Compiler for Trajectory Constraints without Grounding

1 Lifted TCORE

Algorithm 1 outlines a lifted variant of TCORE for ground trajectory constraints. Lifted TCORE is based on a lifted variant of the regression operator, i.e., R^L, which is defined in Algorithm 2. R^L differs from the traditional regression operator in its use of the lifted gamma operator Γ^L, which is defined in Algorithm 3.

Algorithm 1 Lifted TCORE

Require: Set of Operators: A, Set of Ground Constraints: $C^{\doteqdot}$.
Ensure: Set of Operators A' that, $\forall c^{\doteqdot} \in C^{\doteqdot}$, A' cannot violate $c^{\doteqdot}$.

$A' \leftarrow \emptyset$
for $a \in A$ **do**
    $a' \leftarrow$ Transformation of a according to TCORE, using R^L instead of R as the regression operator.
end for
return A'

Algorithm 2 Lifted Regression Operator R^L

Require: Formula: ϕ, Action a.
Ensure: Formula: $R^L(\phi, a)$ is satisfied iff ϕ holds after applying a.

for $f \in \phi$ **do**
    $f \leftarrow \Gamma^L_f(a) \vee (f \wedge \neg \Gamma^L_{\neg f}(a))$
end for
return ϕ

Algorithm 3 Lifted Gamma Operator $\Gamma^L_f(a)$

Require: Literal: f, Action a.
Ensure: Formula: $\Gamma^L_f(a)$ is satisfied iff at least one conditional effect of a that brings about f is satisfied.

$\Gamma^L_f(a) = \emptyset$
for $(\forall \tilde{z} : c \rhd e) \in \mathit{eff}(a)$ **do** ▷ $\tilde{z}$ may be empty
    for $e_i \in e : \exists \xi = (\tilde{u}, \tilde{b}) : e_i[\xi] = f[\xi]$ **do** ▷ $\tilde{u}$ (resp. $\tilde{b}$) contains variables (constants) of e_i (f).
        $\tilde{z}^{free} \leftarrow \tilde{z} \setminus \tilde{u}$
        for $(u_i, b_i) \in \xi$ **do** ▷ $u_i \in \tilde{z}$ or u_i is an argument of a.
            if $u_i \in \tilde{z}$ **then**
                $c = c[u_i | b_i]$
            else ▷ u_i is an argument of a.
                $c = c \wedge (u_i = b_i)$
            end if
        end for
        $c = \exists \tilde{z}^{free} c$
        $\Gamma^L_f(a) = \Gamma^L_f(a) \vee c$
    end for
end for
return $\Gamma^L_f(a)$

Given a literal f and an action a, Algorithm 3 computes $\Gamma^L_f(a)$ as follows. First, for each effect $(\forall \tilde{z} : c \rhd e)$ of a and for each literal $e_i \in e$, we deduce whether there is a unification ξ between the variables in e_i, if any, with the constants in f. (I have run some examples where f also contains variables, by adding constraints with quantified variables, but this has not been tested thoroughly.)

If there is such a unification, then we distinguish two cases. For a pair $(u_i, b_i) \in \xi$, if u_i is an local "effect variable", i.e., it appears in $\tilde{z}$, then we add as a disjunct in $\mathit{\Gamma}_f^L(a)$ formula $c[u_i|b_i]$, which is the condition of the effect after having substituted all instances of variable u_i with constant b_i. Otherwise, if u_i is a global "action variable", i.e., it is one of the arguments of a, then we add disjunct $c \wedge (u_i{=}b_i)$ in $\mathit{\Gamma}_f^L(a)$, as u_i needs to be unified with b_i in all of its instances within operator a.

Finally, we have to take into account the variables in $\tilde{z}$ that do not appear in literal e_i. If there is a least one assignment to those variables that makes $c[u_i|b_i]$ or $c \wedge (u_i{=}b_i)$ true (depending on the case), then a will bring about e_i in the next state. For this reason, we quantify the free variables in the new disjunct of $\mathit{\Gamma}_f^L(a)$ using $\exists \tilde{z}^{free}$.