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

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
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. iPref-R Tool

- A tool that does well in practice for a known hard problem
- Architecture
- Demo
- Use of iPref-R in Security, Software Engineering
Broad view of Decision Theory

What is a decision?
Choosing from a set of alternatives $A$
Choice function: $\Phi(A) \subseteq A$

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

Qualitative

Walking

Driving

Carpooling

Quantitative

Walking = 0.7; Driving = 0.3

Walking = 0.6; Driving = 0.4

Loss of information regarding the incompleteness / imprecision of user preferences
Course selection - which course to take?

- **Subject?**
  - 572: AI
  - 509: SE
  - 586: NW

- **Instructor?**
  - 572: Gopal
  - 509: Tom
  - 586: Bob

- **# Credits?**
  - 572: 4
  - 509: 3
  - 586: 3

- Preference **variables or attributes** used to describe the domain
- Alternatives are **assignments** to preference variables
  - \( \alpha = (\text{instructor} = \text{Gopal}, \text{area} = \text{AI}, \text{credits} = 3) \)
- \( \alpha > \beta \) denotes that \( \alpha \) is **preferred** to \( \beta \)
Qualitative Preference Languages

Qualitative preferences

- Unconditional Preferences
  - TUP-nets [Santhanam et al., 2010]

- Conditional Preferences
  - Models dependencies

- Relative Importance
  - TCP-nets [Brafman et al. 2006]
  - CI-nets [Bouveret et al. 2009]

Idea is to represent comparative preferences
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

- Comparative preferences over:
  - Pairs of values of an attribute
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)

CP-Theories

- Similar to TCP-nets but...

Possible to express relative importance of one variable over a set of variables
Conditional Importance Networks (CI-nets) [Bouveret 2009]

**Cl-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** ($0 > 1$ or $1 > 0$)
  - Subsets preferred to supersets (or vice versa) by default
- Cl-net Statements are of the form $S^+, S^- : S_1 > S_2$
  - 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 $S_2$
Cl-nets  (fair division of goods among agents)

- Example:

\[ a = \text{Name} \]
\[ b = \text{Address} \]
\[ c = \text{Bank Routing Number} \]
\[ d = \text{Bank Account Number} \]

- P1. \( \{d\}, \emptyset : \{b\} \succ \{c\} \)
- P2. \( \{b\}, \{a\} : \{c\} \succ \{d\} \)
- P3. \( \emptyset, \{d\} : \{a, b\} \succ \{c\} \)

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]

In this tutorial ...

- We stick to CP-nets, TCP-nets and CI-nets.
- Overall approach is generic; extensible to all other ceteris paribus preference languages
Relative Expressivity of Preference Languages

Preferences over Multi-domain Variables

- CP-theories
- TCP-nets
- TUP-nets
- CP-nets

Preferences over (sets of) Binary Variables

- C$^3$I-nets
- CI-nets
Part II

Theoretical Aspects of Representing & Reasoning with Ceteris Paribus Preferences
Theoretical Aspects

Part II – Outline

• 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** $\delta(P) = G(V,E)$ of preference spec $P$:
  – Nodes $V$: set of alternatives
  – Edges $E$: $(\alpha, \beta) \in E$ iff there is a *flip induced by some statement in $P$* from $\alpha$ to $\beta$

$\delta(N)$ is acyclic (dominance is a strict partial order)

$\alpha > \beta$ iff there is a *path* in $\delta(N)$ from $\alpha$ to $\beta$ (serves as the *proof*)
Preference Semantics in terms of IPG

• \((\alpha, \beta) \in E\) iff there is a \textit{flip} from \(\alpha\) to \(\beta\) "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
• \((\alpha, \beta) \in E\) iff there is a statement in CP-net such that \(x_1 >_1 x'_1\) (\(x_1\) is preferred to \(x'_1\)) and …

- **V-flip**: *all other variables being equal*, \(\alpha(X_1)=x_1\) and \(\beta(X_1)=x'_1\)

\[A = 0:1 >_B 0\]
\[A = 1:0 >_B 1\]
\[A = 0:0 >_C 1\]
\[A = 1:1 >_C 0\]

**Ceteris paribus** *(all else being equal)*

Single **variable flip** – change value of 1 variable at a time
(α , β) ∈ E iff there is a statement in TCP-net such that \( x_1 \succ_1 x'_1 \) \((x_1 \text{ is preferred to } x'_1)\) and ...

- **V-flip**: all other variables being equal, \( \alpha(X_1)=x_1 \) and \( \beta(X_1)=x'_1 \)
- **I-flip**: all variables except those less important than \( X_1 \) being equal, \( \alpha(X_1)=x_1 \) and \( \beta(X_1)=x'_1 \)

**Multi-variable flip** – change values of multiple variables at a time
• **Recall:** 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 (conventional)
  – $S^+, S^- : S_1 > S_2$ 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 $\alpha, \beta \subseteq X$, $(\alpha, \beta) \in E$ ($\beta$ preferred to $\alpha$) iff
  
  - **M-flip** : all other variables being equal, $\alpha \subset \beta$
  
  - **CI-flip** : there is a CI-net statement s.t. $S^+, S^- : S_1 > S_2$ and $\alpha, \beta$ satisfy $S^+, S^-$ and $\alpha$ satisfies $S^+$ and $\beta$ satisfies $S^-$. 

---

**Flips for a CI-net** [Bouveret 2009]

---

**Representing and Reasoning with Qualitative Preferences** - Ganesh Ram Santhanam, Iowa State University.
Flips for a CI-net [Bouveret 2009]

- For $\alpha, \beta \subseteq X$, $(\alpha, \beta) \in E$ ($\beta$ preferred to $\alpha$) iff
  - **M-flip**: all other variables being equal, $\alpha \subset \beta$
  - **CI-flip**: there is a CI-net statement $S^+, S^- : S_1 > S_2$ s.t.
    - $\alpha, \beta$ satisfy $S^+, S^-$ and $\alpha$ satisfies $S^+$ and $\beta$ satisfies $S^-$. 

- Example:
  
 \[
  \begin{align*}
  a &= \text{Name} \\
  b &= \text{Address} \\
  c &= \text{Bank Routing Number} \\
  d &= \text{Bank Account Number}
  \end{align*}
  \]

  \[
  \begin{align*}
  \text{P1. } \{d\}, \emptyset & : \{b\} > \{c\} \\
  \text{P2. } \{b\}, \{a\} & : \{c\} > \{d\} \\
  \text{P3. } \emptyset, \{d\} & : \{a, b\} > \{c\}
  \end{align*}
  \]
C³I-nets express *preference over subsets* similar to Cl-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 $\alpha, \beta \subseteq X$, $(\alpha, \beta) \in E$ ($\beta$ preferred to $\alpha$) iff

- **M-flip**: all other variables being equal, $|\alpha| < |\beta|$
- **Cl-flip**: there is a Cl-net statement s.t. $S^+, S^- : S_1 \succ S_2$ and $\alpha, \beta$ satisfy $S^+, S^-$ and $\alpha$ satisfies $S^+$ and $\beta$ satisfies $S^-$. Extra cardinality constraint to enable dominance
For $\alpha, \beta \subseteq X$, $(\alpha, \beta) \in E$ ($\beta$ preferred to $\alpha$) iff

- **$M$-flip**: $\alpha \subset \beta$ (all other variables being equal)
- **$CI$-flip**: there is a Cl-net statement $S^+, S^- : S_1 > S_2$ s.t. $\alpha, \beta$ satisfy $S^+, S^-$ and $\alpha$ satisfies $S^+$ and $\beta$ satisfies $S^-$. 
- **$C$-flip**: $|\alpha| < |\beta|$

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

Now we turn to the Reasoning Tasks:
- Dominance & Consistency
- Equivalence & Subsumption
- Ordering

We describe reasoning tasks only in terms of verifying properties of the IPG.

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

Dominance relation:
- $\alpha \succ \beta$ iff there exists a sequence of flips from $\beta$ to $\alpha$
- Property to verify: *Existence of path in IPG* from $\beta$ to $\alpha$

Consistency:
- A set of preferences is *consistent* if $\succ$ is a strict partial order
- Property to verify: *IPG is acyclic*

Ordering: ?
- *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

<table>
<thead>
<tr>
<th>Reasoning Task</th>
<th>Computation Strategy: Property of IPG to check</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dominance: $\alpha \succ \beta$</td>
<td>Is $\beta$ reachable from $\alpha$?</td>
<td></td>
</tr>
<tr>
<td>Consistency of a set of preferences ($P$)</td>
<td>Is the IPG of $P$ acyclic?</td>
<td>Satisfiability of the dominance relation; strict partial order</td>
</tr>
<tr>
<td>Equivalence of two sets of preferences $P_1$ and $P_2$</td>
<td>Are the IPGs of $P_1$ and $P_2$ reachability-equivalent?</td>
<td></td>
</tr>
<tr>
<td>Subsumption of one set of preference ($P_1$) by another ($P_2$)</td>
<td>If $\beta$ reachable from $\alpha$ in the IPG of $P_1$, does the same hold in the IPG of $P_2$?</td>
<td></td>
</tr>
<tr>
<td>Ordering of alternatives</td>
<td>Iterative verification of the IPG for the non-existence of the non-dominated alternatives</td>
<td>Iterative modification of the IPG to obtain next set of non-dominated alternatives</td>
</tr>
</tbody>
</table>
Complexity of Dominance [Goldsmith et al. 2008]

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

- \( \alpha = (A = 1, B = 0, C = 0) \)
- \( \beta = (A = 0, B = 1, C = 1) \)
- \( \alpha \succ \beta \) – Why?

*Dominance testing reduces to STRIPS planning (Goldsmith et al. 2008)*
# Complexity of Reasoning Tasks

<table>
<thead>
<tr>
<th>Reasoning Task</th>
<th>Complexity</th>
<th>Work by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dominance: $\alpha &gt; \beta$</td>
<td>PSPACE-complete</td>
<td>Goldsmith et al. 2008</td>
</tr>
<tr>
<td>Consistency of a set of preferences $(P)$</td>
<td>PSPACE-complete</td>
<td>Goldsmith et al. 2008</td>
</tr>
<tr>
<td>Equivalence of two sets of preferences $P_1$ and $P_2$</td>
<td>PSPACE-complete</td>
<td>Santhanam et al. 2013</td>
</tr>
<tr>
<td>Subsumption of one set of preference $(P_1)$ by another $(P_2)$</td>
<td>PSPACE-complete</td>
<td>Santhanam et al. 2013</td>
</tr>
<tr>
<td>Ordering of alternatives</td>
<td>NP-hard</td>
<td>Brafman et al. 2011</td>
</tr>
</tbody>
</table>
Part III
Practical Aspects of Reasoning with Ceteris Paribus Preferences
Part III – Outline

• Two Sound and Complete Reasoning Approaches:
  – Logic Programming based
    • 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
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]

- **Model Checking**: Given a desired property $\varphi$, (typically expressed as a temporal logic formula), and a (Kripke) structure $M$ with initial state $s$, decide if $M, s \models \varphi$

- **Active area of research in formal methods, AI** (SAT solvers)

- **Broad range of applications**: hardware and software verification, security..

- **Temporal logic languages**: CTL, LTL, $\mu$-calculus, etc.

- **Many model checkers available**: SMV, NuSMV, Spin, etc.

**Advantages of Model Checking:**
1. Formal Guarantees
2. Justification of Results
• Key Idea:

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 temporal logic properties over model
Overview of Approach

**Alternatives Attributes Preferences (Ceteris Paribus Statements)**

**ENCODE**

**Kripke Structure**

**Translate**

**Reasoning Task** (e.g., Dominance: $\alpha > \beta$?)

**Temporal Logic Model Checker**

**Answer**

*States correspond to alternatives; Transitions correspond to flips (induced preferences)*

Santhanam et al. AAAI 2010
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 a labeling (interpretation) function that maps each node to a set of atomic propositions $AP$ that hold in the corresponding state

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 $\varphi$ holds in some state in some path beginning at $s$
Let $P = \{p_i\}$ be a set of ceteris paribus preference statements on a set of preference variables $X = \{x_1, x_2, \ldots\}$

Reasoning Strategy:

- Construct a Kripke model $K_P = (S, S_0, T, L)$ using variables $Z$
  - $Z = \{z_i \mid x_i \in X\}$, with each variable $z_i$ having same domain $D_i$ as $x_i$
  - $K_P$ must mimic the IPG is 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
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 ensure ceteris paribus condition?
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?
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

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

- Auxiliary edge labels don’t contribute to the state space.
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) 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

- For each statement $p$ of the form $\rho : x_i = v_i \succ x_i \ x_i = v'_i$

  where $\rho$ is the “condition” part, guard condition is

\[
G(p) = Allow(p) \land Restrict(p) \text{ s.t.}
\]

- $Allow(p) := \rho \land z_i = v'_i \land h_i = 1$
- $Restrict(p) := \land_{x_j \in X \setminus \{x_i\}} h_j = 0$

---

**Condition**  
**Preference**  
**Ceteris paribus**
Encoding CP-net semantics

Direct & succinct

Kripke Structure

Induced Preference Graph
Encoding CP-net semantics

Representing and Reasoning with Qualitative Preferences - Ganesh Ram Santhanam, Iowa State University.
Encoding TCP-net Semantics

TCP-nets: Same overall idea as CP-nets

- Additional rule for encoding simple relative importance

![Diagram showing the relationships between different states of TCP-nets: Functional, LO, Unavailable; Functional, LO, Official fix; Unproven, LO, Official fix; Functional, HI, Unavailable; Functional, HI, Official fix; Unproven, HI, Official fix; Unproven, LO, Unavailable; Unproven, HI, Unavailable. The diagram uses symbols to represent the relationships and priorities among these states.]
Encoding CP-theory Semantics

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

- **Functional, LO, Unavailable**
- **Functional, LO, Official fix**
- **Unproven, LO, Official fix**
- **Functional, HI, Unavailable**
- **Unproven, HI, Unavailable**
- **Functional, HI, Official fix**
- **Unproven, HI, Official fix**

Edges:
- E (Functional > Unproven)
- A (LO > HI)
- E (Functional: Unavailable > Official fix)
Next:
Specifying and Verifying Properties in Temporal Logic
Translating Reasoning Tasks into 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

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

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

  E [ψ₁U ψ₂] if there exists a path $s = s_1 \rightarrow s_2 \ldots$ such that $∃i \geq 1 : s_i$ satisfies $ψ₂$, and $∀j < i : s_j$ satisfies $ψ₁$

- 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 $\alpha$ and $\beta$, how to check if $\alpha > \beta$?

- Let $\varphi_\alpha$ be a formula that holds in the state corresponding to $\alpha$
- Let $\varphi_\beta$ be a formula that holds in the state corresponding to $\beta$

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

- $a \text{ state in which } \varphi_\beta \text{ holds is reachable from a state in which } \varphi_\alpha \text{ holds}$
- $\alpha > \beta$ iff the model checker NuSMV can verify $\varphi_\alpha \rightarrow \text{EF}\varphi_\beta$ (SAT)
- When queried with $\neg(\varphi_\alpha \rightarrow \text{EF}\varphi_\beta)$, if indeed $\alpha > \beta$, then model checker produces a proof of $\alpha > \beta$ (flipping sequence)
- Experiments show feasibility of method for 100 var. in seconds

Santhanam et al. AAAI 2010
Obtaining a Proof of Dominance

• 011 is preferred to 100

\[ (a = 1 \land b = 0 \land c = 0) \implies EF(a = 0 \land b = 1 \land c = 1) \]

**Improving** flipping sequence:

\[ 100 \rightarrow 101 \rightarrow 001 \rightarrow 011 \]

Proof: \( 011 > 001 > 101 > 100 \)

One of the proofs is chosen non-deterministically
Obtaining a Proof of Dominance

- 011 is preferred to 100
  
  Improving flipping sequence:
  
  100 → 101 → 001 → 000 → 011

  Proof #2: 011 > 000 > 001 > 101 > 100

\[
a = 1 \land b = 0 \land c = 0 \Rightarrow EF(a = 0 \land b = 1 \land c = 1)
\]
Non-dominance

- 011 is not preferred to 000 (if relative importance of B is not stated)
Equivalence and Subsumption Testing

\[ \varphi : \text{AX} \left( g_1 \Rightarrow \text{EX} \ E \left[ g_2 \cup (\psi \land g_2) \right] \right) \]

\[ \delta(P_1) \]

\[ \delta(P_2) \]

\[ \delta(P_1, P_2) \]

\[ \text{Answer} \]

Santhanam et al. ADT 2013
Equivalence and Subsumption Testing

Combined Induced Preference Graph

\[ \overrightarrow{ab} \xrightarrow{1} \overrightarrow{ab} \xrightarrow{2} \overrightarrow{ab} \xrightarrow{1} \overrightarrow{ab} \]

Kripke Structure

\[ \overrightarrow{abg_1g_2} \overrightarrow{abg_1g_2} \]
\[ \overrightarrow{abg_1g_2} \overrightarrow{abg_1g_2} \]
\[ \overrightarrow{abg_1g_2} \overrightarrow{abg_1g_2} \]
\[ \overrightarrow{abg_1g_2} \overrightarrow{abg_1g_2} \]
\[ \overrightarrow{abg_1g_2} \overrightarrow{abg_1g_2} \]

State from which verification is done

\[ \varphi : \text{AX} \left( g_1 \Rightarrow \text{EX} \ E \ [g_2 \cup (\psi \land g_2)] \right) \]
\[ \neg \varphi : \text{EX} \left( g_1 \land \text{AX} \ \neg E \ [g_2 \cup (\psi \land g_2)] \right) \]

True $\iff P_1 \sqsubseteq P_2$

False $\iff P_2 \not\sqsubseteq P_1$

Model Checker returns $\overrightarrow{ab} \rightarrow \overrightarrow{ab}$ as proof

Santhanam et al. ADT 2013
Equivalence and Subsumption Testing

Combined Induced Preference Graph

Kripke Structure

\[ \varphi : \text{AX} \left( g_1 \Rightarrow \text{EX} \ E \left[ g_2 \ U \ (\psi \ \land \ g_2) \right] \right) \]

True \iff P_1 \sqsubseteq P_2

\[ \varphi' : \text{AX} \left( g_2 \Rightarrow \text{EX} \ E \left[ g_1 \ U \ (\psi \ \land \ g_1) \right] \right) \]

True \iff P_2 \sqsubseteq P_1

P_1 \equiv P_2

Santhanam et al. ADT 2013

Representing and Reasoning with Qualitative Preferences - Ganesh Ram Santhanam, Iowa State University.
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
Part IV – Applications

Part IV
Applications
Applications

- Sustainable Design of Civil Infrastructure (e.g., Buildings, Pavements)
- Engineering Design (Aerospace, Mechanical)
- Strategic & mission critical decision making (Public policy, Defense, Security)
- Chemical and Nano-Toxicology
- 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
Sustainable Design
Applications

- Sustainable Design

Table 1: Available Building Components in the Repository

<table>
<thead>
<tr>
<th>Function</th>
<th>Component</th>
<th>IC</th>
<th>FC</th>
<th>RE</th>
<th>TG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heating</td>
<td>Electric</td>
<td>G</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>Heating</td>
<td>Gas</td>
<td>A</td>
<td>G</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>Heating</td>
<td>Solar</td>
<td>P</td>
<td>E</td>
<td>E</td>
<td>E</td>
</tr>
<tr>
<td>Flooring</td>
<td>Ceramic Tile</td>
<td>A</td>
<td>E</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>Flooring</td>
<td>Vinyl Tile</td>
<td>E</td>
<td>G</td>
<td>A</td>
<td>G</td>
</tr>
<tr>
<td>Flooring</td>
<td>Natural Cork</td>
<td>P</td>
<td>E</td>
<td>G</td>
<td>E</td>
</tr>
<tr>
<td>Siding</td>
<td>Brick&amp;Mortar</td>
<td>P</td>
<td>E</td>
<td>P</td>
<td>B</td>
</tr>
<tr>
<td>Siding</td>
<td>Aluminum</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>A</td>
</tr>
<tr>
<td>Siding</td>
<td>Cedar</td>
<td>A</td>
<td>A</td>
<td>G</td>
<td>G</td>
</tr>
</tbody>
</table>

Table 2: Candidate Building Designs

<table>
<thead>
<tr>
<th>Design</th>
<th>Heating</th>
<th>Flooring</th>
<th>Siding</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>Electric</td>
<td>Vinyl Tile</td>
<td>Aluminum</td>
</tr>
<tr>
<td>$D_2$</td>
<td>Gas</td>
<td>Ceramic Tile</td>
<td>Brick&amp;Mortar</td>
</tr>
<tr>
<td>$D_3$</td>
<td>Gas</td>
<td>Vinyl Tile</td>
<td>Aluminum</td>
</tr>
<tr>
<td>$D_4$</td>
<td>Solar</td>
<td>Ceramic Tile</td>
<td>Brick&amp;Mortar</td>
</tr>
<tr>
<td>$D_5$</td>
<td>Solar</td>
<td>Natural Cork</td>
<td>Aluminum</td>
</tr>
</tbody>
</table>
Goal Oriented Requirements Engineering

Fulfill Book Order
- Quote Given
  - Customer Requests Quote
    - Provide Quote
      - Customer Places Order
        - Books Available
          - Books Ordered
            - Contact Supplier
              - Supplier Provides Price
                - Place Order to Supplier
                  - Supplier Ships Books
                    - Books Arrive at Warehouse
                      - Books Acquired
                        - AND
                          - Handle Receipt
                            - Don't Place Receipt in Shipment
                              - Place Receipt in Shipment
                                - AND
                                  - Payment Received
                                    - Payment Via Credit Card
                                      - Get Credit Card Number
                                        - Get CC Authorization
                                          - Charge Credit Card
                                            - Receive Money Order
                                              - Submit Receipt
                                                - Deliver Receipt
                                                  - AND
                                                    - Customer Sends Money Order
                                                      - Print Receipt
                                                        - AND
                                                          - Separate Receipt Sent
                                                            - Print Receipt
                                                              - AND
                                                                - Send Electronic Receipt
                                                                  - Send Printed Receipt
                                                                    - OR
                                                                      - Send Printed Receipt
                                                                        - OR
                                                                          - Use Robust Legal Documentation
                                                                            - ++
                                                                               - Reduce Transaction Costs
                                                                                 - ++
                                                                                    - Payment Traceability
                                                                                           - ++
                                                                                               - ++
                                                                                                   - --
                                                                                                       - ++
                                                                                                           ++
                                                                                                               ++
                                                                                                                   ++
                                                                                                                                   ++
                                                                                   Oster et al. ASE 2011
Goal oriented Requirements Engineering – CI-nets

Cl-net statements
{d}; {} : {c}; {b}
{b}; {a}; {d}; {c}
{} ; {d}; {c}; {a, b}
Applications - Minimizing Credential Disclosure

• 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:

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Required Sensitive Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>QuickQuote</td>
<td>Address, Bank Account #</td>
</tr>
<tr>
<td>2</td>
<td>InsureBest</td>
<td>Name, Address, Bank Routing #</td>
</tr>
<tr>
<td>3</td>
<td>EZCoverage</td>
<td>Name, Address</td>
</tr>
<tr>
<td>4</td>
<td>BankMatch</td>
<td>Bank Routing #</td>
</tr>
</tbody>
</table>

User’s Preferences:

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.
• Finding a sequence of next-preferred
  – Suboptimal sequence of preferred sets of credentials can compromise privacy,
    *when it could have been avoided*

\[
a = \text{Name} \\
b = \text{Address} \\
c = \text{Bank Routing Number} \\
d = \text{Bank Account Number}
\]

\[
P1. \{d\}, \emptyset : \{b\} \succ \{c\} \\
P2. \{b\}, \{a\} : \{c\} \succ \{d\} \\
P3. \emptyset, \{d\} : \{a, b\} \succ \{c\}
\]
iPref-R Preference Reasoning Tool

- α-version of iPref-R freely available at
  - http://fmg.cs.iastate.edu/project-pages/preference-reasoner/


- Currently supports representing and reasoning with
  - CI-nets
  - CP-nets
  - Support for other languages in progress

- Reasoning tasks supported
  - Dominance Testing
  - Consistency
  - Next-preferred (for acyclic CP/CI-nets)
  - Support for Equivalence & Subsumption testing coming
iPref-R Architecture

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

Preference Reasoning tasks
(dominance/consistency/ordering/equivalence)

Query Preprocessor

Equivalence
Subsumption
Next-preferred
Dominance
Consistency

SMV model

NuSMV Model Checker

Justifier

Justification
(proof of result)

Translators written in Java

Preferences in CP-net, TCP-net, CI-net, CP-theory, etc.

Language Preprocessor

Result
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
   - **Translating Reasoning Tasks** into Temporal Logic Properties
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. iPref-R Tool

- A tool that does well in practice for a known hard problem
- Architecture
- Demo
- Use of iPref-R in Security, Software Engineering


• Ganesh Ram Santhanam, Zachary J. Oster, Samik Basu: **Identifying a preferred countermeasure strategy for attack graphs.** CSIIRW 2013


• Ganesh Ram Santhanam and Kasthurirangan Gopalakrishnan. **Pavement Life-Cycle Sustainability Assessment and Interpretation Using a Novel Qualitative Decision Procedure.** Journal of Computing in Civil Engineering 2013.

• Zachary J. Oster, Ganesh Ram Santhanam, Samik Basu: **Automating analysis of qualitative preferences in goal-oriented requirements engineering.** ASE 2011.

• G. Brewka, M. Truszczynski; S. Woltran. **Representing Preferences Among Sets.** AAAI 2010


• G. Brewka. **Answer Sets and Qualitative Decision Making,** Synthese 2005.


• Ronen I. Brafman, Enrico Pilotto, Francesca Rossi, Domenico Salvagnin, Kristen Brent Venable, Toby Walsh: **The Next Best Solution.** AAAI 2011.


• Ronen I. Brafman, Carmel Domshlak, Solomon Eyal Shimony, Y. Silver: **Preferences over Sets.** AAAI 2006.

• Ronen I. Brafman, Yuri Chernyavsky: **Planning with Goal Preferences and Constraints.** ICAPS 2005.
References

- Shirin Sohrabi, Sheila A. McIlraith: **Preference-Based Web Service Composition: A Middle Ground between Execution and Search.** International Semantic Web Conference (1) 2010.
- Jorge A. Baier, Sheila A. McIlraith: **Planning with Preferences.** AI Magazine 2008.
- Zachary J. Oster: **Reasoning with qualitative preferences to develop optimal component-based systems.** ICSE 2013.
- Walid Trabelsi, Nic Wilson, Derek G. Bridge: **Comparative Preferences Induction Methods for Conversational Recommenders.** ADT 2013.
- Francesca Rossi, Kristen Brent Venable, Toby Walsh: Preferences in Constraint Satisfaction and Optimization. AI Magazine 2008.
Thank you

Collaborators
Dr. Vasant Honavar
Dr. Samik Basu
Dr. Giora Slutzki
Dr. Kasthurirangan Gopalakrishnan
Dr. Robyn Lutz
Dr. Zachary Oster
Carl Chapman
Katerina Mitchell

Acknowledgements
NSF Grants IIS 0711356, CCF 0702758, CCF 1143734, CNS 1116050