On Sufficient Conditions for Consistency Checking in CP-theory Preferences

Erik Rauer, **Samik Basu**
Department of Computer Science, Iowa State University
{etrauer, sbasu}@iastate.edu,

Abstract

Checking consistency of qualitative preferences in CP-theory is PSPACE-complete in general. Building on Wilson's seminal work on Complete Search (cs) tree based sufficient conditions for preferences, we characterize the necessary and sufficient conditions for the existence of a cs-tree, yielding the weakest sufficient condition for cs-tree–based consistency, and demonstrate that testing consistency under this condition is coNP-complete. Additionally, we present a polynomial-time computable upper approximation of the dominance relation for cs-tree–consistent CP-theory preferences, and prove that it subsumes all previously proposed approximations. Finally, we introduce set-labeled cs-trees, a generalization of cs-trees, which characterizes necessary and sufficient conditions for consistency checking in CP-theory preferences.

1 Introduction

CP-theory languages (Boutilier et al. 2004; Brafman, Domshlak, and Shimony 2006; Wilson 2004; Santhanam, Basu, and Honavar 2016) offer an expressive framework for specifying qualitative preferences over attributes or properties of choices/outcomes using conditional dependencies and relative importance. A set of CP-theory preferences induces a dominance relation among choices, which gives rise to an induced preference graph over the set of choices.

One of the challenges in preference reasoning is verifying whether a given set of preferences is consistent, that is, whether the preferences do not induce dominance of an outcome over itself. Equivalently, the induced preference graph is acyclic. This issue is particularly critical in preference aggregation, where the preferences of multiple agents are jointly considered to determine a collective optimal choice while ensuring that the aggregation remains consistent. The problem is PSPACE-complete (Goldsmith et al. 2005), in general, for conditional preferences.

In his seminal work, Wilson (Wilson 2011) introduced the notion of a complete search tree, cs-tree, to characterize consistency in conditional preferences. A cs-tree is defined over the attributes in the preferences and induces a total order over the corresponding outcomes. A cs-tree whose induced order extends the partial order induced by the preferences is called a satisfying cs-tree. The existence of such a tree guarantees consistency of the preferences and the corresponding preferences are termed conditionally acyclic. Wilson further developed several cs-tree–based sufficient conditions for consistency, including strong conditional acyclicity and context-uniform conditional acyclicity. Each condition ensures the existence of a satisfying cs-tree and is coNP-complete to test. More recently, (Rauer, Basu, and Honavar 2025) leveraged the concept of a cs-tree to propose a polynomial-time testable sufficient condition for consistency that is incomparable with Wilson's conditions.

A natural question that remains is: *what is the necessary and sufficient condition for conditional acyclicity*? This problem has remained open for more than a decade. In this paper, we resolve this question. Our main contributions are:

1. We establish a necessary and sufficient condition for conditional acyclicity and prove that deciding this condition is coNP-complete.
2. We propose a polynomial-time upper-approximation dominance relation for conditionally acyclic preferences, and show that any dominance induced by the original preferences implies this upper-approximation relation. Moreover, this is the strongest poly-time upper approximation for conditional acyclic preferences among those currently available in the literature.
3. We introduce the notion of a generalized cs-tree, namely a set-labeled cs-tree, which provides a parameterized framework for characterizing progressively weaker sufficient conditions for consistency checking in CP-theory preferences, ultimately converging to the necessary and sufficient condition for consistency.

2 Preliminaries

Notational Convenience. We will use $\mathcal{A}, \mathcal{B}, \mathcal{U}, \mathcal{V}$ to denote sets of variables; upper-case letters X, Y and their primed versions to denote variables in the set. The domain of a variable V is denoted by $\underline{V}$. We overload the notation for $\mathcal{V}$, where $\underline{\mathcal{V}}$ denotes $\prod_{X \in \mathcal{V}} \underline{X}$. We use lower-case letters (such as x, y) to denote some valuation in a domain.

Basic conditions over variable assignments are of the form $X = x$. General conditions are basic conditions and their conjunctions. A conjunction of assignments $\bigwedge_i X_i = x_i$ captures a set of assigned values $\bigcup_i \{x_i\}$. Such assigned values are denoted by ρ. We use $\rho(X)$ to denote the valuation of a specific variable X in the assignment ρ and use var(ρ) to denote the the set of variables assigned in ρ. Observe that, the assignment for a set of variables $\mathcal{V}$ belongs to $\underline{\mathcal{V}}$. $\perp$ will be used to denote the empty assignment.

An assignment ρ is an *extension* of another assignment σ, denoted by $\rho \Vdash \sigma$, when every assignment in σ is also present in ρ. We use the phrases interchangeably–ρ extends σ and σ satisfies ρ. The relation $\Vdash$ is reflexive and transitive.

An assignment ρ is *compatible* with σ, denoted by $\rho \bowtie \sigma$, when $\forall i.\ a_i \in \rho \ \wedge\ b_i \in \sigma \ \Rightarrow\ a_i = b_i$. In short, the assignments in ρ and σ do not contradict each other, though all assignments in ρ may not be present in σ and vice versa. The relation $\bowtie$ is symmetric and reflexive. Additionally, $\sigma \Vdash \rho \Rightarrow \sigma \bowtie \rho$, but the converse might not hold.

2.1 CP-theory Preferences

Qualitative preferences expressed in the CP-theory language (Wilson 2004) are described over a set of attributes or variables and capture the conditional desirability of attribute values in the presence of relative importance relationship among attributes. The general structure of such a preference p is as follows:

$$\rho :\ X = x > X = x' \quad \Omega$$

The statement captures the dominance of outcome or choice o over o' when

- (*intra-attribute preference*) the valuations of X in o and o' are x and x', respectively;
- (*conditional dependency*) the valuations of variables in o and o' satisfies ρ;
- (*all-else-equal except relatively less important variables*) the valuations of variables that are not in $\{X\} \cup \Omega$ are identical in o and o'

The last condition, implicitly, identifies the variables in the set Ω as relatively less important than X. For any preference statement p of the above form, we will denote the condition as ρ_p, the variable for which the preference is expressed as X_p, the relatively less important variable-set as Ω_p and the relation as $>_p$.

The dominance relation $\geq_P$ induced by a set P of preferences is the transitive closure of $>_p$ for all $p \in P$ and results in an induced preference graph (IPG) over choices, where the edges in the graph indicate the dominance relation. A set of preferences is *consistent* when there exists no cycles in the corresponding induced preference graph.

Example 1. *The following are valid CP-theory preferences over the binary variables A, B, C:*

$$p_1. \quad a_1 :\ b_2 > b_1\ [C] \qquad p_2. \quad a_2 :\ c_2 > c_1\ [B]$$

$$p_3. \quad b_2 :\ a_1 > a_2\ [\,]$$

The corresponding induced preference graph an be seen in Figure 1. Here, preference p_3 says that, if two choices both satisfy $B = b_2$ and only differ on the value of A, then the choice with $A = a_1$ is more preferred to (or dominates) that with $A = a_2$. For example, the outcomes $o = a_1b_2c_1$ and $o'a_2b_2c_1$ only differ in the value of variable A and both satisfy $B = b_2$, so o is more preferred to o' according to p_3. Preference p_1 on the other hand, allows comparisons between outcomes that also differ on the value of C, not just B, so long as both outcomes satisfy $A = a_1$.

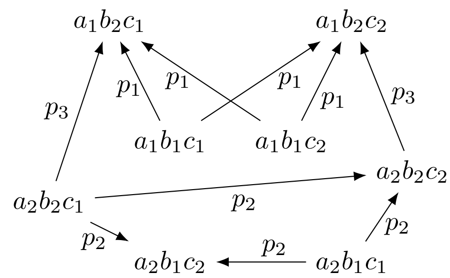


Figure 1: Induced Preference Graph over dominance relation induced by preferences in Example 1, edges are from less to more preferred outcomes. Edge-labels denote the statements which induce them.

We provide a concise overview of the concept and application of the cs-tree, introduced by Wilson (Wilson 2006), which is used to verify the consistency of preferences.

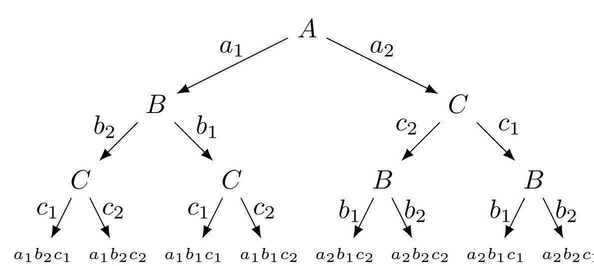
2.2 cs-tree for Consistency Checking

A *cs-tree* τ over a set of variables $\mathcal{V}$ is a rooted directed tree where each internal (i.e. non-leaf) node r is associated with a tuple $(\mathcal{A}_r, a_r, X_r, >_r)$, where

- $\mathcal{A}_r \subset \mathcal{V}$ is a set of variables ($\mathcal{A}_r = \emptyset$ for the root node) and a_r is an assignment to said variables
- $X_r \in \mathcal{V}$ is the variable associated with r and $>_r$ is a total order on the domain of X_r

Each internal node r has a child associated with every value in the domain of X_r, which are ordered according to the relation $>_r$. For child r' associated with value $x \in \underline{X_r}$, $a_{r'} = a_r \cup \{X_r = x\}$. The leaf nodes have an assignment to all variables of $\mathcal{V}$ and are thus associated with a unique outcome. The ordering of leaf-nodes, denoted by the relation $>_\tau$ for cs-tree τ, follows from those of their parents. Every cs-tree defines an ordering on the outcomes and we say that this ordering *satisfies* a CP-theory preference statement-set P, if $\geq_P \subseteq >_\tau$. That is, the total ordering presented in the cs-tree is consistent with the partial order of the induced preference graph corresponding to the preferences in P.

Example 2. *Consider again the preference statements given in Example 1. The following is an example of a cs-tree satisfying these preferences. Tree edges are ordered such that the valuation associated with the left-edge is preferred to that of the right-edge, e.g. $a_1 >_r a_2$ in the root node.*



Wilson developed a necessary and sufficient condition for verifying whether a given cs-tree satisfies a given set of preferences.

Proposition 1 ((Wilson 2011))**.** *Given a set P of CP-theory preferences over non-singular domain attributes and a cs-tree τ, the following conditions holds if and only if τ satisfies P.*

1. *For any $p \in P$ and any outcome o such that $o \Vdash \rho_p$, on the path from the root to o, X_p appears before each element of Ω_p.*
2. *for any body node r in τ and any $p \in P$ such that $X_p = X_r$ and $\rho_p \bowtie a_r$, we have $x_p >_r x'_p$.*

At a high level, the conditions ensure that, for any outcome, the cs-tree contains a path to that outcome in which more important attributes precede less important ones, and that the children of each node are ordered according to the applicable preference statements consistent with the assignments made from the root to that node.

Proceeding further, Wilson establishes a sufficient condition for consistency based on the concept of a cs-tree.

Theorem 1 (Conditional Acyclicity (Wilson 2011))**.** *A set P of CP-theory preferences is consistent if there exists a cs-tree that satisfies P.*

Wilson refers to the preferences that exhibit the above property as **conditionally acyclic**. He introduces two conditions over a given set of preferences which, when satisfied, imply conditional acyclicity. These conditions are known as strong conditional acyclicity and context-uniform conditional acyclicity. *However, identifying necessary and sufficient conditions for conditional acyclicity remains an open problem. In this paper, we address this question.*

3 Characterization of Conditional Acyclicity

3.1 Background Definitions & Properties

We proceed with the definitions that establishes the property for *local consistency* of attributes.

Definition 1 (Variable-value Ordering)**.** *Let P be a set of CP-theory preferences on variables $\mathcal{V}$, $X \in \mathcal{V}$ be a variable, $\mathcal{A} \subseteq \mathcal{V} \setminus \{X\}$ a set of variables, and $a \in \underline{\mathcal{A}}$ a corresponding assignment.*

We define $>^X_a \subseteq \underline{X} \times \underline{X}$ as the transitive closure of all pairs (x, x') where $x, x' \in \underline{X}$ and there exists $p \in P$ with $\rho_p \bowtie a$, $X_p = X$, and $x >_p x'$.

The definition establishes an order over valuations of a variable based on the set of preferences whose conditions are compatible with the given assignment to some of the variables.

Definition 2 (Local Consistency)**.** *Let P be a set of CP-theory preferences on variables $\mathcal{V}$. Consider variable $X \in \mathcal{V}$ and a set of variables $\mathcal{A} \subseteq \mathcal{V} \setminus \{X\}$.*

- *We say that X is* locally consistent under assignment $a \in \underline{\mathcal{A}}$ *if $>^X_a$ is irreflexive.*
- *If for all $\mathcal{A} \subseteq \mathcal{V} \setminus \{X\}$ and assignments $a \in \underline{\mathcal{A}}$, X is locally consistent under a, we say that X is* locally consistent.
- *We say that P is* locally consistent *if all variables in $\mathcal{V}$ are locally consistent.*

Lemma 1 (Checking Local Consistency)**.** *Given a set of CP-theory preferences P, variable $X \in \mathcal{V}$, $\mathcal{A} \subseteq \mathcal{V} \setminus \{X\}$, $a \in \underline{\mathcal{A}}$, the local consistency of X under a can verified in $O(|P||\mathcal{V}| + |\underline{X}|)$ time.*

Proof. Given $a \in \underline{\mathcal{A}}$, checking local consistency of X under a can be done by constructing a directed graph on $\underline{X}$ and checking whether it is acyclic or not. To construct the graph, for every preference statement $p \in P$, check whether $X_p = X$ and $\rho_p \bowtie a$. If both are true, add the edge (x_p, x'_p). Determining compatibility ($\rho_p \bowtie a$) can be done in time $O(\mathcal{V})$. Hence, the construction of the graph takes $O(|P||\mathcal{V}|)$ time. Clearly (x, x') is in this graph if and only if $x >^X_a x'$, so $>^X_a$ is irreflexive if and only if there is no cycle in this graph. Since there are $O(|\underline{X}|)$ vertices and $O(|P|)$ edges in the constructed graph, checking acyclicity can be done in $O(|\underline{X}| + |P|)$ time via a DFS. This gives a total of $O(|P||\mathcal{V}| + |\underline{X}|)$ time. ☐

Recall that, for any preference statement p over a variable X_p, Ω_p denotes the set of variables that are less important than X_p. We instead define on the complementary perspective, describing the variables that are more important than a given variable X, as this perspective will be more convenient for our subsequent analysis.

Definition 3 (Relatively More Important Variables)**.** *Given set of CP-theory preferences P, variables $X, Y \in \mathcal{V}$, $\mathcal{A} \subseteq \mathcal{V}$, $a \in \underline{\mathcal{A}}$, we say that Y is* relatively more important than X under a *if there exists $p \in P$ such that $a \bowtie \rho_p$, $Y = X_p$, and $X \in \Omega_p$.*

We are now ready to define the concept of a-rooted condition, which establishes an ordering among the variables for a given variable assignment a and serves as the basis for our necessary and sufficient condition for conditional acyclicity.

Definition 4 (a-rooted)**.** *Given a set of CP-theory preferences P on $\mathcal{V}$, a set of variables $\mathcal{A} \subset \mathcal{V}$, and an assignment $a \in \underline{\mathcal{A}}$, we say that a variable $X \in \mathcal{V} \setminus \mathcal{A}$ is a-rooted if the following conditions hold*

1. *X is locally consistent under a, and*
2. *$\forall Y \in \mathcal{V}$, Y is relatively more important than X under a implies $Y \in \mathcal{A}$.*

While the conditions for a variable to be a-rooted are similar to those of a (strongly) a-undominated variable (Wilson 2006; Wilson 2011), there is a key difference. Namely, an a-undominated variable requires all variables it conditionally depends on to be assigned in a, on the other hand, an a-rooted variable only requires enough of these assignments to make the variable locally consistent.

3.2 Necessary & Sufficient Condition

We proceed by establishing the relationship between a-rooted condition (Definition 4) and a cs-tree that satisfies a given set of preferences (Proposition 1). The objective is to identify the correspondence between the ordering of variables induced by a-rooted condition and the hierarchical arrangement of nodes (corresponding to these variables) in a cs-tree that satisfies the given preferences. We will then prove that conditional acyclicity holds when for valuation a of any subset of variables, there exists a variable that is a-rooted, which, in turn, will ensure the existence of a satisfying cs-tree. This will lead to the necessary and sufficient condition for conditional acyclicity.

Lemma 2. *If a cs-tree τ satisfies CP-theory preferences P, then for every body node $r \in \tau$ where r is associated with the tuple $(\mathcal{A}_r, a_r, X_r, >_r)$, X_r is a_r-rooted.*

Proof. By contrapositive, suppose that there is some body node $r \in \tau$ such that X_r is not a_r-rooted. Then either condition 1 or condition 2 in Definition 4 does not hold.

In case of the former, there must exist preference statements such that some pair $x, x' \in \underline{X_r}$ with $x \neq x'$ has both $x >_{a_r}^{X_r} x'$ and $x' >_{a_r}^{X_r} x$. However, the relation $>_r \subseteq \underline{X} \times \underline{X}$ at the node r of the cs-tree as per condition 2 in the Proposition 1, is a total order. Hence, it cannot be the case that $x >_r x'$ and $x' >_r x$. As a result, the condition 1 in Definition 4 does not hold true for X_r w.r.t. a_r.

Proceeding further, condition 2 in Definition 4 must not hold. That is, there exists some $Z \in \mathcal{V}$ such that Z is relatively more important than X_r under a_r, but $Z \notin \mathcal{A}_r$. Let $p \in P$ be the preference with $a_r \bowtie \rho_p$, $X_p = Z$ and $X_r \in \Omega_p$, and consider outcome $o \Vdash \rho_p$ and $o \Vdash a_r$ (which must exist since $a_r \bowtie \rho_p$). Hence, r is on the path from the root to o. However since $Z \notin \mathcal{A}_r$, Z appears after X_r which, in turn, implies that the condition 1 in Proposition 1 does not hold for o and p.

Therefore, if there exists a node $r \in \tau$ where X_r is not a_r-rooted then τ does not satisfy P. □

Note that the converse does not hold. This is because, for a body node r, X_r being a_r-rooted does not force the corresponding order $>_r$ to be consistent with $>_{a_r}^{X_r}$.

Lemma 3. *Given set of CP-theory preferences P, variable $X \in \mathcal{V}$, $\mathcal{A} \subseteq \mathcal{V} \setminus \{X\}$, $a \in \underline{\mathcal{A}}$, let X be a-rooted. Then for any assignment a' with $a' \Vdash a$ (a' extending a), X is a'-rooted.*

Proof. It is immediate that this holds if $a = a'$.

So, let us consider $a \neq a'$ and let $\mathcal{A}' \subseteq \mathcal{V} \setminus \{X\}$ be such that $a' \in \underline{\mathcal{A}'}$. We define the following sets:

$$P_a = \{p \in P \mid \rho_p \bowtie a \ \wedge \ X_p = X\},$$
$$P_{a'} = \{p \in P \mid \rho_p \bowtie a' \ \wedge \ X_p = X\}$$

We proceed with proving the locally consistent property for X under a' (see Condition 1 for Definition 4).

Consider any preference statement $p' \in P_{a'}$ and identify the set of variables Y such that $Y \in \mathtt{var}(\rho_{p'}) \cap \mathcal{A}'$. If no such Y exists, then $\rho_{p'} \bowtie a$.

If there exists a Y, then $Y \in \mathcal{A}$ or $Y \notin \mathcal{A}$. In case of former, $Y \in \mathtt{var}(\rho_{p'}) \cap \mathcal{A}$ and as $a' \Vdash a$, $a'(Y) = a(Y)$. Furthermore, $\rho_{p'} \bowtie a'$ and $Y \in \mathtt{var}(\rho_{p'}) \cap \mathcal{A}'$ implies $\rho_{p'}(Y) = a'(Y)$ and hence $\rho_{p'}(Y) = a(Y)$.

In case, when $Y \notin \mathcal{A}$, the valuation of Y in $\rho_{p'}$ does not impact a and hence, $\rho_{p'} \bowtie a$ as well. Therefore, $P_{a'} \subseteq P_a$ and $>_{a'}^X \subseteq >_a^X$. Since X is a-rooted, $>_a^X$ is irreflexive and that implies $>_{a'}^X$ is irreflexive as well.

Next, we consider the Condition 2 in Definition 4. Consider a variable Y that is relatively more important than X under a and the preference statement p' determines that relative importance relation. Consider

$$P_a = \{p \in P \mid \rho_p \bowtie a \ \wedge \ X_p = Y\},$$
$$P_{a'} = \{p \in P \mid \rho_p \bowtie a' \ \wedge \ X_p = Y\}$$

As $a' \Vdash a$, $P_{a'} \subseteq P_a$. Therefore, $p' \in P_{a'}$ implies $p' \in P_a$. Since X is a-rooted, $Y \in \Omega_p'$ belongs to $\mathcal{A}$. Using the relation $a' \Vdash a'$, we have $\mathcal{A} \subseteq \mathcal{A}'$. Hence, Y belongs to $\mathcal{A}'$ as well. □

Theorem 2 (Necessary & Sufficient Condition for Conditional Acyclicity)**.** *A set of CP-theory statements P is conditionally acyclic if and only if for all $\mathcal{A} \subset \mathcal{V}$, $\forall a \in \underline{\mathcal{A}}$, $\exists X \in \mathcal{V} \setminus \mathcal{A}$ such that X is a-rooted.*

Proof. $(\Rightarrow)$: Suppose P is conditionally acyclic and let τ be the corresponding cs-tree satisfying P. Note that this means τ satisfies the conditions of Proposition 1. Let $\mathcal{A} \subset \mathcal{V}$ be an arbitrary set of variables and $a \in \underline{\mathcal{A}}$ an arbitrary assignment for $\mathcal{A}$. We will show that there is some $X \in \mathcal{V} \setminus \mathcal{A}$ that is a-rooted.

Since $\mathcal{A} \neq \mathcal{V}$ there must exist some node r in cs-tree τ associated with the tuple $(\mathcal{A}_r, a_r, X_r, >_r)$, such that $a \Vdash a_r$ and $X_r \notin \mathcal{A}$. This r is the first node on the path to any leaf-node corresponding to outcomes extending a where $X_r \notin A$. By Lemma 2 X_r is a_r-rooted. Additionally, since $X_r \notin A$ and $a \Vdash a_r$, X_r is a-rooted by Lemma 3.

$(\Leftarrow)$: Suppose the RHS above holds. We will show that there exists a cs-tree τ satisfying the conditions of Proposition 1 and hence, satisfies P as well. This, in turn, establishes the conditional acyclicity of P.

For each body node r of τ that is associated with $(\mathcal{A}_r, a_r, X_r, >_r)$, let X_r be a_r-rooted. By the premise that the RHS above holds, such an X_r must exist, regardless of what a_r is. By Definition 4 $>_{a_r}^{X_r}$ is irreflexive, so there exists some total order consistent with it. Let $>_r$ be such a total order consistent with $>_{a_r}^{X_r}$. Additionally, by definition of $>_{a_r}^{X_r}$, $>_r$ must satisfy the second condition of Proposition 1.

Next consider an arbitrary preference $p \in P$ with $\Omega_p \neq \emptyset$. For each $Y \in \Omega_p$, there is some node r in cs-tree τ such that $X_r = Y$ and $a_r \bowtie \rho_p$, i.e., the valuations of variables leading to r satisfies the conditions associated with the preference statement p. Note that X_p, the variable whose preference is expressed in p is relatively more important than Y. Furthermore, by our construction of τ, Y is a_r-rooted. Therefore, by Definition 3 and Definition 4 (condition 2), X_p is relatively more important than Y under a_r and thus X_p appears before Y in τ, i.e., $X_p \in \mathcal{A}_r$. Let o be an arbitrary outcome extending both a_r and ρ_p (i.e. $o \Vdash a_r$ and $o \Vdash \rho_p$). Clearly r is on the path from the root of τ to o and since $X_p \in A_r$, X_p appears before Y on said path. This concludes that given that RHS holds, one can construct a cs-tree τ that satisfies the condition 1 of Proposition 1.

The constructed cs-tree τ satisfies the conditions of Proposition 1 and implies that τ satisfies the given set of preferences, which further implies that the preferences are conditionally acyclic. □

Example 3. *In Example 2 we showed that the preferences of Example 1 are conditionally acyclic via the existence of a cs-tree. These preferences also satisfy the condition given in Theorem 2: The variable A is $\perp$-rooted, so it is u-rooted for any assignment u that does not contain A by Lemma 3. B is u-rooted when u is one of the assignments a_1, a_1c_1, or a_1c_2. Finally, C is u-rooted when u is any other assignment. As such, all assignments u have a variable that is u-rooted.*

However, since B is relatively more important than C and C is relatively more important than B, the high-cardinality dependency graph used to determine cc-acyclicity has a cycle (Rauer, Basu, and Honavar 2025), so these preferences are not cc-acyclic. Additionally, since every variable is conditionally dependent on another variable, there is no $\perp$-undominated variable (Wilson 2011). This means that the preferences are not strong conditionally acyclic either. So this example demonstrates that conditional acyclicity (and its equivalent characterization) is less strict than other existing sufficient conditions.

This new equivalent characterization allows for a complexity bound on determining whether a set of preferences is conditionally acyclic.

Corollary 1. *Determining whether a set of CP-theory preferences P is conditionally acyclic is coNP-complete.*

Proof. If P is not conditionally acyclic, then there exists some $\mathcal{A} \subset \mathcal{V}$ and $a \in \underline{\mathcal{A}}$ such that for all $X \in \mathcal{V} \setminus \mathcal{A}$, X is not a-rooted.

Given a as the certificate, verifying that the condition does not hold can be done in polynomial time. By Lemma 1, local consistency (condition 1 in Definition 4) under a can be checked in polynomial time. Similarly, condition 2 in Definition 4 can be checked by determining whether for all $p \in P$ such that $a \bowtie \rho_p$, the variable $X_p \in \mathcal{A}$ and $X \in \Omega_p$. This can be done in $O(PV)$ time. So determining conditional acyclicity is in coNP.

The same reduction from 3-SAT as in the proof of Proposition 24 in (Wilson 2011) can be used to show coNP-hardness. Consider a 3-SAT instance with m clauses on propositional variables $\mathcal{U}$ and denote the ith clause $C^i = l_1^i \vee l_2^i \vee l_3^i$. Let $\mathcal{V} = \mathcal{U} \cup \{Z_0, Z_1, \ldots, Z_m\}$ where each Z_i has binary domain $\{z_i, z_i'\}$. For each clause C^i and each $j = 1, 2, 3$, add the preference statement $l_j^i\colon z_{i-1} > z_{i-1}'[Z_i]$ to P. Also add the preference statement $z_m > z_m'[Z_0]$. We show that the given reduction is conditionally acyclic if and only if the 3-SAT instance is unsatisfiable.

This reduction is clearly locally consistent, so all variables are locally consistent under any assignment to any subset of $\mathcal{V}$. So condition 1 of Definition 4 holds for all possible assignment-variable pairs. Thus by Theorem 2, P is not conditionally acyclic if and only if there is some $\mathcal{A} \subset \mathcal{V}$ and $a \in \underline{\mathcal{A}}$ such that for all $X \in \mathcal{V} \setminus \mathcal{A}$, condition 2 of Definition 4 does not hold.

If there is some satisfying assignment to the 3-SAT instance, let $\mathcal{A} = \mathcal{U}$ and assignment $a \in \underline{\mathcal{A}}$ be said satisfying assignment. Note that $\mathcal{V} \setminus \mathcal{A} = \{Z_0, Z_1, \ldots, Z_m\}$. For arbitrary Z_k (with $k \in \{1, \ldots, m\}$), there must exist a $j \in \{1, 2, 3\}$ such that l_j^k is true. As such, Z_{k-1} is relatively more important than Z_k under a, however $Z_{k-1} \notin \mathcal{A}$, so Z_k does not satisfy condition 2. The same holds for Z_0, just with Z_m instead of Z_{k-1}. So there is no variable in $\mathcal{V} \setminus \mathcal{A}$ that is a-rooted, making P not conditionally acyclic.

Now suppose there is no satisfying assignment to the 3-SAT instance. We will show that there is some X such that condition 2 of Definition 4 holds for all assignments $a \in \underline{\mathcal{A}}$ with $\mathcal{A} \subset \mathcal{V}$. Note that for all $X \in \mathcal{U}$ there are no preferences $p \in P$ with $X_p = X$, so X is vacuously $\perp$-rooted. By Lemma 5 this means X is a-rooted for any assignment $a \in \underline{\mathcal{A}}$ where $X \notin \mathcal{A}$. So we are only concerned with assignments to $\mathcal{A}$ such that $\mathcal{U} \subseteq \mathcal{A}$.

First, let $\mathcal{U} = \mathcal{A}$ and consider an arbitrary assignment $a \in \underline{\mathcal{A}}$. Since there is no satisfying assignment to the 3-SAT instance, there is some Z_k (with $k \in \{0, \ldots, m\}$) such that l_j^k does not hold for $j \in \{1, 2, 3\}$. As such, there is no preference $p \in P$ with $\rho_p \bowtie a$ and $Z_k \in \Omega_p$. So no variable is relatively more important than Z_k under a and Z_k satisfies condition 2 vacuously.

Now let $\mathcal{U} \subset \mathcal{A}$ and consider an arbitrary assignment $a \in \underline{\mathcal{A}}$. Clearly, there must be some $Z_i \in \mathcal{A}$ with $Z_{i+1} \notin \mathcal{A}$ (or $Z_0 \notin \mathcal{A}$ for $i = m$). By construction of the reduction, Z_i is the only variable that can be relatively more important than this Z_{i+1} under a. But because $Z_i \in \mathcal{A}$, this means that this Z_{i+1} (or Z_0) satisfies condition 2 and is thus a-rooted. □

4 Polynomial Upper-Approximation

Testing dominance for a pair of outcomes is computationally hard for general conditional preferences (Goldsmith et al. 2005). Hence, there is an interest in developing efficiently computable approximates for the dominance relation. In this context, an *upper approximation* of a set of CP-theory preferences P on variables $\mathcal{V}$ is a binary relation $\gg$ on outcomes such that for any pair of outcomes $o, o' \in \underline{V}$, if $o \geq_P o'$ holds, $o \gg o'$ also holds. When $o \gg o'$ can be determined in polynomial time, $\gg$ is said to be a *polynomial upper approximation*. Note that by definition, $o \gg o'$ does not necessarily mean that $o >_P o'$, it is still possible for o, o' to be incomparable. As such, an upper approximation is "most helpful" when $o \gg o'$ is shown not to hold, since it implies that $o \not>_P o'$ must be true.

Several other existing sufficient conditions for consistency have been shown to admit corresponding polynomial upper approximation when they hold (Boutilier et al. 2004; Wilson 2011; Rauer, Basu, and Honavar 2025). In addition to giving a complexity bound for the conditionally acyclic sufficient condition, the characterization of Theorem 2 for conditional acylcity of preferences allows us to establish a polynomial upper approximation relation for preferences that are conditionally acyclic.

We proceed by first presenting a property of upper approximation relation for a given set of preferences in terms of cs-tree satisfying the preferences.

Proposition 2 ((Wilson 2011)). *Let $\mathcal{R}$ be a (non-empty) set of cs-trees satisfying CP-theory preference statements P.*

Define $\gg^{\mathcal{R}}$ *to be the intersection of* $>_\tau$ *for all* $\tau \in \mathcal{R}$*. Then* $\gg^{\mathcal{R}}$ *is an upper approximation of* $\geq_P$*.*

Based on this property, our objective is to define an upper approximation relation that is consistent with the ordering induced by any cs-tree that satisfies a set of conditionally acyclic preferences. Our definition for upper approximation relation over any pair of outcomes relies on the definition of a-rooted and on identifying a common set of variables whose valuations are identical in the outcome pair being considered. This common set of variables determines the different ways a pair of outcomes can be ordered in different cs-trees that satisfy the given preferences. In other words, we will identify all possible nodes and the associated variables that are responsible for the ordering of the outcomes in some cs-tree and see if they agree on the ordering of the two outcomes being considered.

We show that the common set of variables can be defined as a least fixed point of a monotone function over $\mathcal{V}$. We present a quick overview of monotone set functions, fixed points and algorithmic characterization of least fixed points.

Definition 5. *Given a universe* $\mathcal{U}$ *and a function* $f\colon 2^{\mathcal{U}} \to 2^{\mathcal{U}}$ *is* monotonic *if and only if* $\forall S \subseteq T \subseteq \mathcal{U}$*,* $f(S) \subseteq f(T)$*. A* fixed point *of a monotonic function* f *is a set* $X \subseteq \mathcal{U}$ *such that* $f(X) = X$*. There can be multiple fixed points for a function. The* least fixed point *of* f*, denoted* $\mathtt{LFP}(f)$ *is the subset-minimal fixed point of* f*.*

For a monotonic function f, there is a unique $\mathtt{LFP}(f)$ that can be computed by repeated application of f to the empty set: $f^{|U|}(\emptyset) = \mathtt{LFP}(f)$ (due to Tarski-Knaster Theorem (Tarski 1955)).

Definition 6. *Given a set of CP-theory preferences* P *on variables* $\mathcal{V}$ *and two outcomes* $o, o' \in \underline{\mathcal{V}}$*, let*

$$\mathcal{A}_{eq} = \{Y \in \mathcal{V} \mid o(Y) = o'(Y)\} \text{ and } \Delta(o,o') = \mathcal{V} \setminus \mathcal{A}_{eq}$$

Define the function $\mathtt{E}\colon 2^{\mathcal{A}_{eq}} \to 2^{\mathcal{A}_{eq}}$ *as*

$$\mathtt{E}(S) = S \cup \{X \in \mathcal{A}_{eq} \mid X \text{ is } o(S)\text{-rooted}\}$$

Then let $\mathcal{A} = \mathtt{LFP}(\mathtt{E})$ *and* $a = o(\mathcal{A})$*. We define upper approximation relation,* $\gg^{ca}$ *for conditionally acyclic preferences as follows.*

$$o \gg^{ca} o' \Leftarrow \big[\forall X \in \Delta(o,o').X \text{ is } a\text{-rooted} \Rightarrow o(X) >_a^X o'(X)\big]$$

Note that by Lemma 3 $\mathtt{E}$ is a monotonic function (since for $S \subseteq T$, $o(T) \Vdash o(S)$) and by least fixed point property, $\mathcal{A} \subseteq \mathcal{A}_{eq}$.

Example 4. *Consider the following set of CP-theory preference statements over binary variables* A, B, C, D *and the two outcomes* $o = a_1b_1c_1d_1$ *and* $o' = a_2b_2c_1d_2$*.*

$$\begin{array}{llll} p_4. & & a_1 > a_2 & [\ ] \\ p_5. & a_1: & b_2 > b_1 & [C] \\ p_6. & a_2: & b_2 > b_1 & [\ ] \\ p_7. & & c_1 > c_2 & [D] \\ p_8. & & d_1 > d_2 & [\ ] \end{array}$$

Here $\mathcal{A}_{eq} = \{C\}$ *and* $\Delta(o,o') = \{A,B,D\}$*. Since* C *is relatively less important than* B *according to* p_5*, it is not* $\perp$*-rooted, so* $\mathtt{E}(\emptyset) = \emptyset$ *and* $\mathtt{LFP}(\mathtt{E}) = \emptyset$ *by definition. Clearly* A *is* $\perp$*-rooted. Additionally, since* $b_2 > b_1$*, regardless of* A*'s value (see* p_5 *and* p_6*),* B *is locally consistent under* $\perp$ *and thus also* $\perp$*-rooted. However,* D *is relatively less important than* C *according to* p_7*, so it is not* $\perp$*-rooted. By* p_4*,* $o(A) = a_1 >_\perp^A a_2 = o'(A)$*, and* $o'(B) = b_2 >_\perp^B b_1 = o(B)$ *by* p_5 *and* p_6*. So neither* $o \gg^{ca} o'$ *nor* $o' \gg^{ca} o$ *holds.*

Next, we define the node that *determines* the outcome ordering.

Definition 7 (Dividing Node in cs-tree for pair of outcomes)**.** *Let* τ *be a cs-tree over variables* $\mathcal{V}$*. For an outcome* o*, the* path (in τ) to o *is the sequence of nodes along the directed path from the root node to the leaf node associated with* o*.*

For outcomes o *and* o'*, the node that* ***divides*** o *and* o' *is the last node in the path to* o *which is also in the path to* o'*.*

Observe that, for a pair of outcomes o, o', $o >_\tau o'$ if and only if $o \neq o'$ and $o(X_r) >_r o'(X_r)$ where r is the node that divides o and o'.

Theorem 3 below establishes that $\gg^{ca}$ is an upper approximation when the preference statements are conditionally acyclic. Specifically, any cs-tree satisfying the preferences has a dividing node labeled by one of the variables on which $\gg^{ca}$ compares the pair of outcomes. Moreover, this is the minimal set of such variables. Note that, $\mathtt{LFP}(\mathtt{E})$ (Definition 6) represents all variables that might appear on the path to the dividing node in a cs-tree. Any other variables on which o and o' agree always appear after the dividing node and hence can be ignored. This is illustrated in Example 4, where any cs-tree satisfying the given preferences must have B as an ancestor to C (in the context of a_1). Since o and o' differ on B, the value of C does not matter for the comparison. Additionally, since D is relatively less important than C, o and o' do not need to be compared on D: it will never come before C which will never come before B in any cs-tree which satisfies the preferences.

Theorem 3 (Upper approximation $\gg^{ca}$)**.** *If a set of CP-theory preferences* P *is conditionally acyclic, then* $\gg^{ca}$ *as per Definition 6 is an upper approximation.*

Proof. Consider two arbitrary outcomes o, o', let a be defined as in Definition 6, Consider

$$B = \{X \in \mathcal{V} \mid X \text{ is } a\text{-rooted}\} \text{ and } C = \Delta(o,o') \cap B.$$

Next, consider the set of cs-trees that satisfies the preference set P. Each cs-tree determines the ordering between o and o' and the ordering is established by some r that divides o and o' (Definition 7). Let $R = \{X_r \mid \exists$cs-tree satisfying P with body node r that divides $o, o'\}$. Our objective is to prove that $R = C$, i.e., the ordering between o and o' as determined by Definition 6 is exactly the same as the ordering determined by all cs-trees satisfying P based on the dividing node (Definition 7).

$\boxed{(R \subseteq C):}$ By contradiction suppose there exists a cs-tree τ satisfying P such that the node r dividing o and o' is associated to $(\mathcal{A}_r, a_r, X_r, >_r)$ and $X_r \notin C$.

Since r divides the outcomes, $o(X_r) \neq o'(X_r)$, making $X_r \in \Delta(o,o')$. Thus we must have $X_r \notin B$, i.e. X_r is not a-rooted. Note that because r divides the outcomes, the variables in $\mathcal{A}_r$ cannot divide o and o'. That is, $o(Y) = o'(Y)$ for all $Y \in \mathcal{A}_r$. In other words, $A_r \subseteq A_{eq}$ (as per Definition 6).

Additionally, since τ satisfies P, by Lemma 2, for each node r' associated to $(\mathcal{A}_{r'}, a_{r'}, X_{r'}, >_{r'})$ on the path to r from the root of τ, its $X_{r'}$ is $a_{r'}$-rooted and as $a_{r'} = o(\mathcal{A}_{r'})$, we have $X_{r'}$ is $o(\mathcal{A}_{r'})$-rooted.

This means that $X_{r'} \in \mathtt{E}(\mathcal{A}_{r'})$. Furthermore, $X_{r'} \in \mathtt{E}^i(\emptyset)$, where i is the level of the cs-tree for the node r'. This is based on definition of $\mathtt{E}$ in Definition 6 and each application of function $\mathtt{E}$ starting from the $\emptyset$, variables whose valuations do not distinguish between o and o' is added level by level from the root node of any cs-tree.

Observe that, the valuation of i is bounded by $|\mathcal{A}_{eq}|$ as the function $\mathtt{E}$ in Definition 6 is defined over the powerset of A_{eq} and the number of applications of $\mathtt{E}$ starting from $\emptyset$ guarantees the result to be equal to $\mathtt{LFP}(\mathtt{E})$ is $|A_{eq}|$ (by Tarski-Knaster theorem). Since, $\mathcal{A}_r \subseteq \mathcal{A}_{eq}$, $\mathcal{A}_r \subseteq \mathtt{E}^i(\emptyset)$ and $i \leq |\mathcal{A}_{eq}|$, we have $\mathcal{A}_r \subseteq \mathtt{LFP}(\mathtt{E}) = \mathcal{A}$.

Proceeding further $a = o(\mathcal{A}) = o'(\mathcal{A})$, $a_r = o(\mathcal{A}_r) = o'(\mathcal{A}_r)$ and $a \Vdash a_r$ since $\mathcal{A}_r \subseteq \mathcal{A}$. By Lemma 3, since X_r is not a-rooted, it is also not a_r-rooted. Therefore, by Lemma 2, cs-tree τ does not satisfy the preferences P. This leads to contradiction of our assumption. Hence, the assumption is invalid and $R \subseteq C$ holds.

$\boxed{(C \subseteq R):}$ Let $X \in C$ arbitrary. We will construct a cs-tree satisfying P such that the node r that divides o and o' has $X_r = X$, making $X \in R$. Specifically, since we only know that X is a-rooted, the cs-tree we construct must have $\mathcal{A}_r = \mathcal{A}$ and $a_r = a$ by Lemma 2.

To do so, we define the family of sets $F_i = \mathtt{E}^i(\emptyset) \setminus \mathtt{E}^{i-1}(\emptyset)$ to be the variables added in the ith iteration of applying $\mathtt{E}$ to the empty set (with $\mathtt{E}^0(\emptyset) = \emptyset$). Note that, as $\mathtt{E}^{|A_{eq}|+1}$ is least fixed point of $\mathtt{E}$, $F_i = \emptyset$ when $i \geq |A_{eq}| + 1$. Also, by definition of $\mathtt{E}$, the elements of F_i are $\mathtt{E}^{i-1}(\emptyset)$-rooted but not $\mathtt{E}^{i-2}(\emptyset)$-rooted. As such, they require the elements of F_{i-1} (or some subset of the elements) to have been assigned before them on the path to o.

We will next consider a cs-tree τ where the path to the outcome o will include the variables from F_i in order. On the path to o in the cs-tree, let the first $k = |F_1|$ nodes have $X_k \in F_1$ and let $>_k$ be a total order consistent with $>_{a_k}^{X_k}$. That is, the root of the cs-tree τ has $X_r = Y$ for some $Y \in F_1$, and it has a total order under a_r. Then the child node r' with $a_{r'} \bowtie o$ has $X_{r'} = Z$ for some $Z \in F_1$ with $Z \neq Y$. This continues until no elements of F_1 remain.

Once that occurs, the child node $\bar{r}$ with $a_{\bar{r}} \bowtie o$ is assigned $X_{\bar{r}} = Y'$ for some $Y' \in F_2$; and including $\bar{r}$, $|F_2|$ nodes are present in the path. Continue repeating this for the elements of F_i where $2 \leq i < j$ and $F_j = \emptyset$.

Finally, let r be the node with $\mathcal{A}_r = \mathcal{A}$ and $a_r = a$. Recall that we are considering an arbitrary $X \in C$. By definition X is a-rooted and $X \in \Delta(o, o')$, i.e., $o(X) \neq o'(X)$. Hence, the variable X_r associated with r can be made X, which makes it a dividing node for o and o' and $X \in R$.

For any other node r' in τ that has not been assigned above, choose any Z that is $a_{r'}$-rooted as $X_{r'}$ and corresponding $>_{r'}$ consistent with $>_{a_{r'}}^{Z}$. Since P is conditionally acyclic this is possible by Theorem 2. It remains to prove that the generated cs-tree τ satisfies P.

All nodes in this cs-tree are the same as that constructed in the proof of Theorem 2, except for those on the path to r. For each node r' on this path, there is some i such that $X_{r'} \in F_i$. As such $X_{r'}$ is $o(\mathtt{E}^{i-1}(\emptyset))$-rooted. Additionally, by construction of τ, $a_{r'} \Vdash o(\mathtt{E}^{i-1}(\emptyset))$ so $X_{r'}$ is $a_{r'}$-rooted by Lemma 3. Thus τ is a cs-tree that could have been constructed in the ($\Leftarrow$) portion of Theorem 2's proof and so must be consistent with P following the same reasoning used in said proof.

Therefore,

$$
\begin{aligned}
& o \gg^{ca} o' \\
\Leftrightarrow\ & \left[\forall X \in C,\ o(X) >_a^X o'(X)\right] \text{ by Definition 6} \\
\Leftrightarrow\ & \left[\forall X \in R,\ o(X) >_a^X o'(X)\right] \text{ as } R = C \\
\Leftrightarrow\ & \left[\forall \tau \text{ satisfying } P,\ o >_\tau o'\right] \text{ as } R \text{ is dividing node} \\
\Leftrightarrow\ & \left[o \gg^{\mathcal{R}} o'\right] \text{ where } \mathcal{R} \text{ is all satisfying cs-tree for } P
\end{aligned}
$$

□

Theorem 4 (Polynomial Upper approximation $\gg^{ca}$)**.** *If a set of CP-theory preferences P is conditionally acyclic, then $\gg^{ca}$ as per Definition 6 is polynomial-time computable.*

Proof. To show that $\gg^{ca}$ is a polynomial upper approximation, note that by the proof of Corollary 1 determining whether a variable is a'-rooted for a given assignment a' can be done in polynomial time. As such, determining $\mathtt{E}(S)$ for some S can also be done in polynomial time by checking whether X is $o(S)$-rooted for all $X \in \mathcal{A}_{eq} \setminus S$ since $|\mathcal{A}_{eq}| \in O(|\mathcal{V}|)$. By Tarski-Knaster Theorem, calculating $\mathcal{A} = \mathtt{LFP}(\mathtt{E})$ can be done in $|\mathcal{A}_{eq}| \in O(|\mathcal{V}|)$ iterations. Therefore, finding valuations of variables in $\mathtt{LFP}(\mathtt{E})$ for a given pair of outcomes can be done in $O(PV^3)$ time.

Additionally, $\Delta(o, o')$ can be found in $O(|\mathcal{V}|)$ time by simply comparing the values of each variable in o, o'. For $X \in \Delta(o, o')$, determining whether X is a-rooted can be done in polynomial time, and for each X that is, determining whether $o(X) >_a^X o'(X)$ can also be done in polynomial time. So $o \gg^{ca} o'$ can be decided in polynomial time, making it a polynomial upper approximation. □

Recall that, any set P of preferences that is fully acyclic (Boutilier et al. 2004), $\mathtt{cuc}$-acyclic (Wilson 2006) or $\mathtt{cc}$-acyclic (Rauer, Basu, and Honavar 2025), is also conditionally acyclic. The upper-approximations for P in all these cases is characterized using satisfiable cs-trees as per Proposition 2. Hence, if $\gg$ is an upper approximations based on the existing cs-tree based sufficient condition for consistency, then $o \gg^{ca} o' \quad \Rightarrow \quad o \gg o'$. This can be seen by the outcomes of Example 4, where all existing upper approximations would determine that $o \gg o'$ holds, since they only compare the values of $o(A)$ and $o'(A)$.

5 Consistency with Generalized cs-tree

5.1 set-labeled cs-tree: A Generalization

While cs-trees provide a useful framework through which to check the consistency of a set of preferences, due to only being a sufficient condition there are still consistent sets for which a cs-tree does not exist.

Example 5. *Consider the preferences on variables X_1 and X_2 with domains of size 2.*

$$
\begin{array}{l}
X_1 = x_1 : \; X_2 = x_2 > X_2 = x_2' \\
X_1 = x_1' : \; X_2 = x_2' > X_2 = x_2 \\
X_2 = x_2 : \; X_1 = x_1 > X_1 = x_1' \\
X_2 = x_2' : \; X_1 = x_1' > X_1 = x_1
\end{array}
$$

These statements are consistent with both x_1x_2 and $x_1'x_2'$ being preferred over $x_1'x_2$ and x_1x_2'. However, there is no cs-tree which satisfies these preferences, since neither X_1 nor X_2 are $\bot$-rooted.

To cover such cases, we introduce a more general structure called a set-labeled cs-tree. Intuitively, a set-labeled cs-tree is similar to a cs-tree, except that each node has a corresponding set of variables, not just a single one. As such, the total order on each node is on all assignments to its corresponding set of variables. By restricting which sets are allowed to label a node, different families of set-labeled cs-trees can be formed. We give a family of set-labeled cs-trees which allows us to define a chain of weakening sufficient conditions for the consistency of a set of preferences. These sufficient conditions eventually become equivalent to the general problem of checking whether the set of preferences is consistent or not.

Definition 8. *A set-labeled cs-tree τ on variables $\mathcal{V}$ is a rooted directed tree where each internal (i.e. non-leaf) node r is associated with a tuple $(\mathcal{A}_r, a_r, \mathcal{B}_r, >_r)$ where*

- *$\mathcal{A}_r \subset \mathcal{V}$ is a set of variables ($\mathcal{A}_r = \emptyset$ for the root node) and a_r is an assignment to said variables*
- *$\mathcal{B}_r \subseteq (\mathcal{V} \setminus \mathcal{A}_r)$ is a nonempty set of variables associated with r and $>_r$ is a total order on $\underline{\mathcal{B}_r}$*

Each internal node r has a child associated with every value in $\underline{\mathcal{B}_r}$, which are ordered according to the relation $>_r$. For child r' associated with value $b \in \underline{\mathcal{B}_r}$, $a_{r'} = a_r \cup \{b\}$. The leaf nodes also have an assignment to all variables of $\mathcal{V}$ and are thus associated with a unique outcome. These leaf nodes have an ordering $>_\tau$ which follows from those of their parents. So every set-labeled cs-tree defines an ordering on the outcomes and we say that this ordering satisfies *a set of CP-theory preference statements P, if $\geq_P \subseteq >_\tau$.*

These set-labeled cs-trees are similar to cp-trees as defined by Wilson (Wilson 2009; Wilson 2012) with one difference. Namely, nodes are associated with a total order $>_r$, unlike the nodes of Wilson's cp-trees which are associated with weak orders. As such set-labeled cs-trees form a total order on the outcomes, whereas cp-trees form a weak order.

Given a collection of sets of variables $\mathcal{Y} \subseteq \mathcal{P}(\mathcal{V})$, a *$\mathcal{Y}$-set-labeled cs-tree* is a set-labeled cs-tree where each node r has $Y_r \in \mathcal{Y}$. For $k \in \mathbb{N}$, a *k-cs-tree* is a $\mathcal{Y}$-set-labeled cs-tree with $\mathcal{Y} = \{\mathcal{S} \subseteq \mathcal{V} \mid |\mathcal{S}| \leq k\}$.

Notice that cs-trees are equivalent to 1-cs-tree. Additionally, the path and dividing node can be defined for set-labeled cs-trees analogously to their definition for cs-trees in Definition 7.

Example 6. *The following 2-cs-tree satisfies the preferences of Example 5.*

$$
\begin{tikzcd}[column sep=small]
 & & \{X_1, X_2\} \arrow[dll] \arrow[dl] \arrow[dr] \arrow[drr] & & \\
x_1x_2 & x_1'x_2' & & x_1x_2' & x_1'x_2
\end{tikzcd}
$$

Clearly the existence of a set-labeled cs-tree satisfying a set of preferences P implies that P is consistent. Using k-cs-tree we define a chain of sufficient conditions for consistency.

Theorem 5. *Given a set of CP-theory preferences P and $k \in \mathbb{N}$, if there exists a k-cs-tree satisfying P, then P is consistent. When this is the case, we say that P is k-conditionally acyclic.*

Proof. Follows from the definition of a set-labeled cs-tree satisfying P and the definition of a consistent set of preferences. $\square$

Note that 1-conditionally acyclic is the same as conditional acyclicity. Additionally, when $k = |\mathcal{V}|$, k-conditionally acyclic is the same as general consistency.

Corollary 2. *A set of preferences P is consistent if and only if it is $|\mathcal{V}|$-conditionally acyclic.*

Proof. If P is consistent, there exists some total order $>_t$ consistent with $\geq_P$. Clearly the $|\mathcal{V}|$-cs-tree with a single root node r with $\mathcal{B}_r = \mathcal{V}$ and $>_r = >_t$ will satisfy P. By Theorem 5, if P is $|\mathcal{V}|$-conditionally acyclic, then it is consistent. $\square$

5.2 Characterization of *k*-Conditional Acyclicity

Analogous to the cs-trees, we discuss the properties of set-labeled cs-trees. A similar relationship between this general tree structure and the preferences it satisfies, in terms of existence of *root*-ed variable can be established. This relationship, in turn, forms the basis for deriving the necessary and sufficient conditions for determining k-conditional acyclicity of preferences.

Proposition 3. *Let P be a set of CP-theory preferences and let τ be a set-labeled cs-tree. Then τ satisfies P if and only if the following two conditions hold.*

1. *for any $p \in P$ and any outcome o such that $o \Vdash \rho_p$, on the path from the root to o, X_p appears before or on the same level as each element of Ω_p*
2. *for any body node r and $p \in P$ such that $X_p \in \mathcal{B}_r$ and $\rho_p \bowtie a_r$, for all $b \in \underline{\mathcal{B}_r \setminus (\Omega_p \cup \{X_p\})}$ with $a_r b \bowtie \rho_p$ and all $z, z' \in \underline{\mathcal{B}_r \cap \Omega_p}$, it is the case that $bx_pz >_r bx_p'z'$*

Proof. Similar to the proof of Proposition 13 given by Wilson (2011) with some modification to handle sets of variables. $\square$

We define an analogous term to a-rooted for sets of variables. This will then be used to give an equivalent characterization of k-conditionally acyclic similar to the characterization of conditional acyclicity given by Theorem 2.

Definition 9. *Given a set of CP-theory preferences P, $k \in \mathbb{N}$, a set of variables $\mathcal{A} \subset \mathcal{V}$, and an assignment $a \in \underline{\mathcal{A}}$, we say that a set of variables $\mathcal{B} \subseteq \mathcal{V} \setminus \mathcal{A}$ is a-k-rooted if the following all hold*

1. $0 < |\mathcal{B}| \leq k$
2. $>_a^{\mathcal{B}}$ *is irreflexive*
3. $\forall Z \in \mathcal{V}$ *such that Z is relatively more important than X under a for some $X \in \mathcal{B}$, $Z \in \mathcal{A} \cup \mathcal{B}$*

We update Lemmas 2 & 3 to apply to a-k-rooted sets.

Lemma 4. *If a k-cs-tree τ satisfies CP-theory preferences P, then for every body node $r \in \tau$, $\mathcal{B}_r$ is a_r-k-rooted.*

Proof. By contrapositive, suppose that there is some body node $r \in \tau$ such that $\mathcal{B}_r$ is not a_r-k-rooted. Showing condition 2 or condition 3 of Definition 9 does not hold for $\mathcal{B}_r$ follows similar reasoning as the proof of Lemma 2. ☐

Lemma 5. *Given a set of CP-theory preferences P, $k \in \mathbb{N}$, a set of variables $\mathcal{B} \subset V$, $\mathcal{A} \subseteq \mathcal{V} \setminus \mathcal{B}$, $a \in \underline{\mathcal{A}}$, let $\mathcal{B}$ be a-k-rooted. Then for any assignment $a' \Vdash a$ where $a' \in \underline{\mathcal{A}'}$ and $\mathcal{B} \not\subseteq \mathcal{A}'$, $\mathcal{B} \setminus \mathcal{A}'$ is a-k-rooted.*

Proof. Clearly this holds if $a = a'$, so let $a \neq a'$. Denote $\mathcal{B}' = \mathcal{B} \setminus \mathcal{A}'$ and since $\mathcal{B} \not\subseteq \mathcal{A}'$, it must be the case that $\mathcal{B}' \neq \emptyset$. So $0 < |\mathcal{B}'| \leq |\mathcal{B}| \leq k$. Showing the other two conditions follows similar reasoning as the proof of Lemma 3. ☐

This allows us to show a version of Theorem 2 for set-labeled cs-trees.

Proposition 4. *Given a set of preferences P and $k \in \mathbb{N}$, P is k-conditionally acyclic if and only if $\forall \mathcal{A} \subset \mathcal{V}$, $\forall a \in \underline{\mathcal{A}}$, $\exists \mathcal{B} \subseteq \mathcal{V} \setminus \mathcal{A}$ such that $\mathcal{B}$ is a-k-rooted.*

Proof. $(\Rightarrow)$: Let τ be a k-cs-tree satisfying P. Consider arbitrary $\mathcal{A} \subset \mathcal{V}$ and $a \in \underline{\mathcal{A}}$. Since $\mathcal{A} \neq \mathcal{V}$, there exists some node r in τ such that $a \Vdash a_r$ and $\mathcal{B}_r \not\subseteq \mathcal{A}$ (i.e. this is the first node on the path to any outcome extending a where there is a $X \in \mathcal{B}_r \cap (\mathcal{V} \setminus \mathcal{A})$). By Lemma 4, $\mathcal{B}_r$ is a_r-k-rooted. Additionally, since $\mathcal{B}_r \not\subseteq \mathcal{A}$ and $a \Vdash a_r$, $\mathcal{B}_r \setminus \mathcal{A}$ is a-k-rooted by Lemma 5, as desired.

$(\Leftarrow)$: Suppose the RHS above holds and construct a k-cs-tree as follows. For each body node r with corresponding $\mathcal{A}_r$ and a_r, let $\mathcal{B}_r$ be a a_r-k-rooted set. Let $>_r$ be a total order consistent with $>_{a_r}^{\mathcal{B}_r}$. By the RHS and condition 2 of Definition 9 such a set and total order must exist. Additionally, by condition 1 of Definition 9, $0 < |\mathcal{B}_r| \leq k$ making τ a valid k-cs-tree. Showing that τ satisfies P can be done in a manner similar to the $(\Leftarrow)$ proof of Theorem 2 but using Proposition 3. ☐

This characterization shows that for any fixed k, determining whether a set of preferences is k-conditionally acyclic is coNP-complete. However, in the worst case it might require one to check all subsets of size k. As such, doing so is most practical for smaller k.

Corollary 3. *For a fixed $k \in \mathbb{N}$, determining whether a set of CP-theory preferences P is k-conditionally acyclic is coNP-complete.*

Proof. Given a certificate a, verifying that the condition does not hold can be done in a manner similar to that of Corollary 1, except that sets of variables of size k or less must be checked. Since k is constant this does not change the polynomial time of this verification. The hardness proof uses the same reduction and a similar argument as in Corollary 1. The main difference arises when the number of clauses of the 3-SAT instance is $\leq (k-1)$ since it could result in $\mathcal{Z} = \{Z_0, \ldots, Z_m\}$ being considered in the $\mathcal{U} = \mathcal{A}$ case. However, $>_a^{\mathcal{Z}}$ is irreflexive for all $a \in \mathcal{A}$ if and only if there is no satisfying assignment to the 3-SAT instance. ☐

It is clear that for any $k \in \mathbb{N}$ if a set of preferences is k-conditionally acyclic, then it is $(k+1)$-conditionally acyclic. Interestingly, the converse does not always hold.

Proposition 5. *Given a set of CP-theory preferences P, for any $k \in \mathbb{N}$, if P is k-conditionally acyclic, then it is $(k+1)$-conditionally acyclic. The converse does not necessarily hold, i.e. for any $k \in \mathbb{N}$ there is some set of preferences P such that P is $(k+1)$-conditionally acyclic, but not k-conditionally acyclic.*

Proof. Since P is k-conditionally acyclic, there is a k-cs-tree τ that satisfies P. Because $\{\mathcal{S} \subseteq \mathcal{V} \mid |\mathcal{S}| \leq k\} \subseteq \{\mathcal{S} \subseteq \mathcal{V} \mid |\mathcal{S}| \leq k+1\}$, τ is also a $(k+1)$-cs-tree by definition. So P is $(k+1)$-conditionally acyclic.

To show the converse does not hold we give a set of preferences P that is $(k+1)$-conditionally acyclic but not k-conditionally acyclic. Let there be $k+1$ binary variables and for each variable X_i, let P contain the following two sets of preferences $p_{i,1}$ and $p_{i,2}$.

$$\bigcup_{j \in M} \{x'_j\} \bigcup_{0 \leq j \leq k, j \notin M, j \neq i} \{x_j\} : x_i > x'_i[\emptyset] \qquad \forall M \subset (\{0, \ldots, k\} \setminus \{i\})$$

$$\bigcup_{0 \leq j \leq k, j \neq i} \{x'_j\} : x'_i > x_i[\emptyset]$$

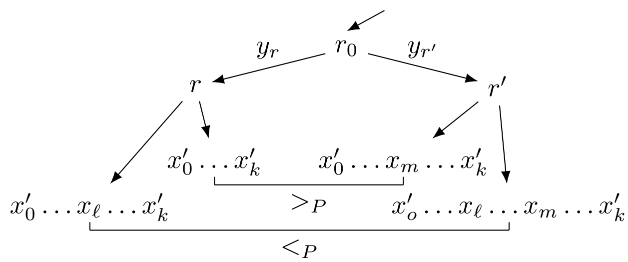
The second set of preferences says that if all other X_j are assigned to value x'_j, then X_i is also preferred to be x'_i over x_i. This induces the outcome $o = x'_0 \ldots x'_k$ to be preferred over any outcome with exactly one variable X_j whose value is x_j (i.e. non-prime). The first set of preferences says that for all other assignments to the other variables (i.e. all assignments to the other variables with at least one variable assigned to be non-prime), X_i is preferred to be non-prime over prime. Intuitively, this induces outcome $o >_P o'$ as long as the number of variables assigned to be non-prime is higher in o than in o' and both outcomes have at least one such non-prime variable.

Consider the $(k+1)$-cs-tree τ with only the root node containing all $(k+1)$ variables. For o, o' define $o >_r o'$ if neither outcome is $\bar{o} = x'_0 \ldots x'_k$ and the number of variables assigned to be non-prime is higher in o than in o' (break ties using some lexicographic ordering). Otherwise, let $\bar{o} >_r o$ for all other $o \neq \bar{o}$. This total order is consistent with that induced by P so τ satisfies P and P is $(k+1)$-conditionally acyclic.

By contradiction suppose there is a k-cs-tree τ that satisfies P. Since there are $k+1$ variables and τ is a k-cs-tree, there must exist at least two nodes on the path to

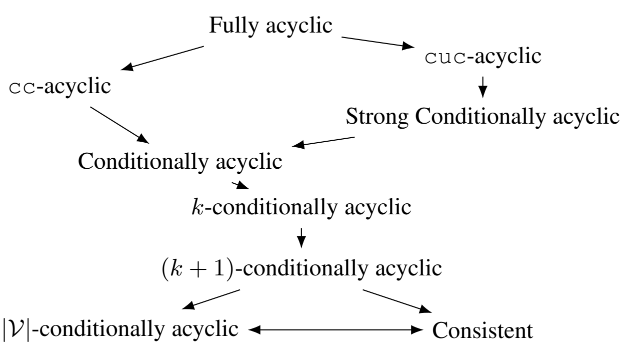
$o = x'_0 \dots x'_k$ and let r be a non-root node on said path. Fix some $X_\ell \in \mathcal{B}_r$. Consider the parent node of r, denoted by r_0, and let $X_m \in \mathcal{B}_{r_0}$. WLOG suppose that $m > \ell$. Let $y_r = \bigcup_{X_j \in \mathcal{B}_{r_0}} x'_j$ and $y_{r'} = x_m \bigcup_{X_j \in \mathcal{B}_{r_0} \setminus \{X_m\}} x'_j$. and note that $a_r = a_{r_0} y_r$. Also let body node r' be the child of r_0 associated with $a_{r'} = a_{r_0} y_{r'}$. Note that a_r and $a_{r'}$ differ only on the value of X_m.

Regardless of the total orders of r_0 and it's descendants, $>_\tau$ will always order one of the pairs of outcomes of the figure below in a manner contradictory to $\geq_P$. So no k-cs-tree can satisfy the given preferences.



Note that in the above figure, intermediate nodes are omitted and all variables X_i take value x'_i in the outcomes unless explicitly shown otherwise. □

This gives the following implication order among known sufficient conditions for consistency.



6 Conclusion

We resolve a long-standing open problem by presenting a necessary and sufficient condition for conditional acyclicity of CP-theory preferences characterized by cs-tree. We further generalize cs-trees to establish a framework realizing progressively weaker sufficient conditions for consistency of CP-theory preferences. Finally, we develop a poly-time computable upper-approximation for conditionally acyclic preferences, the strongest among the ones presented in the existing literature. As part of future work, we plan to study upper approximations induced by different sufficient conditions and analyze the trade-off between computational efficiency and approximation quality.

AI Declaration

Generative AI was not used in the process of making this paper.

Acknowledgments

This work is supported in part by NSF grant IIS 2225823.

References

Boutilier, C.; Brafman, R. I.; Domshlak, C.; Hoos, H. H.; and Poole, D. 2004. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research* 21:135–191.

Brafman, R. I.; Domshlak, C.; and Shimony, S. E. 2006. On graphical modeling of preference and importance. *Journal of Artificial Intelligence Research* 25:389–424.

Goldsmith, J.; Lang, J.; Truszczynski, M.; and Wilson, N. 2005. The computational complexity of dominance and consistency in CP-nets. In *Proceedings of International Joint Conference on Artificial Intelligence*, 144–149.

Rauer, E.; Basu, S.; and Honavar, V. 2025. Checking consistency of cp-theory preferences in polynomial time. *Proceedings of the AAAI Conference on Artificial Intelligence* 39(14):15126–15133.

Santhanam, G.; Basu, S.; and Honavar, V. 2016. *Representing and reasoning with qualitative preferences: Tools and applications*, volume 10 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*. United States: Morgan and Claypool Publishers.

Tarski, A. 1955. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*.

Wilson, N. 2004. Extending CP-nets with stronger conditional preference statements. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, 735–741. AAAI Press.

Wilson, N. 2006. An efficient upper approximation for conditional preference. In *Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy*, volume 141, 472–476.

Wilson, N. 2009. Efficient inference for expressive comparative preference languages. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, IJCAI'09, 961–966. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Wilson, N. 2011. Computational techniques for a simple theory of conditional preferences. *Artificial Intelligence* 175(7):1053–1091. Representing, Processing, and Learning Preferences: Theoretical and Practical Challenges.

Wilson, N. 2012. Importance-based semantics of polynomial comparative preference inference. In *Proceedings of the 20th European Conference on Artificial Intelligence*, ECAI'12, 852–857. NLD: IOS Press.