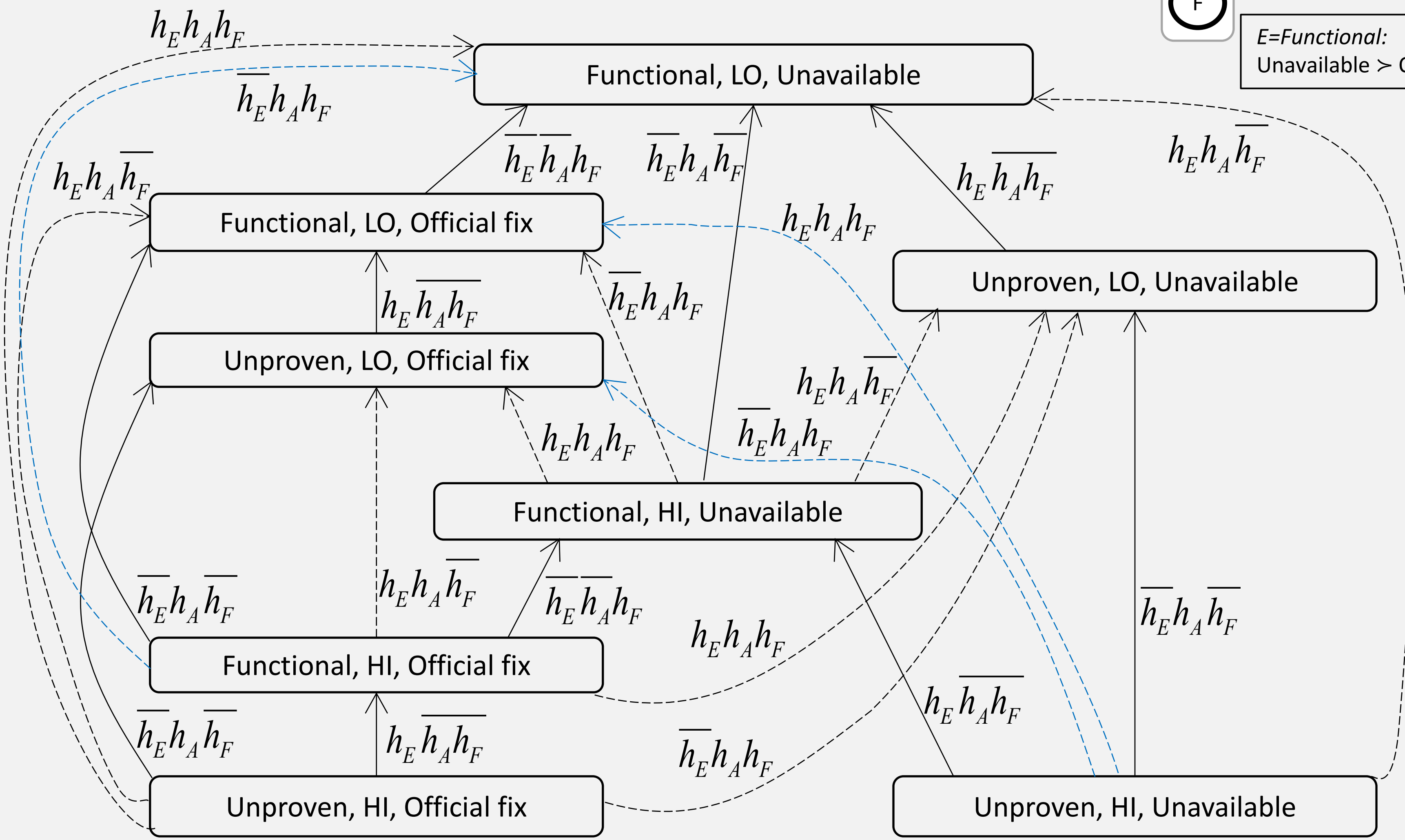




Encoding CP-theory Semantics

CP-theory: Same idea as TCP-net + Additional rule





Encoding Reasoning Tasks as Temporal Logic Properties

Next :

Specifying and Verifying Properties in Temporal Logic

Translating Reasoning Tasks into Temporal Logic Properties

Encoding Reasoning Tasks as Temporal Logic Properties

Computational tree temporal logic (CTL) [Clark et al. 1986] is an extension of propositional logic

- Includes temporal connectives that allow specification of properties that hold over states and paths in a Kripke structure
- CTL Syntax & Semantics

  $\texttt{EX}\ \psi$ if there exists a path $s = s_1 \rightarrow s_2 \ldots$ such that s_2 satisfies ψ

  $\texttt{AX}\ \psi$ if for all paths such that $s = s_1 \rightarrow s_2 \ldots$, s_2 satisfies ψ

  $\texttt{E}\ [\psi_1 \texttt{U}\ \psi_2]$ if there exists a path $s = s_1 \rightarrow s_2 \ldots$ such that $\exists i \geq 1 : s_i$ satisfies ψ_2, and $\forall j < i : s_j$ satisfies ψ_1
- Translating Reasoning Tasks into Temporal Logic Properties
  - Dominance Testing
  - Consistency
  - Equivalence & Subsumption Testing
  - Ordering alternatives

NuSMV [Cimatti et al. 2001]:
Our choice of model checker

Dominance Testing (via NuSMV)

Given outcomes α and β, how to check if $\alpha \succ \beta$?

- Let φ_α be a formula that holds in the state corresponding to α
- Let φ_β be a formula that holds in the state corresponding to β

By construction, $\alpha \succ \beta$ wrt iff in the Kripke Structure K_N :

a state in which $\boldsymbol{\varphi_\beta}$ *holds is reachable from a state in which* $\boldsymbol{\varphi_\alpha}$ *holds*

- $\alpha \succ \beta$ iff the model checker NuSMV can verify $\varphi_\alpha \rightarrow \mathrm{EF}\varphi_\beta$ (SAT)
- When queried with ¬($\varphi_\alpha \rightarrow \mathrm{EF}\varphi_\beta$), if indeed $\alpha \succ \beta$, then model checker produces a proof of $\alpha \succ \beta$ (flipping sequence)
- Experiments show feasibility of method for 100 var. in seconds

Santhanam et al. AAAI 2010

Obtaining a Proof of Dominance

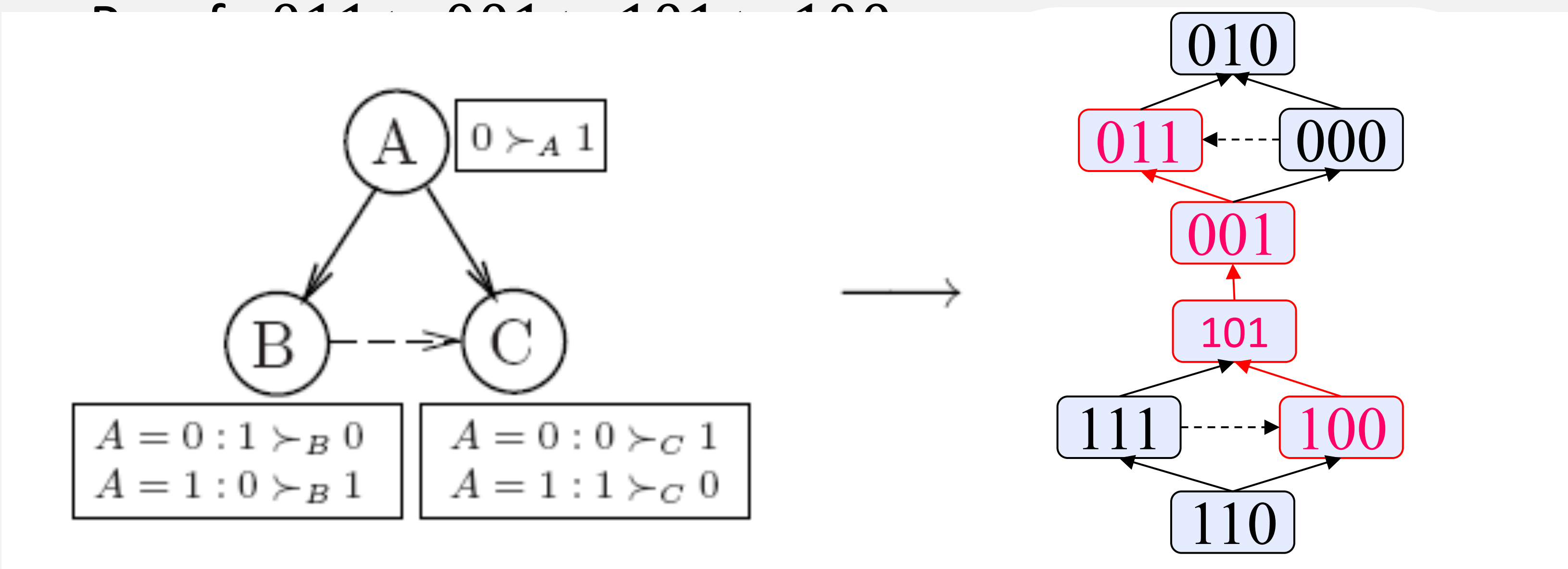
$$(a = 1 \wedge b = 0 \wedge c = 0) \Rightarrow EF(a = 0 \wedge b = 1 \wedge c = 1)$$

- 011 is preferred to 100

  Improving flipping sequence:

  100 →101 → 001 → 011

One of the proofs is chosen non-deterministically



Santhanam et al. AAAI 2010

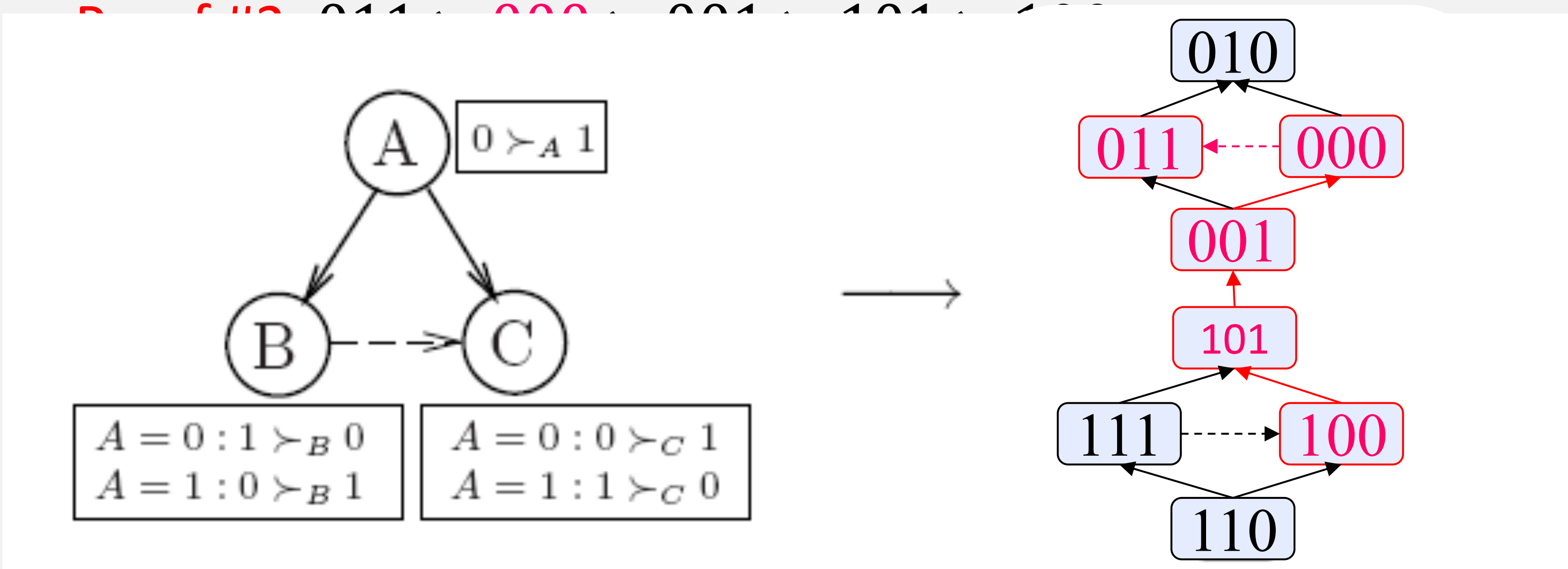
Obtaining a Proof of Dominance

$$(a = 1 \wedge b = 0 \wedge c = 0) \Rightarrow EF(a = 0 \wedge b = 1 \wedge c = 1)$$

- 011 is preferred to 100

  Improving flipping sequence:

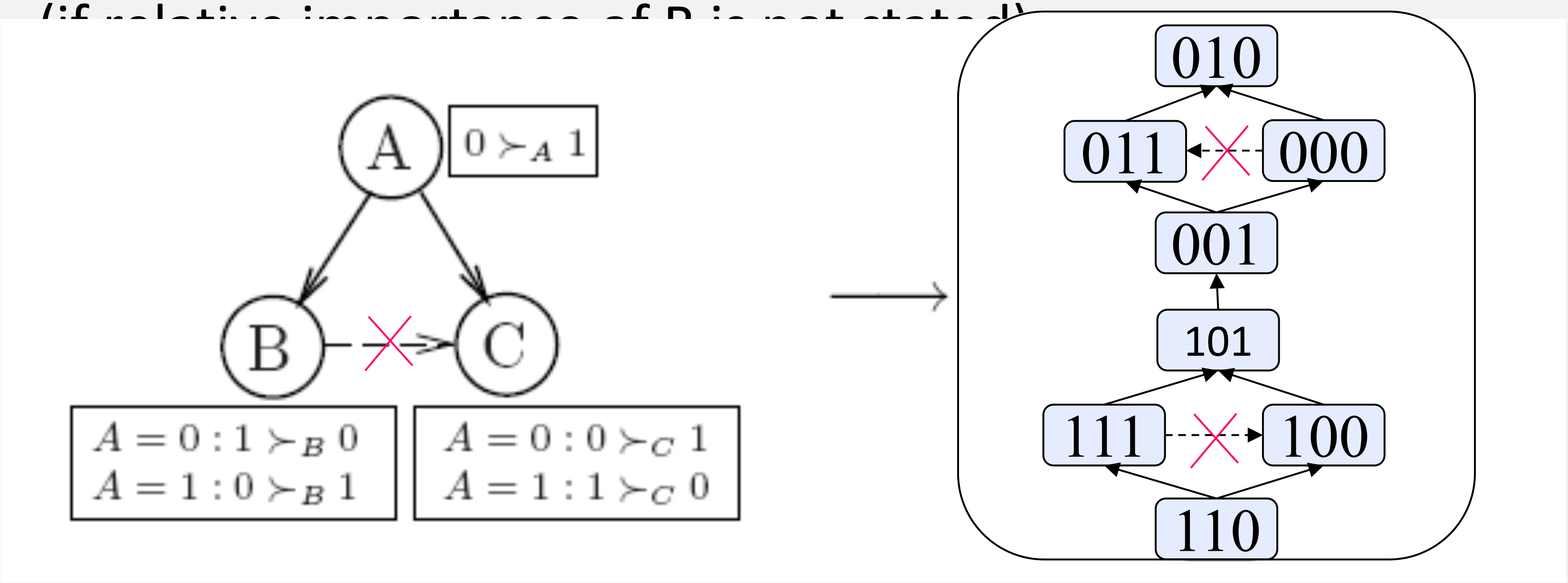
  100 →101 → 001 → 000 → 011



Santhanam et al. AAAI 2010

Non-dominance

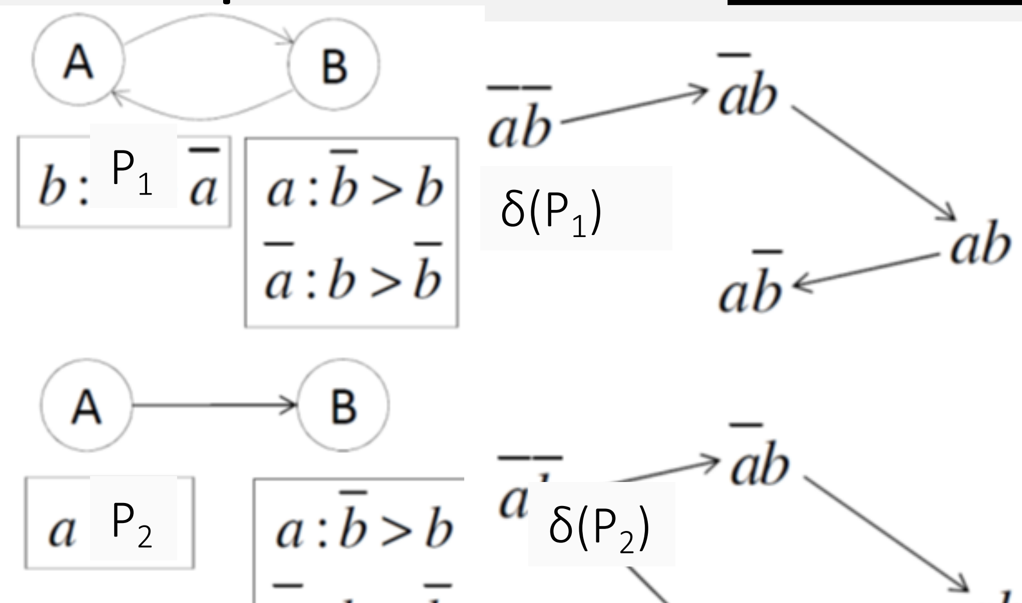
- 011 is not preferred to 000
(if relative importance of B is not stated)

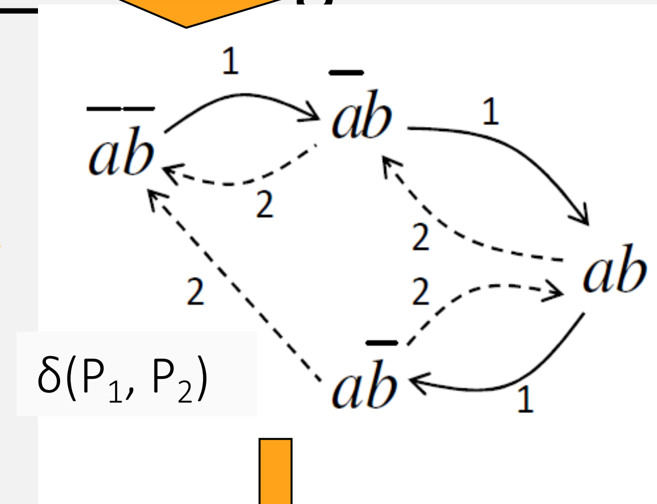


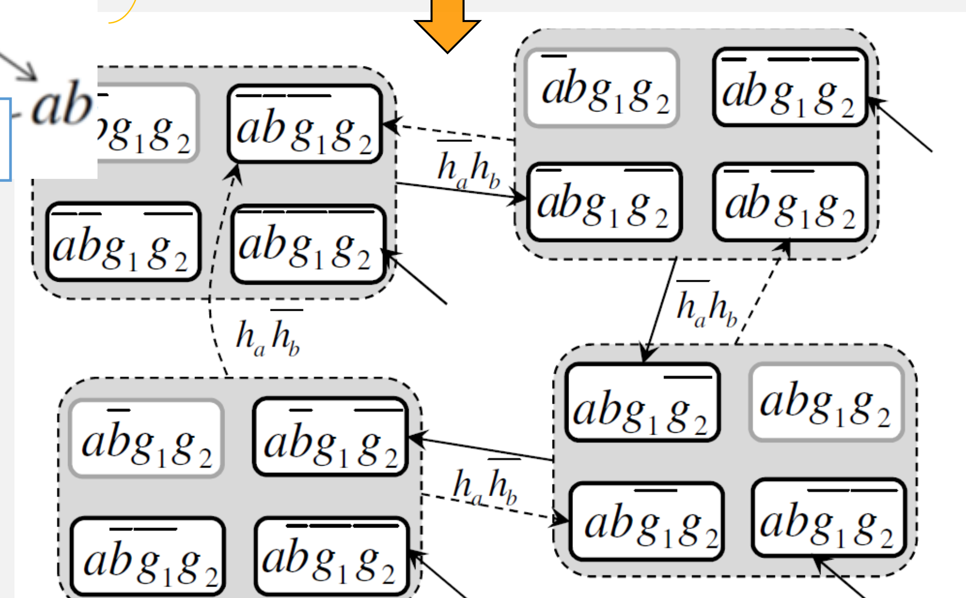
Santhanam et al. AAAI 2010

Equivalence and Subsumption Testing

Combined Induced Preference Graph







$$\varphi : \mathbf{AX}\ (\ g_1 \Rightarrow \mathbf{EX}\ \ \mathbf{E}\ [g_2\ \mathbf{U}\ \ (\psi\ \wedge\ g_2)\ ]\ )$$

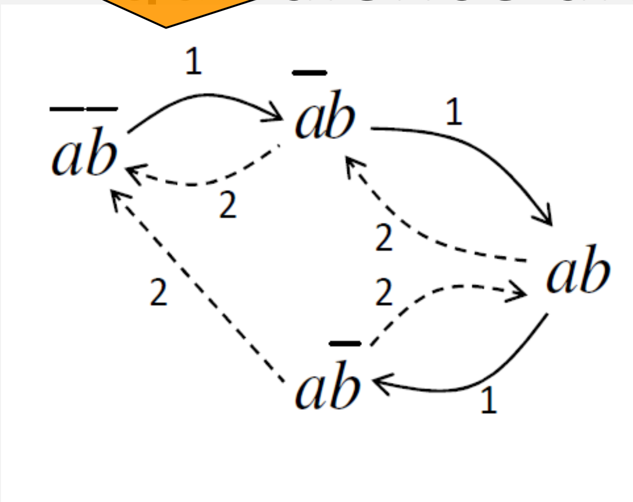
CTL Model Checking

Answer

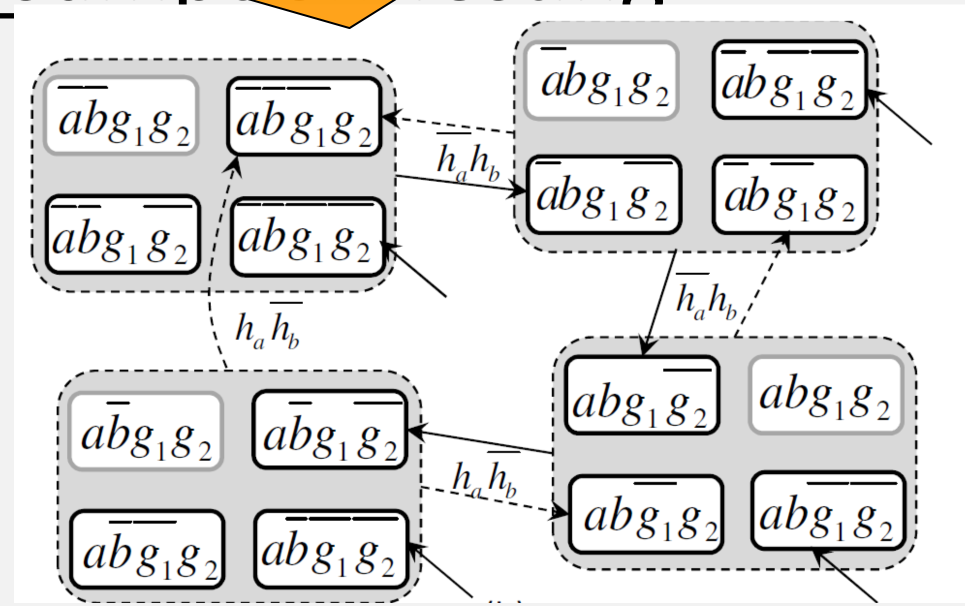
Santhanam et al. ADT 2013

Equivalence and Subsumption Testing

Combined Induced Preference Graph



Kripke Structure



te from which verification is done

$$\varphi : \mathbf{AX}\ (\ g_1 \Rightarrow \mathbf{EX}\ \mathbf{E}\ [g_2\ \mathbf{U}\ (\psi \wedge\ g_2)\ ]\ )$$

$$\neg\varphi : \mathbf{EX}\ (\ g_1 \wedge \mathbf{AX}\ \neg\mathbf{E}[\ g_2\ \mathbf{U}\ (\psi\ \wedge\ g_2)\ ]\ )$$

True $\Leftrightarrow$ $P_1 \sqsubseteq P_2$

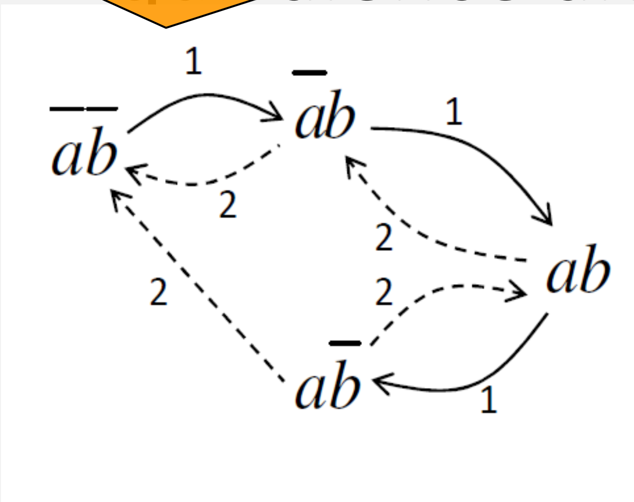
False $\Leftrightarrow$ $P_2 \not\sqsubseteq P_1$

Santhanam et al. ADT 2013

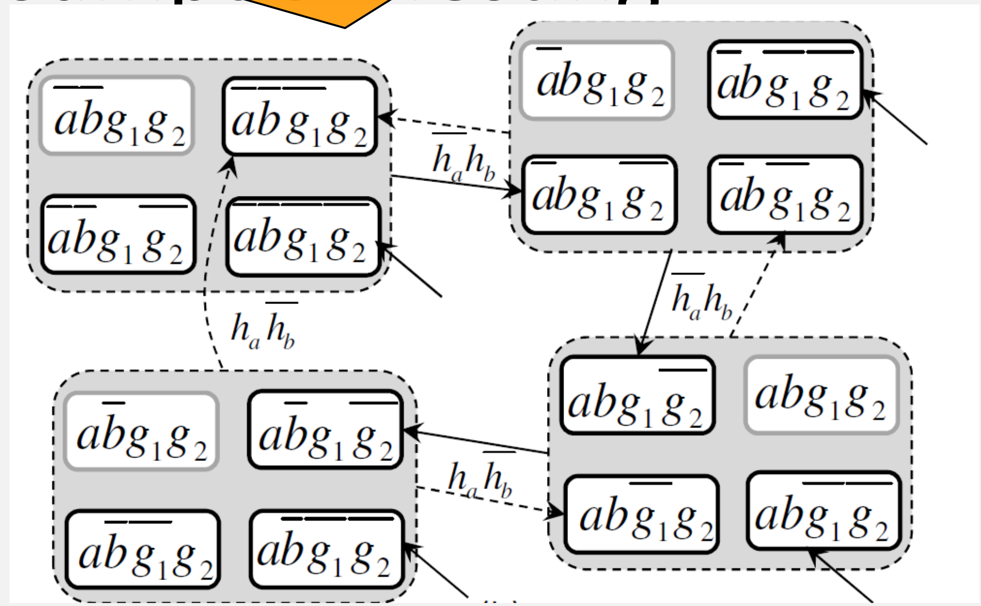
Model Checker returns $\overline{a}\,\overline{b}$ $a\overline{b}$roof

Equivalence and Subsumption Testing

Combined Induced Preference Graph



Kripke Structure



$$\varphi : \mathbf{AX}\ (\ g_1 \Rightarrow \mathbf{EX}\ \ \mathbf{E}\ [g_2\ \mathbf{U}\ \ (\psi\ \wedge\ g_2)\ ]\ )$$

$$\varphi' : \mathbf{AX}\ (\ g_2 \Rightarrow \mathbf{EX}\ \ \mathbf{E}[\ g_1\ \mathbf{U}\ \ (\psi\ \wedge\ g_1)\ ]\ )$$

True $\Leftrightarrow P_1 \sqsubseteq P_2$

True $\Leftrightarrow P_2 \sqsubseteq P_1$

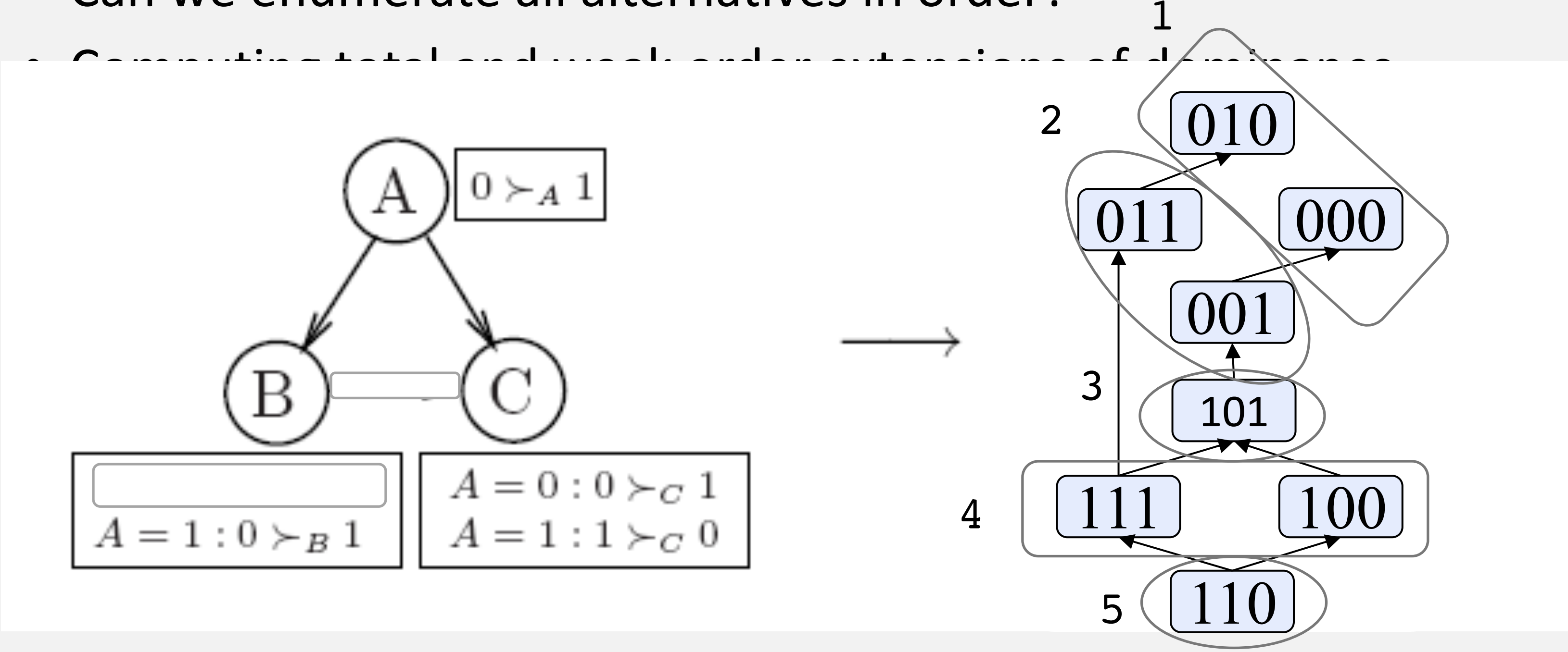
$P_1 \equiv P_2$

Santhanam et al. ADT 2013

Ordering : Finding the Next-preferred Alternative

- Which alternatives are most-preferred (non-dominated)?
- Can we enumerate all alternatives in order?
- Computing total and weak order extensions of dominance

How to deal with cycles?



We verify a sequence of reachability properties encoded in CTL

Acyclic Case: Oster et al. FACS 2012

Applications

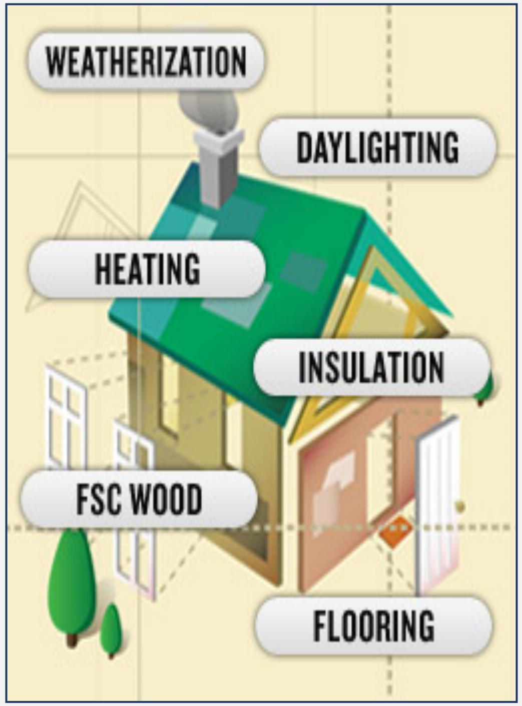
- Sustainable Design of Civil Infrastructure (e.g., Buildings, Pavements)
- Engineering Design (Aerospace, Mechanical)
- Strategic & mission critical decision making (Public policy, Defense, Security)
- Site Selection for Nuclear Waste and setting up new nuclear plants

---

- Software Engineering
  - Semantic Search
  - Code Search, Search based SE
  - Program Synthesis, Optimization
  - Test prioritization
  - Requirements Engineering
- Databases – Skyline queries
- Stable Marriage problems
- AI Planning, configuration
- Recommender Systems

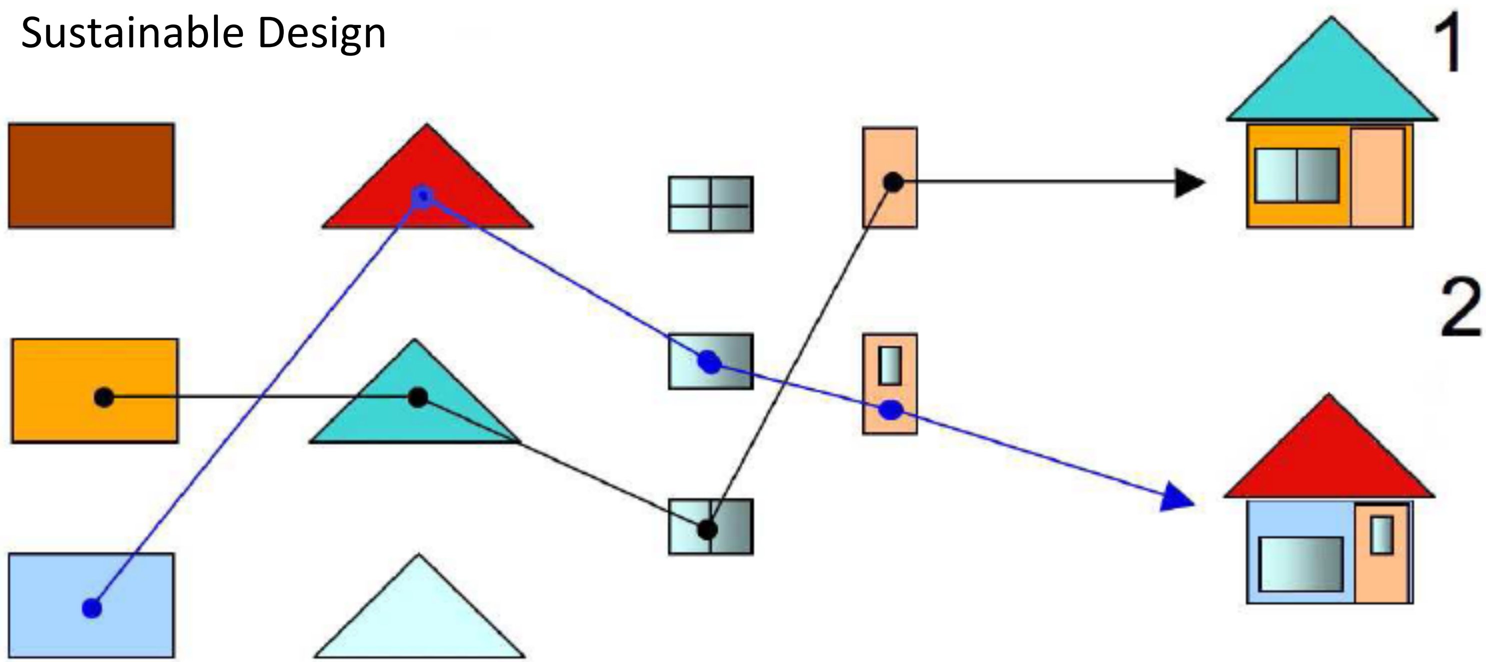
Applications

- Sustainable Design



Applications

- Sustainable Design



Function	Component	IC	FC	RE	TG
Heating	Electric	G	B	B	B
Heating	Gas	A	G	B	B
Heating	Solar	P	E	E	E
Flooring	Ceramic Tile	A	E	B	B
Flooring	Vinyl Tile	E	G	A	G
Flooring	Natural Cork	P	E	G	E
Siding	Brick&Mortar	P	E	P	B
Siding	Aluminum	G	G	G	A
Siding	Cedar	A	A	G	G

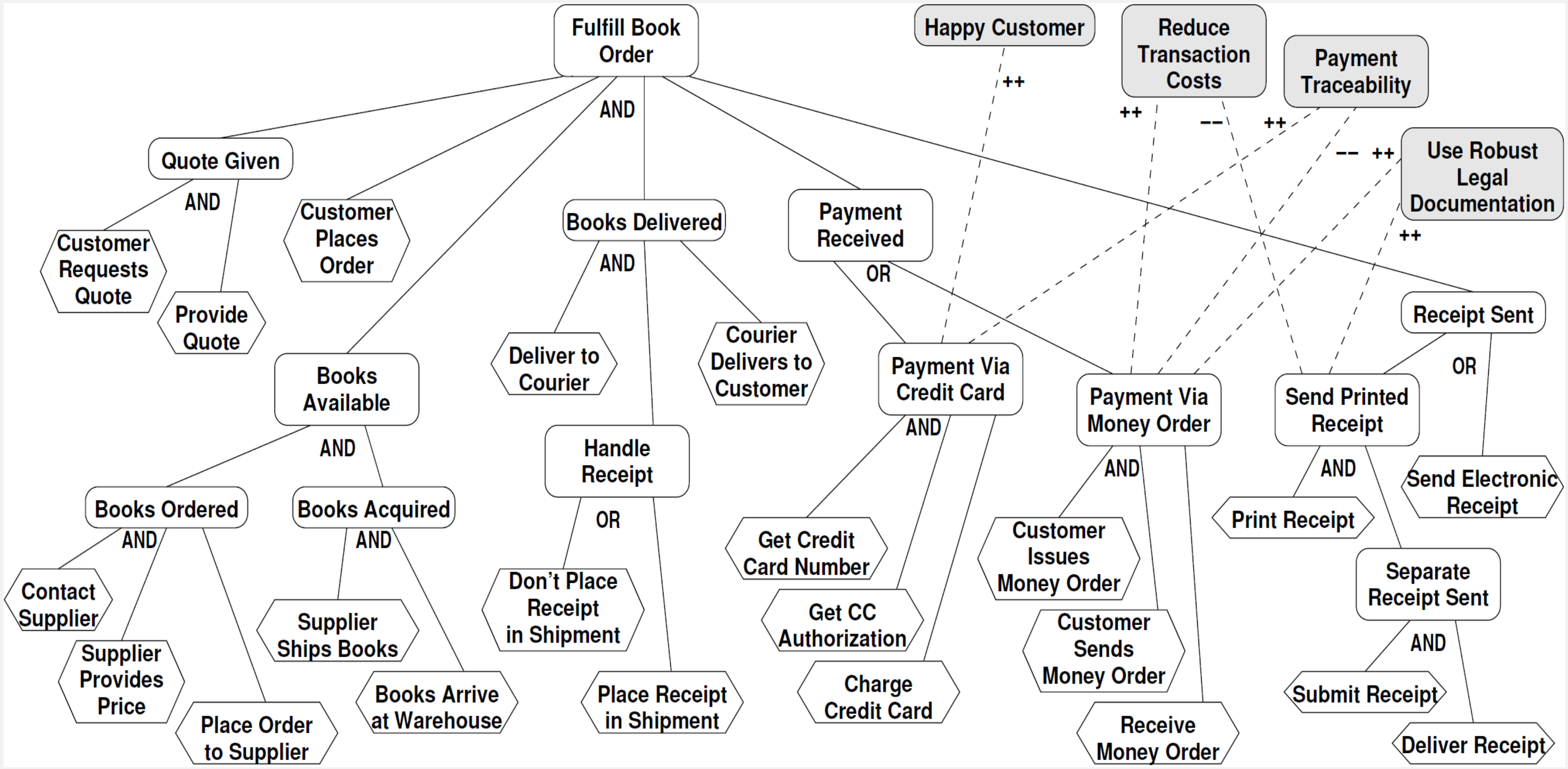
Table 1: Available Building Components in the Repository

Design	Heating	Flooring	Siding
D_1	Electric	Vinyl Tile	Aluminum
D_2	Gas	Ceramic Tile	Brick&Mortar
D_3	Gas	Vinyl Tile	Aluminum
D_4	Solar	Ceramic Tile	Brick&Mortar
D_5	Solar	Natural Cork	Aluminum

Table 2: Candidate Building Designs

- Goal Oriented Requirements Engineering

Oster et al. ASE 2011

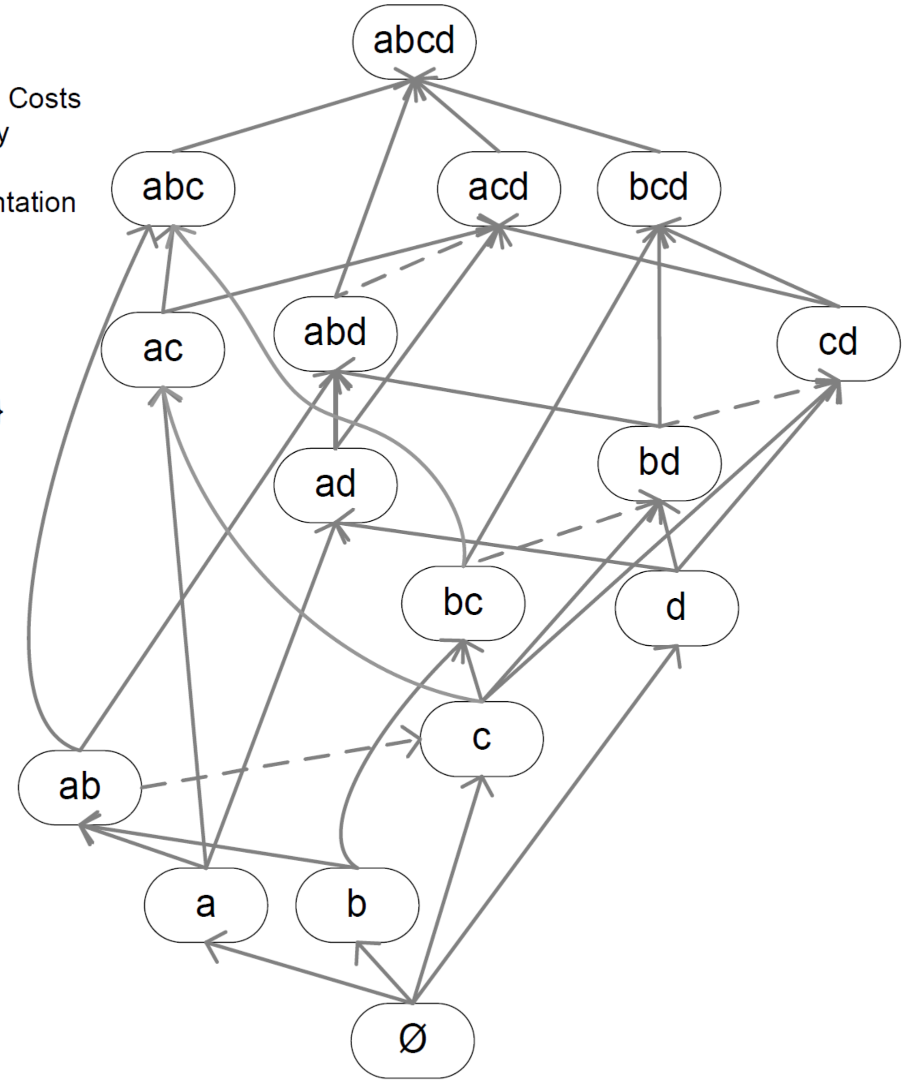


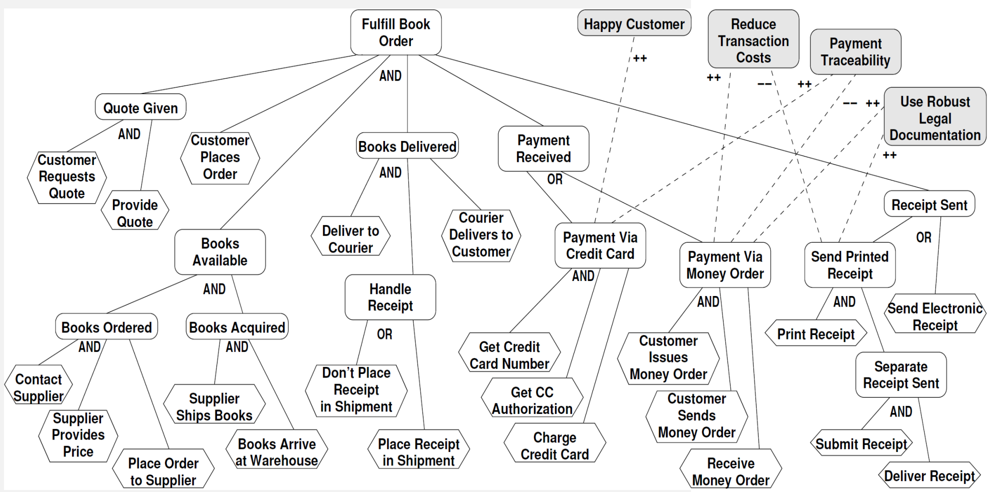
Goal oriented Requirements Engineering – CI-nets

a: Happy Customer
b: Reduce Transaction Costs
c: Payment Traceability
d: Use Robust Legal Documentation

<u>CI-net statements</u>

{d} ; {} : {c} ; {b}
{b} ; {a} : {d} ; {c}
{} ; {d} : {c} ; {a,b}





Oster et al. ASE 2011

Applications - Minimizing Credential Disclosure

Oster et al. FACS 2012

- User needs renter's insurance for new apartment
  - Which service to choose to get a quote?
  - Privacy issue – disclosure of sensitive credentials
- All services do the same tasks (from user's perspective) info:

#	*Name*	*Required Sensitive Information*
1	QuickQuote	Address, Bank Account #
2	InsureBest	Name, Address, Bank Routing #
3	EZCoverage	Name, Address
4	BankMatch	Bank Routing #

<u>User's Preferences:</u>

P1. If bank account number is disclosed, then I would rather give my address than bank routing number to the server

P2. If I have to disclose my address but not my name, then I would prefer to give my bank routing number rather than my bank account number

P3. If I don't need to disclose my bank account number, I will give my name and address instead of my bank routing number.

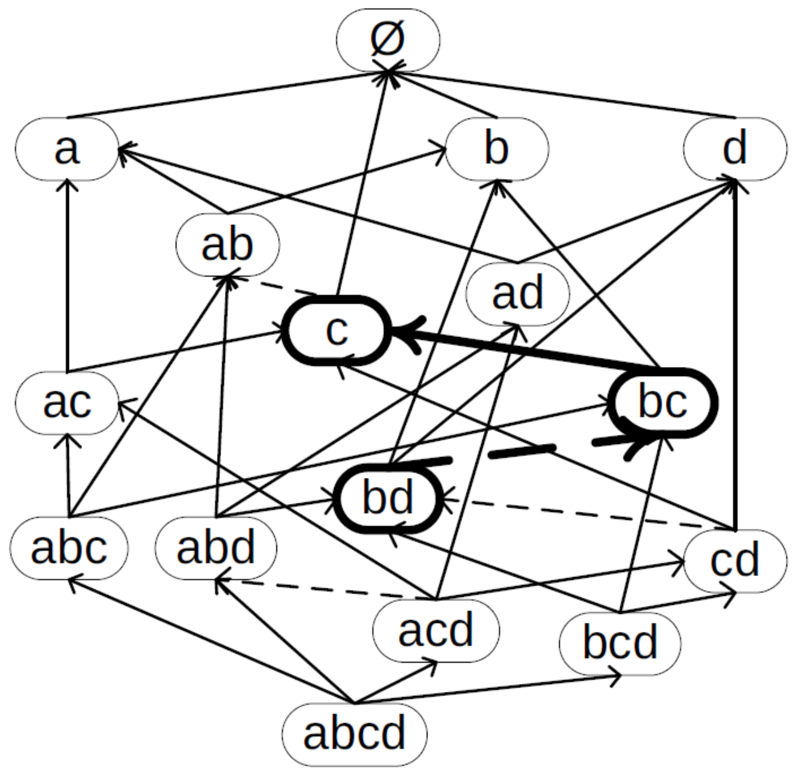
Applications - Minimizing Credential Disclosure

Oster et al. FACS 2012

- Finding a sequence of next-preferred
  - Suboptimal sequence of preferred sets of credentials

a = Name
b = Address
c = Bank Routing Number
d = Bank Account Number

P1. $\{d\}, \{\} \; : \; \{b\} \succ \{c\}$
P2. $\{b\}, \{a\} \; : \; \{c\} \succ \{d\}$
P3. $\{\}, \{d\} \; : \; \{a, b\} \succ \{c\}$



CRISNER Preference Reasoning Tool

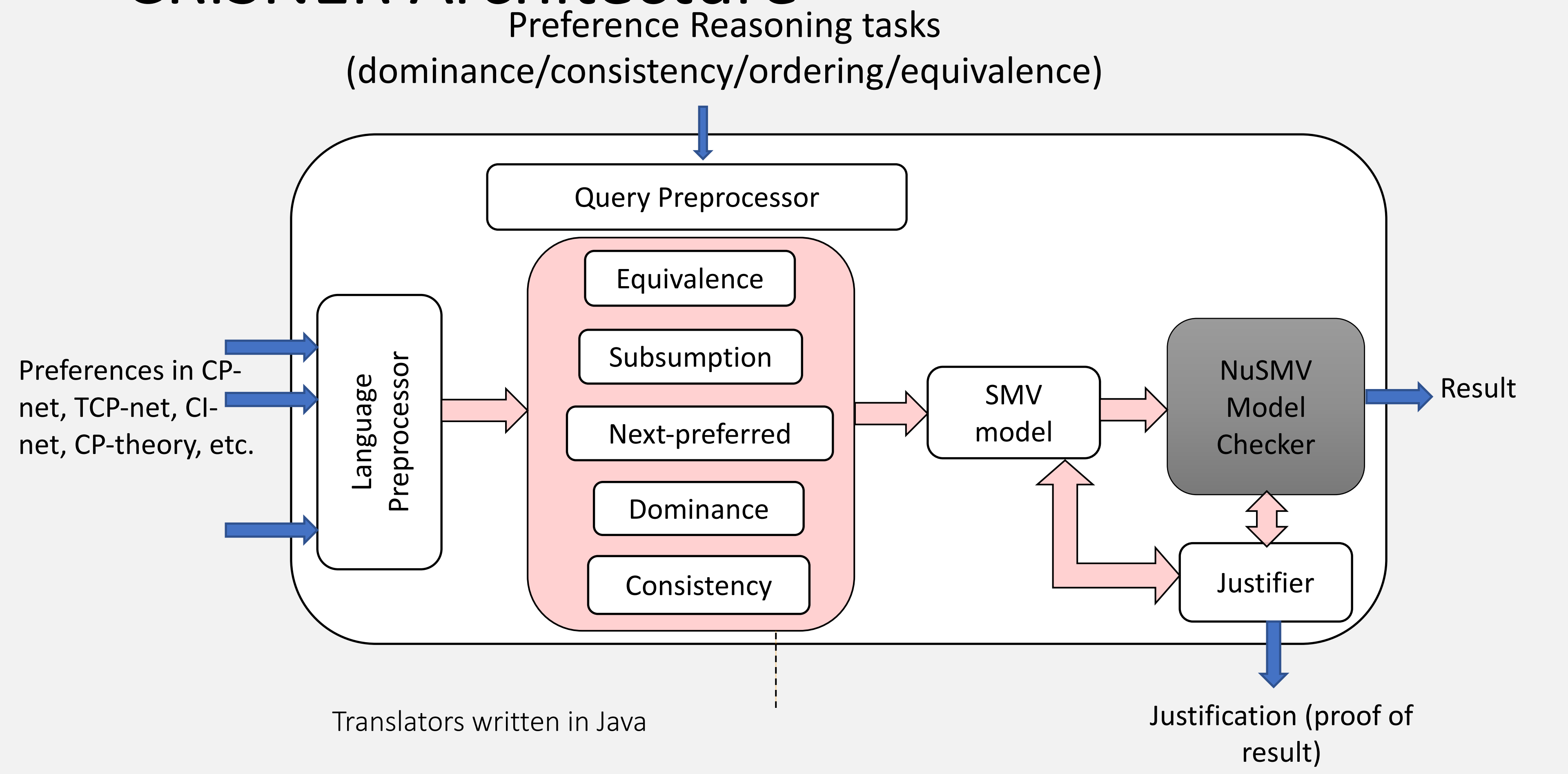
- CRISNER freely available at
  - http://www.ece.iastate.edu/~gsanthan/crisner.html
- Currently supports representing and reasoning with
  - CI-nets
  - CP-nets
- Reasoning tasks supported
  - Dominance Testing
  - Consistency
  - Next-preferred (for acyclic CP/CI-nets)
  - Support for Equivalence & Subsumption testing coming

CRISNER Architecture

- Architecture decouples preference reasoning from choice of
  - Model checker
  - Translation of preference
  - Preference languages
  - Modular design enables extension to other ceteris paribus languages, reasoning tasks and encodings

- Tool Dependencies
  - Model Checker – NuSMV or Cadence SMV
  - Java Runtime Environment

CRISNER Architecture



CRISNER Architecture

Tool Dependencies

- Model Checker – NuSMV or Cadence SMV
- Java Runtime Environment

Input/Output

- Preference specifications encoded in XML
  - Translated to SMV (Kripke model encoding)
- Parsers to translate output of model checker
- Iterative process to compute alternatives in order

Summary

I. Qualitative Preference Languages

- Representation : Syntax of languages CP-nets, TCP-nets, CI-nets, CP-Theories

II. Qualitative Preference Languages

- Ceteris Paribus semantics: the induced preference graph (IPG)
- Reasoning: Consistency, Dominance, Ordering, Equivalence & Subsumption
- Complexity of Reasoning

III. Practical aspects: Preference Reasoning via Model Checking

- From ceteris paribus semantics (IPG) to Kripke structures
- Specifying and verifying properties in temporal logic
- Reasoning tasks reduce to verification of temporal properties

Summary

IV. Applications

- *Engineering*: Civil, Software (SBSE, RE, Services), Aerospace, Manufacturing
- *Security*: Credential disclosure, Cyber-security
- *Algorithms*: Search, Stable Marriage, Allocation, Planning, Recommender systems
- *Environmental applications*: Risk Assessment, Policy decisions, Environmental impact, Computational Sustainability

V. CRISNER

- A *general, practically useful* Preference Reasoner for ceteris paribus languages
- Architecture
- Use of CRISNER in Security, Software Engineering