Preference Trees: A Language for Representing and Reasoning about Qualitative Preferences

Xudong Liu and Miroslaw Truszczynski

Department of Computer Science
University of Kentucky
Lexington, KY, USA

Overview

Example

Let us consider preferences over vacations, combinations of values from domains of the following issues:

1. *activity*: hiking (x_1) or water sports ($\neg x_1$),
2. *destination*: Florida (x_2) or Colorado ($\neg x_2$),
3. *time*: summer (x_3) or winter ($\neg x_3$), and
4. *transportation*: car (x_4) or plane ($\neg x_4$).

Example

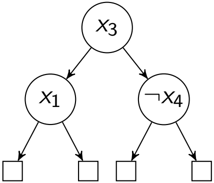


Figure : Vacations

$$x_1x_2x_3x_4, \ldots \succ \neg x_1x_2x_3x_4, \ldots \succ x_1x_2\neg x_3\neg x_4, \ldots \succ x_1x_2\neg x_3x_4, \ldots$$

Contributions

1. We introduce a novel qualitative preference representation language, *preference trees*, or *P-trees*.
2. We show that the language provides an intuitive way to specify preferences over combinatorial domains and how it relates to existing preference formalisms such as *LP-trees*, *ASO* and *possibilistic logic*.
3. We study decision problems in the setting of P-trees and obtain computational complexity results.

- Let $\mathcal{I}$ be a set of binary issues. The *combinatorial domain* determined by $\mathcal{I}$, $CD(\mathcal{I})$, is the set of outcomes represented by complete and consistent sets of literals over $\mathcal{I}$.
- A *preference tree* (*P-tree*) over $CD(\mathcal{I})$ is a binary tree whose nodes other than the leaves are labeled with propositional formulas over $\mathcal{I}$.
- Let T be a P-tree. Given an outcome $M \in CD(\mathcal{I})$, we define the leaf $l_T(M)$ of M in T as the leaf reached by traversing the tree T. When at a node N labeled with φ, if $M \models \varphi$, we descend to the left child of N; otherwise, to the right.

- For $M, M' \in CD(\mathcal{I})$, we set $M' \succeq_T M$ if $l_T(M') \succeq_T l_T(M)$, and $M' \succ_T M$ if $l_T(M') \succ_T l_T(M)$.
- We say that M is *equivalent* to M', $M \approx_T M'$, if $l_T(M) = l_T(M')$.
- Outcome M is *optimal* if there exists no M' such that $M' \succ_T M$.

Preference Trees: Example

$$\varphi_1 = x_1 \vee x_3,\ \varphi_2 = x_1 \wedge \neg x_2,$$
$$\varphi_3 = (x_3 \rightarrow x_4) \vee (\neg x_3 \rightarrow \neg x_4),$$
$$\varphi_4 = x_2 \wedge x_4.$$

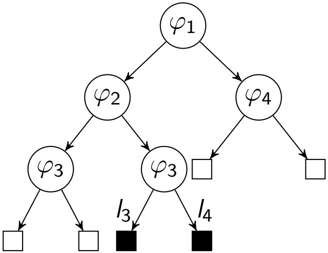


Figure : P-trees

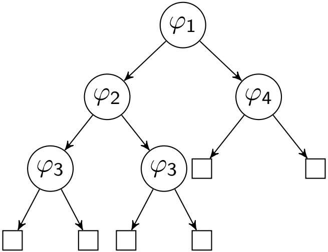
For outcomes $M = (x_1, x_2, \neg x_3, \neg x_4)$, $M' = (\neg x_1, x_2, x_3, x_4)$, and $M'' = (\neg x_1, \neg x_2, x_3, \neg x_4)$, we have $l_T(M) = l_T(M') = l_3$ and $l_T(M'') = l_4$, so $M \approx_T M' \succ_T M''$.

Compact Representation of P-trees

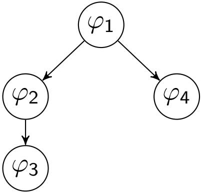
A *compact P-tree over* $CD(\mathcal{I})$ is a tree where

1. every node is labeled with a Boolean formula over $\mathcal{I}$, and
2. every non-leaf node t labeled with φ has either two outgoing edges with the left one representing that φ is satisfied and the right representing that φ is falsified, or one outgoing edge e such that e points
   - straight-down indicating that the two subtrees of t are *identical* and the formulas labeling every pair of corresponding nodes in the two subtrees are the *same*, or
   - left (right) indicating empty right (left, resp.) subtree of t.

$$\varphi_1 = x_1 \vee x_3,\ \varphi_2 = x_1 \wedge \neg x_2,$$
$$\varphi_3 = (x_3 \rightarrow x_4) \vee (\neg x_3 \rightarrow \neg x_4),$$
$$\varphi_4 = x_2 \wedge x_4.$$



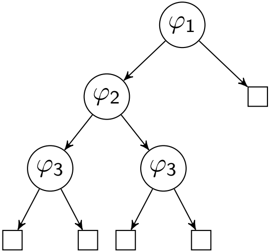
(a) Full



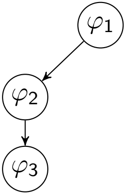
(b) Compact

Figure : P-trees

$$\varphi_1 = x_1 \vee x_3,\ \varphi_2 = x_1 \wedge \neg x_2,$$
$$\varphi_3 = (x_3 \rightarrow x_4) \vee (\neg x_3 \rightarrow \neg x_4).$$

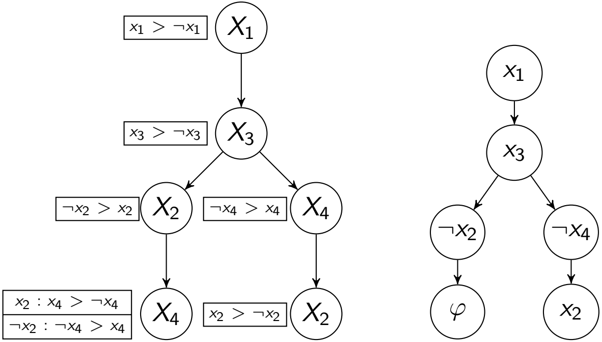


(a) Full



(b) Compact

Figure : P-trees



(a) An LP-tree L

(b) The P-tree T_L[2]

Figure : P-trees extend LP-trees

[2] $\varphi = (x_2 \to x_4) \vee (\neg x_2 \to \neg x_4)$

[3] Booth, Chevaleyre, Lang, Mengin, and Sombattheera. *Learning conditionally lexicographic preference relations*, 2010.

- An ASO-rule[4] r over $\mathcal{I}$ is a preference rule of the form

$$C_1 > \ldots > C_m \leftarrow B,$$

  where all C_i's and B are propositional formulas over $\mathcal{I}$.

- For an outcome $M \in CD(\mathcal{I})$, the rule determines its *satisfaction degree*[5], denoted by $SD_r(M)$ such that

$$SD_r(M) = \begin{cases} 1, & M \models \neg B \\ m+1, & M \models B \wedge \bigwedge_{1 \le i \le m} \neg C_i \\ min\{i : M \models C_i\}, & \text{otherwise.} \end{cases}$$

- $M \succeq_r M'$ if $SD_r(M) \le SD_r(M')$.

[4] Brewka, Niemela, and Truszczynski. *Answer set optimization*, 2003.

[5] This definition is a slight adaptation of the original one.

From ASO-Rules to P-Trees

From the ASO-rule r, we build a P-tree T_r, where $\varphi_1 = \neg B \vee C_1$, $\varphi_i = C_i$ $(2 \leq i \leq m)$.

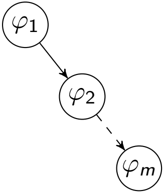


Figure : A P-tree T_r

Theorem

Given an ASO-rule r, there exists a P-tree T_r of size polynomial in the size of r such that for every two outcomes M and M'

$$M \succeq_r^{ASO} M' \text{ iff } M \succeq_{T_r} M'$$

From P-Trees to ASO-Rules?

Figure : A P-tree T

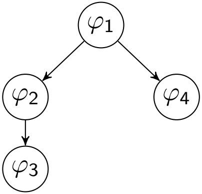
$$r_T : \underbrace{\varphi_1 \wedge \ldots \wedge \varphi_m \succ \varphi_1 \wedge \ldots \wedge \neg\varphi_m \succ \ldots \succ \neg\varphi_1 \wedge \ldots \wedge \neg\varphi_m}_{2^m \text{ formulas!}}$$

Ranked Answer Set Optimization (RASO)

- An (unranked) ASO-theory, or ASO-theory, P is a set of ASO-rules aggregated by the Pareto method.
  1. $M \succeq_P^{un} M'$ if $SD_r(M) \leq SD_r(M')$ for every $r \in P$.
  2. $M \succ_P^{un} M'$ if $M \succeq_P^{un} M'$ and $SD_r(M) < SD_r(M')$ for some $r \in P$.
  3. $M \approx_P^{un} M'$ if $SD_r(M) = SD_r(M')$ for every $r \in P$.
- An ranked ASO-theory[6], or RASO-theory, $P = \{P_1, \ldots, P_g\}$ is a set of ASO-rules with g ranks, the smaller the rank the more important the rules.
  1. $M \succeq_P^{rk} M'$ if for every i, $1 \leq i \leq g$, $M \approx_{P_i}^{un} M'$, or if there exists a rank i such that $M \approx_{P_j}^{un} M'$ for every j, $j < i$, and $M \succ_{P_i}^{un} M'$.

[6] Brewka, Niemela, and Truszczynski. *Answer set optimization*, 2003.

From P-trees to RASO-Theories



(a) P-tree T

$$
\begin{gathered}
x_1 \vee x_3 \xleftarrow{1} \\
x_1 \wedge \neg x_2 \xleftarrow{2} x_1 \vee x_3 \\
x_2 \wedge x_4 \xleftarrow{2} \neg(x_1 \vee x_3) \\
(x_3 \rightarrow x_4) \vee (\neg x_3 \rightarrow \neg x_4) \xleftarrow{3} x_1 \vee x_3
\end{gathered}
$$

(b) RASO-theory Φ_T

Figure : P-trees

Theorem

Given a P-tree T, there exists an RASO-theory Φ_T of size polynomial in the size of T such that for every two outcomes M and M'

$$M \succeq^{RASO}_{\Phi_T} M' \ \text{ iff } \ M \succeq_T M'$$

P-Trees and Answer Set Optimization

ASO-rules $\subset$ P-trees $\subset$ RASO-theories

Dominance-testing (DomTest)

Given a P-tree T and its two distinct outcomes M and M', decide whether $M' \succeq_T M$.

Optimality-testing (OptTest)

Given a P-tree T and an outcome M, decide whether M is an optimal outcome with respect to T.

Optimality-with-property (OptProp)

Given a P-tree T and some property α expressed as a Boolean formula, decide whether there is an optimal outcome that satisfies α.

Computational Complexity Results

DomTest	OptTest	OptProp
P	coNP-complete[7]	$\Delta_2^P(P^{NP})$-complete[8]

Figure : Computational complexity results

[7] The complement problem is reduced from the SAT problem.

[8] The problem is reduced from the Maximum Satisfying Assignment (MSA) problem.

Future Work

1. Relating P-trees with CP-nets[9].
2. Learning P-trees: elicitation, learning.
3. Aggregating P-trees: the Pareto method, social choice rules.

Questions? Thank you!

[9] Boutilier, Brafman, Domshlak, Hoos, and Poole. *CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements*, 2004.