April 21,2008

Translating Cognitive Radio (Policy) Language to the Universal Policy Logic
Version 0.1

ICS-16763-TR-08-01
SRI Project No. 16763
Contract No. FA8750-05-C-0230

Prepared by
Daniel Elenius
SRI International
333 Ravenswood Ave
Menlo Park, CA 94025-3493

Prepared for
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

Acknowledgment and Disclaimer:
This material is based upon work supported by the Defense Advanced Projects Agency and the United State Air Force under Contract Number FA8750-05-C-0230.
Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Defense Advanced Research Projects Agency and the United States Air Force.

1 Introduction

This document describes how to translate from Cognitive Radio (Policy) Language (CoRaL) to the Universal Policy Logic (UPL).

2 Translating CoRaL to UPL

A UPL *policy* (Γ, Δ) consists of a well-typed context Γ and a set Δ of well-typed sentences over Γ. Intuitively, Γ contains the *type* information of the policy, and Δ the rules.

We treat a CoRaL Knowledge Base (KB) K as a set of statements. Each statement is a *declaration* and/or an *axiom* (only `defconst` is both). We let K_d be the set of declarations in K and K_a the set of axioms in K.

The translation of K is defined as

$$\begin{aligned}\mathcal{T}(K) &= (\mathcal{T}_\Gamma(K), \mathcal{T}_\Delta(K)), \text{ where} \\ \mathcal{T}_\Gamma(K) &= \bigcup\{\mathcal{T}_\gamma(x) | x \in K_d\} \\ \mathcal{T}_\Delta(K) &= \{\mathcal{T}_\delta(x) | x \in K_a\}\end{aligned}$$

The translation functions for statements, $\mathcal{T}_\gamma$ and $\mathcal{T}_\delta$, are defined in Tables 1 and 2, respectively. $\mathcal{T}_\gamma$ is set-valued (but only `deftype` generates more than one statement, the others generate singleton sets). Tables 3–5 define auxiliary functions which are used in the translation.

Conjectures

(1) *Completeness.* If CoRaL KB $K \models_{CoRaL} \phi$ then $\mathcal{T}(K) \models_{UPL} \mathcal{T}(\phi)$.

(2) *Soundness.* If $\mathcal{T}(K) \models_{UPL} \mathcal{T}(\phi)$ then $K \models_{CoRaL} \phi$.

These should be easy to prove given that the semantics on the two sides is essentially the same.

Notes

We assume that all CoRaL predicate names are also UPL propositional variables, and that all CoRaL constants and variables are also UPL ordinary variables.

The translation works on CoRaL KBs, not on separate policy/ontology files. Therefore, all statements in all ontologies loaded into the KB, and the `use` closure of all such policies, will be translated into one UPL policy. An alternative would be to treat used ontologies as separate UPL policies, which are *composed* with the top-level policy, but we chose to avoid this additional complication for the time being.

The CoRaL KB must be well-typed according to the CoRaL type system, defined elsewhere.

Table 1: Translation of declarations.

CoRaL Syntax, Declaration S	Translation, $\mathcal{T}_\gamma(D)$
`type` T	$\mathcal{T}_t(T) : \texttt{Type}$
`deftype` $T_1 = T_2$	$T_1 : \texttt{Type},\ T_1 = \mathcal{T}_t(T_2)$
`const` $c : T$	$c : \mathcal{T}_t(T)$
`defconst` $c : T = M$	$c : \mathcal{T}_t(T)$

Table 2: Translation of rules and equations.

CoRaL Syntax, Rule R	Translation, $\mathcal{T}_\delta(R)$
`defconst` $c : T = M$	$c = \mathcal{T}_m(M)$
F	$\mathcal{T}_f(F)$

Table 3: Translation of formulae.

CoRaL Syntax, Formula F	Translation, $\mathcal{T}_f(F)$
`True`	`True`
`False`	`False`
$P(M_1, \ldots, M_n)$	$P(\mathcal{T}_m(M_1), \ldots, \mathcal{T}_m(M_n))$
$M_1 = M_2$	$\mathcal{T}_m(M_1) \equiv \mathcal{T}_m(M_2)$
$M_1 < M_2$	$\mathcal{T}_m(M_1) < \mathcal{T}_m(M_2)$
$M_1 =< M_2$	$\mathcal{T}_m(M_1) \leq \mathcal{T}_m(M_2)$
$M_1 > M_2$	$\mathcal{T}_m(M_1) > \mathcal{T}_m(M_2)$
$M_1 >= M_2$	$\mathcal{T}_m(M_1) \geq \mathcal{T}_m(M_2)$
M_1 `in` $\{M_2 \ \ldots \ M_3\}$	$\mathcal{T}_m(M_1) \geq \mathcal{T}_m(M_2)$ `and` $\mathcal{T}_m(M_1) \leq \mathcal{T}_m(M_3)$
M_1 `in` M_2	$\mathcal{T}_m(M_1) \in_l \mathcal{T}_m(M_2)$
	if M_2 is a list term
`not` F	`not` $\mathcal{T}_f(F)$
F_1 `and` F_2	$\mathcal{T}_f(F_1)$ `and` $\mathcal{T}_f(F_2)$
F_1 `or` F_2	$\mathcal{T}_f(F_1)$ `or` $\mathcal{T}_f(F_2)$
F_1 `implies` F_2	$\mathcal{T}_f(F_1)$ `implies` $\mathcal{T}_f(F_2)$
F_1 `if` F_2	$\mathcal{T}_f(F_2)$ `implies` $\mathcal{T}_f(F_1)$
`(forall` $v_1 : T_1, \ldots, v_n : T_n$`)` F	$(\forall\ v_1 : \mathcal{T}_t(T_1), \ldots, v_n : \mathcal{T}_t(T_n))\mathcal{T}_f(F)$
`(exists` $v_1 : T_1, \ldots, v_n : T_n$`)` F	$(\exists\ v_1 : \mathcal{T}_t(T_1), \ldots, v_n : \mathcal{T}_t(T_n))\mathcal{T}_f(F)$

Table 4: Translation of terms.

CoRaL Syntax, Term M	Translation, $\mathcal{T}_m(M)$
c	c
i (an integer)	i
f (a float)	`float`(f_r)
	where f_r is the rational corresponding to f
`true`	`true`
`false`	`false`
$M_1 + M_2$	$\mathcal{T}_m(M_1) + \mathcal{T}_m(M_2)$
$M_1 - M_2$	$\mathcal{T}_m(M_1) - \mathcal{T}_m(M_2)$
$M_1 * M_2$	$\mathcal{T}_m(M_1) * \mathcal{T}_m(M_2)$
M_1 / M_2	$\mathcal{T}_m(M_1) / \mathcal{T}_m(M_2)$
$-M$	$-\mathcal{T}_m(M)$
$(M_1, \ldots, M_n)$	$(\mathcal{T}_m(M_1), (\mathcal{T}_m(M_2), \ldots (\mathcal{T}_m(M_{n-1}), \mathcal{T}_m(M_n)) \ldots))$
$[\,]$	$[\,]$
$[M_1, \ldots, M_n]$	$[\mathcal{T}_m(M_1)] \,\|_l \ldots \|_l [\mathcal{T}_m(M_n)]$

Table 5: Translation of types.

CoRaL Syntax, Type T	Translation, $\mathcal{T}_t(T)$
T (an atomic type)	T
`int`	`Data`
`float`	`Data`
`bool`	`Data`
$[T]$	$[\mathcal{T}_t(T)]$
$(T_1, \ldots, T_n)$	$\mathcal{T}_t(T_1) * \ldots * \mathcal{T}_t(T_n)$
$\{T\}$	$\{\mathcal{T}_t(T)\}$
`Pred`	`Prop`
`Pred`$(T_1, \ldots, T_n)$	$\mathcal{T}_t(T_1) * \ldots * \mathcal{T}_t(T_n) \rightarrow$ `Prop`
$T_1, \ldots, T_n \rightarrow T_r$	$\mathcal{T}_t(T_1) * \ldots * \mathcal{T}_t(T_n) \rightarrow \mathcal{T}_t(T_r)$