Modelling Integration of Responsible AI Values for Ethical Decision Making

Yousef Sojasi Taheri, Gauvain Bourgne, Jean-Gabriel Ganascia

Modelling Integration of Responsible AI Values for Ethical Decision Making

Yousef Taheri Gauvain Bourgne
Jean-Gabriel Ganascia

Sorbonne Universite
CNRS, LIP6, 75005 Paris, France
{yousef.taheri, gauvain.bourgne, Jean-gabriel.ganascia}@lip6.fr

A principal requirement to achieve trustworthy AI is to consider ethical aspects in the design and development of AI systems. This is particularly challenging when it comes to automatic decision making and requires appropriate tools to incorporate these aspects in the decision process. One way to address this issue is to evaluate the available alternatives in terms of their adherence to AI-specific moral values, i.e. responsible-AI values. In this article, we propose a hierarchical model to incorporate responsible AI values as ethical criteria in the decision-making process. We adopt our framework on the problem of ranking several recommendation systems according to their adherence to responsible AI values. We use Answer Set Programming (ASP) to formally represent the implementation of our framework on this problem and derive a ranking of alternatives using logical reasoning.

1 Introduction

As AI progresses, it is being used more and more in applications that have an effect on people and/or society. These applications range from medical diagnosis and criminal investigations to recommendations. In these cases, it is essential to factor ethical considerations into the decision-making process to increase the accountability of AI systems.

One way to address this is to consider a pluralistic set of moral values in the decision-making process, or the so-called responsible-AI values that should be respected by AI autonomous agents. Many of the ethics guidelines discuss these values, e.g. privacy, fairness, explainability, etc. [10],[1]. These values must be respected at all stages of AI design and development to ensure its responsible use. In order to compare available options by considering responsible AI values in an automatic setting, we need to incorporate these values into the reasoning and decision process.

Several works on computational ethics have used a pluralistic view in their implementation of utilitarianism theories [3], [4]. These methods evaluate actions by quantifying goodness (or utility) on a cardinal scale and combine them by weightings and arithmetic calculations. They use elicited values for specific cases [14] or random numbers to explore ethical dilemmas without proposing a method to actually obtain them [8], [6]. In certain cases, it is possible to compare options rationally by quantification. However, it may add arbitrary variables to the process that can bias the decision and lead to counterintuitive results.

In this article, we propose a framework based on multiple criteria with a hierarchical structure to integrate responsible AI values into decision making. The hierarchical criteria structure provides an explicit and explanatory representation of moral values. Each criterion represents a certain aspect of a moral value that can be used to classify alternatives on an ordinal scale. We then investigated several approaches for aggregating preferences and propose applicable methods based on the assumption on the criteria. Lastly, we adopt our framework on a case study, where the goal is to ethically rank a set of

Submitted to:
CME 2023

recommendation algorithms according to their adherence to responsible AI values. To implement this use case, we make use of Answer Set Programming (ASP) [12], which is a paradigm of knowledge representation and reasoning with expressive formalism and efficient solvers.

The remainder of the paper is organized as follows. Section 2 presents related work. In Section 3 we explain a formalization of our framework, and propose some aggregation approaches in Section 3.1. In Section 4 we describe a case study of adopting our framework for ethical classification of various recommendation systems. In Section 5 we discuss the conclusions and future perspectives.

2 Related Works

A line of research in computational ethics is to model moral theories, for example consequential or deontological, in order to judge the ethical permissibility of agent actions [17]. Consequentialist theories, e.g. act utilitarianism, often require a notion of utility to identify the best possible action. Most current work generally proposes a quantified model for the evaluation of utility which is supposed to be given [14], obtained from a source[13], or is obtained by weighting and summing utilities according to some components such as moral values [6]. Quantification and arithmetic aggregation can be reasonable in some cases, however, it may cause counterintuitive results in general. We avoid this issue in our framework by modeling the evaluation of alternatives as an ordinal preference relation.

We have drawn inspiration from multi-criteria decision-making(MCDM) methods for the design of our framework. These approaches are used to analyze and solve decision problems involving multiple criteria [9], [5]. We adopt a hierarchical structure of criteria as in Analytic Hierarchy Process (AHP)[18]. AHP is used to derive relative priorities from continuous and discrete paired comparisons in hierarchical structures at multiple levels [16]. In MCDM, alternatives can be evaluated in several ways; we use ordinal rankings of alternatives [11], which enables us to make use of voting rules in the theory of social choice. There are a broad number of voting rules, e.g. plurality, approval, Copeland's rule, etc. Several properties are desired in a voting rule like Pareto's efficiency, monotonicity, Condorcet principle, etc. The latter is particularly required in our framework, which means that a voting rule should select the alternative that beats every other alternative in pairwise comparisons [7]. Most of the well-known voting rules and their properties have already been discussed in the literature on social choice theory [15].

3 Framework

In this section, we describe our framework for the integration of moral values in the decision-making process. We are interested in evaluating the adherence or alignment of alternatives to a set of moral values. The alternatives have different significations depending on the context; however, they represent or refer to an action or a series of actions. To incorporate moral values into the decision process, we view them as criteria in a hierarchical structure that can be decomposed into more basic measurable sub-criteria. Evaluation is modeled as an ordinal preference relation over alternatives. We introduce multiple approaches for aggregating preferences that can take into account the priority of subcriterion. More precisely, our framework is designed to classify a set of given alternatives $\mathscr{A}$, $|\mathscr{A}| \geq 2$ according to their adhesion to a set of moral values. A *hierarchy of values setting H* on a set of alternatives $\mathscr{A}$ is represented by the tuple $\langle \mathscr{A}, \langle N, \mathbf{R} \rangle, \rho, \mathbf{\Psi} \rangle$.

Criteria Hierarchy is a tree-like graph represented by the tuple $\langle N, \mathbf{R} \rangle$ where N is the set of nodes that represent criteria, $\mathbf{R} \subseteq N \times N$ is the child relation that assigns to each parent node its child nodes, in other words, this relation associates each criterion with its sub-criterion in a hierarchical structure.

We denote the set of the sub-criteria or the children of a node n by $r(n) = \{x|(n,x) \in \mathbf{R}, n \in N\}$. The root node in this structure is considered the super-criterion and is denoted by $\varepsilon \in N$. The nodes at the bottom of the structure (leaves) represent evaluative criteria, i.e. criteria that can be evaluated based on the characteristics of the given alternatives. The set of leaf nodes or leaf criteria is represented by $L = \{n \in N | r(n) = \emptyset\}$.

Leaf Criteria Assessment ρ is the function to evaluate alternatives based on the bottom-level or leaf criteria. This evaluation is modeled as an ordinal preference over alternatives, and is represented by $\rho : L \longmapsto 2^{\mathscr{A}\times\mathscr{A}}$, where $\rho(l)$, $\forall l \in L$ is a preference relation, given as a complete pre-order.

Aggregation functions $\boldsymbol{\Psi} = \{\psi_n\}_{n \in N \setminus L}$ are a family of aggregators, each node uses to combine the preference of its children. For each parent node n, we associate an aggregator $\psi_n : (2^{\mathscr{A}\times\mathscr{A}})^{|r(n)|} \longmapsto 2^{\mathscr{A}\times\mathscr{A}}$.

3.1 Preference Aggregation

Here, we describe several aggregation methods that can be used in our hierarchical model to obtain the preference of a parent node based on the preference of its children. To maintain the generality of the problem, we consider a finite set of criteria $C = \{1, \ldots, |C|\}$, such that each criterion $i \in C$ specifies a transitive and complete preference over the set of alternatives $P_i \subseteq D$, where $D = 2^{\mathscr{A}\times\mathscr{A}}$. We discuss different rules to aggregate the set of these preferences, as noted by $\mathbb{P} = \langle P_1, \ldots, P_{|C|} \rangle \in D^{|C|}$, according to available information on the importance and priorities of the criteria. The preference obtained by the aggregation rule is denoted by $P \subseteq D$. Each method is based on an assumption on the available information about the importance of criteria.

3.2 Voting Approaches

When all criteria have equal importance, their corresponding preferences can be viewed as votes over alternatives. In such a case, the aggregation rules in social choice and voting theory can be used to combine the preferences of the criteria. We denote such an aggregator by $\psi_v : D^{|C|} \longmapsto D$ and the obtained preference by $P^v = \psi_v(\mathbb{P})$. There are general properties that are desired in voting rules, like Pareto's efficiency, monotonicity, etc. An important property in our case is the Condorcet principle, a lack of this property may cause a cyclic preference and lead to loss of transitivity. We limit the rules to the ones that satisfy these properties, e.g. Copeland's rule, max-min, etc. For example, the Copeland rule chooses the alternative that wins the most pairwise majority, for each $a, b \in \mathscr{A}$ let r_{ab} be defined as follows:

$$r_{ab} = \begin{cases} 1 & |\{i \in C | aP_ib \wedge \neg bP_ia\}| > |\{i \in C | bP_ia \wedge \neg aP_ib\}| \\ \frac{1}{2} & |\{i \in C | aP_ib \wedge \neg bP_ia\}| = |\{i \in C | bP_ia \wedge \neg aP_ib\}| \\ 0 & |\{i \in C | aP_ib \wedge \neg bP_ia\}| < |\{i \in C | bP_ia \wedge \neg aP_ib\}| \end{cases} \tag{1}$$

Then, according to Copeland's rule, an alternative wins in pairwise comparison if its overall score defined by the following relation is higher, i.e. $aP^vb \Leftrightarrow \sum_{c \in \mathscr{A}} r_{a,c} \geq \sum_{c \in \mathscr{A}} r_{b,c}$. Note that this is an example of a voting rule that has our desired properties, other rules that satisfy the Condorcet principle can be used interchangeably.

3.3 Voting with Dominant Criteria

A possible situation is that there may be a group of criteria that dominate the others, meaning that their vote has an absolute priority over other criteria. We suppose that there are $k \leq |C|$ groups of criteria such that $C = \cup_{i\in\{1,\dots,k\}} C_i$, and the priority relation between the groups is given by a total strict order $\succ^s \subseteq 2^C \times 2^C$.

One way to aggregate criteria preferences in this case is to obtain the votes in each group and rank them in lexicographic order. More precisely, consider the *group* aggregation function $\psi_g : D^{|C|} \times C^2 \longmapsto D$ and its resulting preference $P^g = \psi_g(\mathbb{P}, \succ^s)$. The aggregation method in this case can be formulated as follows, that is, a combination of voting rule and lexicographic aggregation, i.e. $a\, P^g\, b \iff \exists\, i \in \{1,\dots,k\} \wedge a\, P^v_{C_i}\, b\, \wedge (\forall\, j \in \{1,\dots,k\} \wedge C_j \succ^s C_i \Rightarrow a\, P^v_{C_j}\, b)$, where $a\, P^v_X\, b = \psi_v(\mathbb{P}^X)$, $\mathbb{P}^X \subseteq D^{|X|}$ is the vector of the preference of the criteria in a set $X \subseteq C$, and ψ_v is the aggregation function based on a voting rule as discussed in Section 3.2. When the only partition is C, then P^g is equivalent to P^v.

4 A Use Case Model

In this section, we show how our framework can be used for ethical evaluation of moral values. We describe the evaluation procedure on a use case model of ranking recommender systems, which concerns privacy, fairness, and performance of the systems. Our framework in such context provides an explicit evaluation process that is explainable in terms of responsible AI values by indicating what values are taken into account.

Use Case Description: A company with an online employment platform collaborates with multiple partners to provide job recommendations to its users. Their partners change regularly and/or update their recommender systems very often. The company uses an automatic ranking process to select the best system, taking into account the responsible AI values. We show the evaluation process of our framework in a simplified version of this problem to classify a set of 3 recommender systems i.e. alternatives `{sys1, sys2 ,sys3}`. Alternatives are represented in ASP using the predicate `alt/1`, see Listings C.1

Criteria Hierarchy Here, we describe what moral values are taken into account and by which alternatives are evaluated. These criteria which represent responsible-AI values are the following:

Privacy is one of the most important values in the current stage of AI. This value is considered as a criterion, which is decomposed into 3 sub-criterion. *1) Data Minimization*, based on this criterion, a system that uses fewer categories of data respects the privacy of the user more. *2) Data Sensitivity* is another criterion that counts the number of sensitive categories of data, e.g., political, racial, etc. Processing these types of data entails an increased ethical risk to privacy [2]. *3) Scale and Complexity*, using large-scale processing, that is, big data with multiple unknown sources of data, increases the risk to privacy [2]. According to this criterion, small-scale processing gets a higher rank than large-scale (big data) processing.

Fairness in general has a broad meaning, however, in this context it means having no bias toward any particular group of items (*Item Fairness*) or users (*User Fairness*)[19]. User fairness can also be decomposed into gender fairness (*gender fairness*) and racial fairness (*race fairness*). Fairness in this case is measured using a metric called *statistical parity* which is the difference in the ratio of favorable recommendations between two groups. A recommender system that has a lower statistical parity among monitored groups is ranked higher by fairness criteria.

Performance represents the intrinsic value of the recommendation system related to its beneficence. The notion of performance has various significations; here it represents how fit the recommendations

Parent node	**Children priorities**	**Sys1**	**Sys2**	**Sys3**
Privacy	$\{Sensibility\} \succ^s \{Minimization\} \succ^s \{ScaleComplexity\}$	1	2	1
User Fairness	$\{genderFairness, racialFairness\}$	1	2	2
Fairness	$\{userFairness\} \succ^s \{itemFairness\}$	1	2	2
root	Case 1: $\{Privacy,\ Fairness,\ Performance\}$	1	2	2
root	Case 2: $\{Fairness\} \succ^s \{Privacy\} \succ^s \{Performance\}$	1	2	2
root	Case 3: $\{Fairness, Privacy\} \succ^s \{Performance\}$	1	2	1
root	Case 4: $\{Fairness, Performance\} \succ^s \{Privacy\}$	1	1	2

Table 1: Parent Criteria Evaluations

are and whether the users are satisfied with the recommendations provided. The value composition diagram in this use case is shown in Figure 1, and is represented in ASP using the predicate `child/2`, see C.2.

Leaf Criteria Assessment In order to apply our framework in this case study, we need to evaluate the preference for leaf criteria over alternatives according to their characteristics. First, we represent these characteristics or the knowledge about the recommendation systems that can be used for ethical evaluations. These characteristics include i) the required categories of data, ii) the performance metric of each system, iii) the scale and complexity of the underlying algorithm, and the statistical parity for iv) gender fairness, v) racial equality, and vi) statistical parity for item fairness. Knowledge about available systems and their mentioned characteristics is shown in Table 2 and is represented in ASP using the predicate `has/3`. See Listings C.3. The preference of the leaf criteria is represented by the predicate `pref/3`, as in Listings C.4. The evaluated rankings are represented in Table B.

Aggregation Here, we describe the aggregation process for every parent node. The root node is the top node that represents the final decision criteria. We represent the priority of subvalues using the predicate `childPref/3`, as in Listings C.5. Multiple priority settings are considered for the root node to show the functionality of the aggregation. Votes are aggregated using the rules described in Sections 3.2 and 3.3, are translated into ASP as in the Listings C.6 and C.6. The results obtained for each parent node are shown in Table 1. The results show the ability of our framework to take into account subjective preferences on moral values. For example, in *case 1* the rankings are obtained by collective voting of all moral values, but in *case 2* the ranking of the dominant voter, i.e. fairness, is prioritized.

5 Conclusions

The proposed model for ethical evaluation provides an explicit decision-making process in terms of responsible AI values. An advantage of our model is that the rankings are obtained by logical aggregation rules through a reasoning process, which is essential in ethical decision making. One of the future works is to improve the efficiency of the framework and create an ontology of the principles and prescriptions in AI ethics guidelines for responsible AI values.

References

[1] Ethics guidelines for trustworthy ai. URL `https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai`.

[2] Ethics and data protection, 2021.

[3] M. Anderson, S. L. Anderson, and C. Armen. Towards machine ethics. In *AAAI-04 workshop on agent organizations: theory and practice, San Jose, CA*, 2004.

[4] M. Anderson, S. L. Anderson, and C. Armen. MedEthEx: a prototype medical ethics advisor. In *Proceedings of the national conference on artificial intelligence*, volume 21, page 1759. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[5] M. Aruldoss, T. M. Lakshmi, and V. P. Venkatesan. A survey on multi criteria decision making methods and its applications. *American Journal of Information Systems*, 1(1):31–43, 2013.

[6] F. Berreby, G. Bourgne, and J.-G. Ganascia. A declarative modular framework for representing and applying ethical principles. In *16th Conference on Autonomous Agents and MultiAgent Systems*, 2017.

[7] F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. *Multiagent systems*, 2:213–284, 2012.

[8] N. Cointe, G. Bonnet, and O. Boissier. Ethical Judgment of Agents' Behaviors in Multi-Agent Systems. In *AAMAS*, pages 1106–1114, 2016.

[9] S. Greco, J. Figueira, and M. Ehrgott. *Multiple criteria decision analysis*, volume 37. Springer, 2016.

[10] A. Jobin, M. Ienca, and E. Vayena. The global landscape of AI ethics guidelines. *Nature Machine Intelligence*, 1(9):389–399, 2019.

[11] J. G. Kemeny and L. Snell. Preference ranking: an axiomatic approach. *Mathematical models in the social sciences*, pages 9–23, 1962.

[12] V. Lifschitz. *Answer set programming*. Springer Heidelberg, 2019.

[13] R. Limarga, M. Pagnucco, Y. Song, and A. Nayak. Non-monotonic reasoning for machine ethics with situation calculus. In *AI 2020: Advances in Artificial Intelligence: 33rd Australasian Joint Conference, AI 2020, Canberra, ACT, Australia, November 29–30, 2020, Proceedings 33*, pages 203–215. Springer International Publishing, 2020.

[14] F. Lindner, R. Mattmüller, and B. Nebel. Evaluation of the moral permissibility of action plans. *Artificial Intelligence*, 287:103350, 2020.

[15] H. Nurmi. Voting systems for social choice. In *Handbook of Group Decision and Negotiation*, pages 167–182. Springer, 2010.

[16] T. L. Saaty, L. G. Vargas, et al. *Decision making with the analytic network process*, volume 282. Springer, 2006.

[17] S. Tolmeijer, M. Kneer, C. Sarasua, M. Christen, and A. Bernstein. Implementations in machine ethics: A survey. *ACM Computing Surveys (CSUR)*, 53(6):1–38, 2020.

[18] L. G. Vargas. An overview of the analytic hierarchy process and its applications. *European journal of operational research*, 48(1):2–8, 1990.

[19] Y. Wang, W. Ma, M. Zhang, Y. Liu, and S. Ma. A survey on the fairness of recommender systems. *ACM Transactions on Information Systems*, 41(3):1–43, 2023.

A Figures

The hierarchy of criteria in this use case is shown in Figure 1.

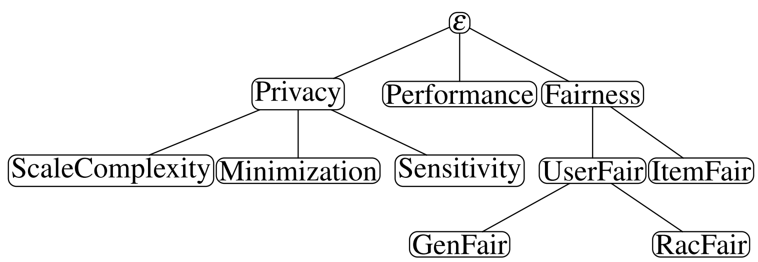

Figure 1: Criteria Hierarchy Diagram

B Tables

Characteristics	**Sys1**	**Sys2**	**Sys3**
Data Categories	dataHabit, activity, interests	interests, politicalBelief	activity, interests
Performance metric	30%	25%	20%
Process Type	large scale	large scale	small scale
Gender parity	0,1	0,2	0,3
Racial Parity	0,2	0,4	0,1
Item Parity	0,3	0,6	0,4

Table 2: Recommender Algorithms' Characteristics

Leaf Criteria	Sys1	Sys2	Sys3
Minimization	2	1	1
Sensibility	1	2	1
ScaleComplexity	2	2	1
Performance	1	2	3
Gender fairness	1	2	3
Racial fairness	2	3	1
Item fairness	1	3	2

Recommender Systems' Leaf Ranking

:

C ASP Codes

C.1 Alternatives

```
alt(sys1).
alt(sys2).
alt(sys3).
```

C.2 Criteria Hierarchy

```
child(root,privacy).
child(root,fairness).
child(root,performance).

child(privacy, sensitivity).
child(privacy, minimization).
child(privacy, scaleComplexity).
child(fairness, item_fairness).
child(fairness, user_fairness).
child(user, racial_fairness).
child(user, gender_fairness).
```

C.3 Systems' Characteristics

This knowledge has been represented in ASP using the predicate `has/3`.

```
has(sys1,requiredData,clickHabit ).
has(sys1,requiredData,activity ).
has(sys2,requiredData,interests ).
...
has(sys1, perfMetric, 30 ).
has(sys2, perfMetric, 25 ).
...
has(sys1, gender_parity, 1 ).
has(sys2, racial_parity, 4 ).
has(sys3, item_parity, 4 ).
```

...

Note that since Clingo has difficulties with grounding float numbers, we represent them using integers. for example:

C.4 Leaf Criteria Evaluations

The preference of the leaf criteria is represented by the predicate `pref/3`.

```
% Minimisation
has(Alt,nbData, N):-
    N = #count{ Data : has(Alt, requiredData, Data)},
    alt(Alt).

pref(minimization,Alt1,Alt2):-
    has(Alt1, nbData, N1),
    has(Alt2, nbData, N2),
    N2>=N1.

%Sensitivity
has(Alt, nbSensitiveData, N):-
    N = #count{ Data : has(Alt, requiredData, Data),
    has(Data, category, sensitiveData)},
    alt(Alt).
pref(sensitivity, Alt1, Alt2):-
    has(Alt1, nbSensitiveData, N1),
    has(Alt2,nbSensitiveData, N2),
    N2>=N1.

%Scale and Complexity
rankAux(largeScale, 2).
rankAux(smallScale, 1).
pref(scaleComplexity,Alt1,Alt2):-
    has(Alt1, processType, Type1),
     has(Alt2, processType, Type2),
     rankAux(Type1, R1),
     rankAux(Type2, R2),
     R2>=R1, alt(Alt1), alt(Alt2).

% Performance
pref(performance, Alt1, Alt2):-
    has(Alt1, perfMetric, P1),
    has(Alt2, perfMetric, P2),
    P1>=P2, alt(Alt1), alt(Alt2).

% Item Fairness
pref(item_fairness,Alt1,Alt2):-
    has(Alt1, item_parity,P1),
    has(Alt2, item_parity,P2),
    P1<=P2, alt(Alt1), alt(Alt2).

%Item Fairness
pref(racial_fairness,Alt1,Alt2):-
    has(Alt1, racial_parity,P1),
    has(Alt2, racial_parity,P2),
    P1<=P2, alt(Alt1), alt(Alt2).

%Gender Fairness
pref(gender_fairness,Alt1,Alt2):-
    has(Alt1, gender_parity,P1),
    has(Alt2, gender_parity,P2),
    P1<=P2, alt(Alt1), alt(Alt2).
```

C.5 Priorities

We represent the priority of sub values using the predicate `childPref/3` in ASP, as shown below.

```
childPref(privacy, sensitivity, minimization).
childPref(privacy, minimization, scaleComplexity).

childPref(fairness, user_fairness, item_fairness).
childPref(user_fairness,racial_fairness,gender_fairness).

childPref(root, fairness, privacy).
childPref(root, privacy, performance).
```

C.6 Aggregation

The grouped aggregation function introduced in Section 3.3 is implemented in ASP the following way.

```
prefGr(Node, Alt1, Alt2):-
    childrenLayers(Node,Layer),
    pVote(Node, Layer, Alt1, Alt2),
    is_dominant(Node, Layer, Alt1, Alt2).

is_dominant(Node, Layer,Alt1,Alt2):-
    not is_dominated(Node, Layer,Alt1,Alt2),
    childrenLayers(Node,Layer),
    pVote(Node, Layer, Alt1, Alt2).

is_dominated(Node, Layer, Alt1,Alt2):-
    childrenLayers(Node,Layer),
    pVote(Node, Layer, Alt1, Alt2),
    superiorThan(Node, Layer1, Layer),
    not pVote(Node, Layer1, Alt1,Alt2).

superiorThan(Node, Layer1, Layer2):-
    childrenLayers(Node, Layer1),
    childrenLayers(Node, Layer2),
    not inferiorThan(Node, Layer1, Layer2).

inferiorThan(Node, Layer1, Layer2):-
    childrenLayers(Node, Layer1),
    childrenLayers(Node, Layer2),
    belongs(Node, Child1 , Layer1),
    belongs(Node, Child2 , Layer2),
    not childPref(Node, Child1, Child2).

childrenLayers(Node, Layer):-
    belongs(Node, _ , Layer).

belongs(Node, Child , Layer):-
    Layer = #count{ Child1 :
        childPref(Node , Child, Child1)},
    child(Node, Child).
```

The predicate `pVote/4` in the code above corresponds to the Copeland's rule in Section 3.2.

```
pVote(Node, Layer, Alt1, Alt2):-
    copeland_score( Node, Layer, Alt1, S1),
    copeland_score( Node, Layer, Alt2, S2),
    S1>S2.
```

```
copeland_score( Node, Layer, Alt, S):-
    S= #sum{ S1:
        nb_pairwise_wins( Node, Layer, Alt, Alt1, N1),
        nb_pairwise_ties( Node, Layer, Alt, Alt1, N2),
        S1 = N1*2+N2,
        alt(Alt1)},
    childrenLayers(Node, Layer), alt(Alt) .

nb_pairwise_ties( Node, Layer, Alt1, Alt2 , N):-
    N= #count{ N1 :
        nb_strict_voters(Node, Layer, Alt1, Alt2 , N1),
        nb_strict_voters(Node, Layer, Alt2, Alt1 , N2),
        belongs(Node, Child, Layer), N1=N2},
    childrenLayers(Node,Layer), alt(Alt1), alt(Alt2).

nb_pairwise_wins( Node, Layer, Alt1, Alt2, N):-
    N= #count{ N1 :
        nb_strict_voters(Node, Layer, Alt1, Alt2 , N1),
        nb_strict_voters(Node, Layer, Alt2, Alt1 , N2),
        belongs(Node, Child, Layer), N1>N2},
    childrenLayers(Node, Layer), alt(Alt1), alt(Alt2).

nb_strict_voters(Node, Layer, Alt1, Alt2, N):-
    N = #count{ Child :
        pref(Child, Alt1, Alt2),
        not pref(Child , Alt2, Alt1),
        belongs(Node, Child, Layer) },
    childrenLayers(Node, Layer), alt(Alt1), alt(Alt2).
```