On Proactive, Transparent and Verifiable Ethical Reasoning for Robots: Supplementary Material

Paul Bremner, Louise A. Dennis, Michael Fisher and Alan F. Winfield

I. Informal Proofs that the Comparison Functions in Box 1 Define Antisymmetric, Transitive Relations

We wish to prove that our comparison relations are antisymmetric and transitive on the assumption that certain properties hold in the real world (e.g. if position x is closer to some danger than position y and position y is closer to it than position z, then position x is closer to the danger than position z.)

Recall we defined (in Box 1 definition 3) a set of relations $\prec_m$ where $m \in \{hd, ro, rd\}$ (hd for human danger distance, ro for robot objective distance, rd for robot danger distance). So, for instance, $x \prec_{hd} y$ means that task y results in the the human being closer to danger than task x. So for any two tasks either $t_1 \prec_m t_2$ or $t_2 \prec_m t_1$ or $t_1 \approx_m t_2$. In particular if $t_1 \approx_m t_2$ then $t_1 \not\prec_m t_2$ and $t_2 \not\prec_m t_1$.

We now seek to prove that $\lhd_{wd}$ is antisymmetric (theorem 1) and transitive 2.

Theorem 1: If $\prec_m$ is antisymmetric for all m then $\lhd_{wd}$ is antisymmetric.

Proof: Let us assume, for a contradiction, that $t_1 \lhd_{wd} t_2$ and $t_2 \lhd_{wd} t_1$. We consider each case in Box 1 definition 1 in turn.

1) $t_1 \prec_{hd} t_2$ (definition 1 case 1). Since $\prec_{hd}$ is antisymmetric then if $t_2 \prec_{hd} t_1$, $t_1 = t_2$ and we are done. Alternatively $t_2 \lhd_{wd} t_1$ because of one of the other cases in 1. However all these cases have $t_1 \approx_{hd} t_2$ as a condition which implies by Box 1 definition 4 that $t_1 \not\prec_{hd} t_2$.
2) $t_1 \approx_{hd} t_2$ and $t_1 \prec_{ro} t_2$ (definition 1 case 1). if $t_2 \prec_{ro} t_1$ then $t_1 = t_2$ (by antisymmetry) and we are done. Otherwise $t_2 \lhd_{wd} t_1$ because of one of the other clauses in definition 1. This means either $t_2 \prec_{hd} t_1$ (case 1) which contradicts $t_1 \approx_{hd} t_2$ or $t_1 \approx_{ro} t_2$ (cases 3 and 4) which contradicts $t_1 \prec_{ro} t_2$.
3) Similar reasoning applies as in case 2.
4) Similar reasoning applies as in case 2 with the additional observation that $<$ is antisymmetric.

■

Theorem 2: If

1) $\prec_m$ is transitive for all m, and
2) $\forall t_1, t_2, t_3 . (t_1 \approx_m t_2 \wedge t_3 \prec_m t_1) \rightarrow t_3 \prec_m t_2$

then $\lhd_{wd}$ is transitive.

Proof: Let us suppose that $t_1 \lhd_{wd} t_2$ and $t_2 \lhd_{wd} t_3$. We need to show that $t_1 \lhd_{wd} t_3$. Once again we proceed by considering each case in definition 1 in turn, but now need to consider these as they apply to both $t_1 \lhd_{wd} t_2$ and $t_2 \lhd_{wd} t_3$.

1) $t_1 \prec_{hd} t_2$ ($t_1 \lhd_{wd} t_2$ because of case 1)
   - $t_2 \prec_{hd} t_3$ ($t_2 \lhd_{wd} t_3$ because of case 1). In this case $t_1 \prec_{hd} t_3$ by the transitivity of $\prec_{hd}$ and so $t_1 \trianglerighteq_{wd} t_3$.
   - $t_2 \approx_{hd} t_3$ and $t_2 \prec_{ro} t_3$ (case 2). Since $t_2 \approx_{hd} t_3$ and $t_1 \prec_{hd} t_2$. $t_1 \prec_{hd} t_3$ (by our third assumption) and so $t_1 \lhd_{wd} t_3$.
   - Similar reasoning applies to cases 3 and 4 with the additional observation that $<$ is transitive.
2) Similar reasoning applies as in case 1.
3) Similar reasoning applies as in case 1.
4) Similar reasoning applies as in case 1.

■

It remains to show that the assumptions we have stated for each theorem hold in our system – i.e, that our relations $\prec_m$ are antisymmetric, transitive and have the property that $\forall t_1, t_2, t_3 . (t_1 \approx_m t_2 \wedge t_3 \prec_m t_1) \rightarrow t_3 \prec_m t_2$. Since these relations are based on distances or times (i.e., $t_1 \prec_{hd} t_3$ if t_3 places the human closer to danger than t_1) then transitivity follows from the transitivity of relations on distances and times. We have made the assumption that the tasks are chosen to make these relations antisymmetric (i.e., no two tasks share the same distance or time valuation[1]).

For our last assumption we need to recall from definition 3 that $t_1 \approx_m t_2$ if for some valuation, v on t_1 and t_2 and some threshold th_m if $v(t_1) < th_m$ and $v(t_2) < th_m$ in the case where some value is to be minimized or $v(t_1) > th_m$ and $v(t_2) > th_m$ in the case where some value is to be maximised.

Theorem 3: For all metrics $m \in \{hd, ro, rd\}$ $\forall t_1, t_2, t_3 . (t_1 \approx_m t_2 \wedge t_3 \prec_m t_1) \rightarrow t_3 \prec_m t_2$

[1] It is further work to adapt our implementation to make this assumption unnecessary.

Proof: We present the proof for when m is a property that is to be minimised. $t_1 \approx_m t_2$ if $v(t_2) < th_m$ and $v(t_1) < th_m$ (from Box 1 definitions 3 and 4). By assumption, $t_3 \prec_m t_1$, since m is to be minimised this means $v(t_3) > th_m$. In this case $t_3 \prec_m t_1$ again by definition 3. ■

II. Semantics of *BDIPython* Belief Bases

BDIPython belief bases are represented as dictionary of key/value pairs, $k \rightarrow v$ where the key, k, is a string and the value, v can be 0/1, a double, d or an array of strings, $[s_1, \ldots, s_n]$.

We assume a logical language $\mathcal{L}$ consisting of constant symbols, c, predicate/function symbols f, (and variable symbols, v, though these are not relevant here). We assume that each string k, s_i appearing in the Python belief base can be interpreted as a constant or function symbol, written as $[\![k]\!]$ etc., (In practice we simply interpret the string "k" as the constant or function k).

These are interpreted as predicates as follows:

$k \rightarrow 0$	$\neg[\![k]\!]$
$k \rightarrow 1$	$[\![k]\!]$
$k \rightarrow d$	$[\![k]\!](d)$
$k \rightarrow [\![s_1, \ldots, s_n]\!]$	$[\![k]\!]([\![s_1]\!]), \ldots, [\![k]\!]([\![s_n]\!])$

So for instance if the string 'danger_close' indexes the value 1 in our belief base dictionary then we interpret that as meaning the agent believes danger is close, $\mathcal{B}(danger\_close)$. If the string 'task' indexes a list of strings ['t1', 't2', 't3']. Then this is interpreted as meaning the agent believes t1, t2 and t3 are tasks – $\mathcal{B}(task(t1)), \mathcal{B}(task(t2))$ and $\mathcal{B}(task(t3))$.

Where the value of the belief is 0/1 then the belief is interpreted a logical constant during reasoning. So the value 1, indexed by the string " obstacle_close " is considered to be the logical constant, $obstacle\_close$ Where the value of the belief is some list, then the belief is considered to be a set of predicates with the list members as parameters. So the value [' john ',' jane ',' emily '] indexed by the string "name" is considered to be the set of logical predicates $name('john'), name('jane'), name('emily')$.

Where the value of the belief is some Python built-in type it is interpreted to be a logical predicate with the value as a parameter. So the value 0.5, indexed by the string " distance " is considered to be the logical predicate $distance(0.5)$.