Representing & Reasoning with Qualitative Preferences: Tools and Applications

Ganesh Ram Santhanam\textsuperscript{1}, Samik Basu\textsuperscript{1} & Vasant Honavar\textsuperscript{2}
\textsuperscript{1}Iowa State University, \textsuperscript{2}Penn State University

gsanthan@iastate.edu
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
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

**Quantitative**

- Walking = 0.7; Driving = 0.3
- Walking = 0.6; Driving = 0.4

False sense of precision
False sense of completeness
Course selection - which course to take?

<table>
<thead>
<tr>
<th>Subject?</th>
<th>Instructor?</th>
<th># Credits?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Gopal</td>
<td>4</td>
</tr>
<tr>
<td>SE</td>
<td>Tom</td>
<td>3</td>
</tr>
<tr>
<td>NW</td>
<td>Bob</td>
<td>3</td>
</tr>
</tbody>
</table>

- 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 *relative* 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

- 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)

Instructor

Credits

Area

AI > _area_ SE

AI: Gopal > _instr_ Tom
SE: Tom > _instr_ Gopal

E

E = Functional > Unproven

A

LO > HI

Representing and Reasoning with Qualitative Preferences - Ganesh Ram Santhanam, Iowa State University.
CP-Theories

- Similar to TCP-nets but...

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

Instructor: Gopal >_{\text{instr}} Tom
SE: Tom >_{\text{instr}} Gopal

Area: AI >_{\text{area}} SE

Functional > Unproven
LO > HI

E=Functional: Unavailable > Official fix
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 $S_2$
Conditional Importance Networks (CI-nets) [Bouveret 2009]

CI-nets (fair division of goods among agents)

- 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\}

  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

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

Preferences over (Sets of) Binary Variables

- $C^3I$-nets
  - CI-nets
Preference Reasoning

Focus of this tutorial:
- 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
  - 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
  - Nic 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 \( (C^3I\text{-nets}) \) [Santhanam et al. 2013]

In this tutorial ...

- We stick to CP-nets, TCP-nets and CI-nets
- Approach extensible to all other ceteris paribus preference languages
Part II – Theoretical Aspects

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 (IPG) [Boutilier et al. 2001]

- **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*)

Santhanam et al. AAAI 2010
Preference Semantics in terms of IPG

• \((\alpha, \beta) \in E\) iff there is a \textit{flip} from \(\alpha\) to \(\beta\) “\textit{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 \(x_1 \succ_1 x'_1\) (\(x_1\) 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
(α , β) ∈ E iff there is a statement in TCP-net such that $x_1 >_1 x'_1$ ($x_1$ is preferred to $x'_1$) and ...

- **V-flip**: all other variables being equal, $α(X_1)=x_1$ and $β(X_1)=x'_1$

- **I-flip**: all variables *except those* less important than $X_1$ being equal, $α(X_1)=x_1$ and $β(X_1)=x'_1$

**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 (convention)
  – $S^+, S^- : S_1 > S_2$ 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]

- 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{enumerate}
  \item P1. $\{d\}, \{\}\ : \{b\} \succ \{c\}$
  \item P2. $\{b\}, \{a\} : \{c\} \succ \{d\}$
  \item P3. $\{\}, \{d\} : \{a, b\} \succ \{c\}$
  \end{enumerate}
• $C^3I$-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)
  – $S^+, S^- : S_1 > S_2$ 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, $|\alpha| < |\beta|$
  - **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 $S^-$. 
  - Extra cardinality constraint to enable dominance
Flips for a C³I-net [Santhanam et al. 2013]

- 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 CI-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. $\emptyset, \{d\}$ : $\{a, b\} \succ \{c\}$

**M-flip**

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

- $\{c\} \succ \{bc\}$ due to Monotonicity
- $\{bc\} \succ \{bd\}$ due to P2
- $\{ab\} \not\succ \{c\}$ due to Cardinality despite P3

Santhanam et al. CSIIRW 2013
Reasoning Tasks

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

Reasoning tasks reduce to verifying 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 &gt; \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>
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
    • 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
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]

- **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 verification of temporal logic properties on the model
Overview: Preference Reasoning via Model Checking

Representing and Reasoning with Qualitative Preferences - Ganesh Ram Santhanam, Iowa State University.
A Kripke structure is a 4-tuple $K=\langle S, S_0, T, L \rangle$ 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 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 in some sense

- The State-Space of $K_P$
  - $S = \prod_i D_i$ : 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

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

- 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) 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 \( \varrho \) is the “condition” part, guard condition is

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

\[
\text{Allow}(p) := \varrho \land z_i = v_i' \land h_i = 1
\]

\[
\text{Restrict}(p) := \land_{x_j \in X \setminus \{x_i\}} h_j = 0
\]
Encoding CP-net semantics

Direct & succinct

E = Functional: Unavailable > Official fix

Kripke Structure

Induced Preference Graph
Encoding CP-net semantics

Functional, LO, Unavailable

Functional, LO, Official fix

Unproven, LO, Official fix

Unproven, HI, Official fix

Functional, HI, Unavailable

Functional, HI, Official fix

Unproven, HI, Official fix

Unproven, HI, Unavailable

Unproven, LO, Unavailable

E = Functional: Official fix > Unproven

A = LO > HI
Encoding TCP-net Semantics

TCP-nets: Same overall idea as CP-nets

- Additional rule for encoding simple relative importance

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

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

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

[Diagram showing relationships between different states and preferences]
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 $\psi$ if there exists a path $s = s_1 \rightarrow s_2 \ldots$ such that $s_2$ satisfies $\psi$

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

  $E [\psi_1 U \psi_2]$ if there exists a path $s = s_1 \rightarrow s_2 \ldots$ such that $\exists i \geq 1 : s_i$ satisfies $\psi_2$, and $\forall j < i : s_j$ satisfies $\psi_1$

• 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
Given outcomes $\alpha$ and $\beta$, how to check if $\alpha > \beta$?

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

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

*a state in which $\phi_\beta$ holds is reachable from a state in which $\phi_\alpha$ holds*

- $\alpha > \beta$ iff the model checker NuSMV can verify $\phi_\alpha \rightarrow EF\phi_\beta$ (SAT)
- When queried with $\neg(\phi_\alpha \rightarrow EF\phi_\beta)$, 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
Obtaining a Proof of Dominance

- **011 is preferred to 100**
  
  *Improving* flipping sequence:
  
  \[100 \rightarrow 101 \rightarrow 001 \rightarrow 011\]
  
  Proof: \(011 > 001 > 101 > 100\)

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

One of the proofs is chosen non-deterministically

---

**Santhanam et al. AAAI 2010**
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

Santhanam et al. AAAI 2010
Non-dominance

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

Representing and Reasoning with Qualitative Preferences - Ganesh Ram Santhanam, Iowa State University.

Answer

Santhanam et al. ADT 2013
Equivalence and Subsumption Testing

Combined Induced Preference Graph

\[ \overline{ab} \rightarrow ab \rightarrow \overline{ab} \]

Kripke Structure

\[ \overline{ab} \rightarrow ab \rightarrow \overline{ab} \]

State from which verification is done

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

Santhanam et al. ADT 2013

True \iff P_1 \subseteq P_2

Model Checker returns \( \overline{ab} \rightarrow \overline{ab} \) as proof

False \iff P_2 \not\subseteq P_1

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

**Combined Induced Preference Graph**

**Kripke Structure**

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

True ⇔ \( P_1 \sqsubseteq P_2 \)

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

True ⇔ \( P_2 \sqsubseteq P_1 \)

\( P_1 \equiv P_2 \)

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

We represent and reason with qualitative preferences.

Ganesh Ram Santhanam, Iowa State University.
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
Applications

- Sustainable Design
Applications

- Sustainable Design

<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 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
Applications

- Goal Oriented Requirements Engineering

Oster et al. ASE 2011
Goal oriented Requirements Engineering – CI-nets

CI-net statements
{d} ; {} : {c} ; {b}
{b} ; {a} : {d} ; {c}
{} ; {d} : {c} ; {a,b}

Oster et al. ASE 2011
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,

\[ \text{when it could have been avoided} \]

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

\[ P1. \{ d \}, \{ \} : \{ b \} \succ \{ c \} \]
\[ P2. \{ b \}, \{ a \} : \{ c \} \succ \{ d \} \]
\[ P3. \{ \}, \{ d \} : \{ a, b \} \succ \{ c \} \]
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
  - Modular design enables extension to other ceteris paribus languages, reasoning tasks and encodings

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

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

Query Preprocessor

Language 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.

Result
iPref-R 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

Working implementation to demonstrate our approach
(soon on our website)
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. iPref-R

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

*Representing and Reasoning with Qualitative Preferences - Ganesh Ram Santhanam, Iowa State University.*
References

References

- Walid Trabelsi, Nic Wilson, Derek G. Bridge: *Comparative Preferences Induction Methods for Conversational Recommenders*. ADT 2013
- Francesca Rossi, Kristen Brent Venable, Toby Walsh: *A Short Introduction to Preferences: Between Artificial Intelligence and Social Choice*. Synthesis Lectures on Artificial Intelligence and Machine Learning 2011.
- Francesca Rossi, Kristen Brent Venable, Toby Walsh: *Preferences in Constraint Satisfaction and Optimization*. AI Magazine 2008.
References

- Cristina Cornelio, Judy Goldsmith, Nicholas Mattei, Francesca Rossi, Kristen Brent Venable: *Updates and Uncertainty in CP-Nets*. Australasian Conference on Artificial Intelligence 2013: 301-312
- Umberto Grandi, Andrea Loreggia, Francesca Rossi, Vijay A. Saraswat: *From Sentiment Analysis to Preference Aggregation*. ISAIM 2014
- Minyi Li, Quoc Bao Vo, Ryszard Kowalczyk: *Efficient heuristic approach to dominance testing in CP-nets*. AAMAS 2011
Thank you

Team:
Ganesh Ram Santhanam, Samik Basu, Vasant Honavar

Other Collaborators
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
Efficient Dominance Testing for Unconditional Preferences*

- Approach – restrict the preference language
  - TUP-net = Unconditional fragment of TCP-net
    - Less expressive, yet practically useful

<table>
<thead>
<tr>
<th>Type of Preference</th>
<th>CP-net</th>
<th>TUP-net</th>
<th>TCP-net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unconditional Intra-variable preference</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Unconditional Relative importance</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Conditional Intra-variable preference</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Conditional relative importance</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

- Introduce a new dominance relation (QBF) for TUP-nets
- Dominance testing in TUP-nets = QBF SAT in polynomial time
- Compare the new dominance relation with unconditional counterparts of TCP-nets, CP-nets

*Santhanam, Basu, Honavar. KR 2010
Dominance for Unconditional Preferences

- Approach – restrict the preference language
  - TUP-net = Unconditional fragment of TCP-net
    - Less expressive, yet practically useful
  - Introduce a new dominance relation (QBF) for TUP-nets
  - Dominance testing in TUP-nets = QBF SAT in polynomial time

Definition (Dominance for Unconditional Preferences)

Given \( \succ_i \) and \( \succ \), and outcomes \( \alpha \) and \( \beta \), \( \alpha \) dominates \( \beta \) iff:

\[
\exists X_i : \alpha(X_i) \succ_i \beta(X_i) \\
\land \forall X_k : (X_k \succ X_i \lor X_k \sim X_i) \Rightarrow \alpha(X_k) \succeq_k \beta(X_k)
\]

where \( X_k \sim X_i \iff X_k \not\succeq X_i \land X_i \not\succeq X_k \), and \( X_i \) is called the witness of the relation.
Properties of Dominance

- Desirable – Strict Partial Order
  - Irreflexivity – Yes
  - Transitivity – No!

- How to make dominance transitive?
  - Restrict $\triangleright$ to an interval order
  - Interval order – a partial order not containing $2 \oplus 2$ substructure

**Theorem**

If $\succ_i$ are strict partial orders and $\triangleright$ is an interval order, $\triangleright_d$ is a strict partial order.
Comparison of Semantics

\( \triangleright \) is NOT an Interval Order

\( \triangleright \) Wilson

\( \triangleright \) Santhanam et al.

\( \triangleright \) is an Interval Order

\( \triangleright \) Brafman et al.
• Questions
• Tool demo