RuleML - Semantic Profile for the Reified Classical Situation Calculus

Adrian Paschke

RuleML Inc., Canada
paschke AT inf.fu-berlin.de

Abstract. This is the description of the Semantic Profile for the **Reified Classical Situation Calculus**. This profile can be used, e.g. in the **Knowledge Representation (KR) dialect** of **Reaction RuleML**.

1 Introduction

A Semantic Profile[1] in RuleML defines the intended semantics of a RuleML knowledge base as needed e.g. for rule interchange and knowledge imports, updates and querying, as well as semantically safe translations from RuleML into platform specific languages for the execution and reasoning.

This document describes the Semantic Profile for the **Reified Classical Situation Calculus**. This profile can be used, e.g. in the **Knowledge Representation (KR) dialect** of **Reaction RuleML 1.0**[2] [1]. The formalization is based on Pinto's axioms [2] which itself originates from Reiter's formalization that appeared in [3].

2 Signature

The signature is a multi-sorted signature Σ defined as a tuple $\langle A^T, F^T, S^T, D^T; \overline{E}, \overline{F}, \overline{S}, \overline{D}, arity, sort, <, do, Poss, holds, S_0, \wedge, \vee, \neg, \rightarrow, \equiv, \forall, \exists\rangle$ with sorts A^T for action **type** (^{T}) symbols, sorts F^T for fluent type symbols, sorts S^T for situation type symbols, and sorts D^T for domain objects; $\overline{A}$ is the non-empty set of action function symbols of sort A^T, called **actions**, $\overline{S}$ is a non-empty set of situation function symbols of sort S^T, called **situations**, $\overline{F}$ is a non-empty set of fluent function symbols of sort F^T, called **fluents**, $\overline{D}$ is the signature of the domain language which includes domain **constant** symbols, **function** symbols, and **predicate** symbols. The function $arity(A_i)$, $arity(F_i)$, $arity(S_i)$, $arity(D_i)$ associates a non-zero natural number with each action A_i, fluent F_i, situation S_i and D_i. The function $sort$ associates with each n-ary action, fluent or situation function symbol a $n+1$-tuple of sorts. That is, if $expr$ is an action, fluent, or situation function of arity n, then $sort(expr)$ is a $n+1$-tuple

[1] `http://wiki.ruleml.org/index.php/Glossary_of_Reaction_RuleML_1.0#gloss-Profile`

[2] `http://reaction.ruleml.org/` and `http://wiki.ruleml.org/index.php/Specification_of_Reaction_RuleML_1.0#Quick_Links`

of sorts $sort(expr) = (T_1, .., T_n, T_{n+1})$ where $(T_1, .., T_n)$ defines the sorts (types) of the domain of $expr$ and T_{n+1} defines the sort of the range of $expr$, where each T_i is some sort in A^T, F^T, or S^T. Similarly, $sort(D)$ gives the sorts (types) D^T of D. Furthermore, the signature includes the relation $<\subseteq S^T \times S^T$, which defines a **history** ordering of situations, the function $\mathbf{do} : A^T \times S^T \rightarrow S^T$, the predicate $\mathbf{holds} \subseteq F^T \times S^T$ the relation $\mathbf{Poss} \subseteq F^T \times S^T$, and the special constant S_0 of sort S^T which is the initial situation. $\wedge, \vee, \rightarrow, \neg \equiv$ are the standard operators for conjunction, disjunction, implication, equivalence (iff), and negation and $\forall, \exists$ are standard universal and existential quantifiers.

In the following, the smaller case letters a (actions), s (situations), f (fluents), d (domain) are used for variables and upper case A,S,F,D are used for constants. φ is a predicate variable over functions of arity 1. Unless stated with an explicit quantifier, free variables are always assumed to be universially quantified. $s_1 \leq s_2$ is an abbreviation for $s_1 < s_2 \vee s_1 = s_2$. Another used abbreviation is $do([a_1, ..., a_n], s)$ for $do(a_n, do(..., do(a_1, s)...))$.

3 Axioms

$$(\forall \varphi).[\varphi(S_0) \wedge (\forall s, a)(\varphi(s) \rightarrow \varphi(do(a, s)))] \rightarrow (\forall s)\varphi(s), \tag{1}$$

$$(\forall a_1, a_2, s_1, s_2).do(a_1, s_1) = do(a_2, s_2) \rightarrow a_1 = a_2, \tag{2}$$

$$(\forall s_1, s_2, a).s_1 < do(a, s_2) \equiv s1 \leq s_2, \tag{3}$$

$$(\forall s_1, s_2).s_1 < s_2) \rightarrow \neg s1 < s_2, \tag{4}$$

$$(\forall s_1, s_2).s_1 < s_2) \rightarrow \neg s1 < s_2, \tag{5}$$

$$s_1 < s_2 \equiv (\exists s_2).(s_2 = do([a_1, a_2, ..., a_k], s_1)), \tag{6}$$

Note: $s_1 < s_2$ is true iff there exists a sequence of actions $a_1, a_2, ..., a_k$ is weaker than Reiter's $\preceq$ in which every action additionally must be possible:

$$\neg s \preceq S_0, \tag{7}$$

$$s_1 \preceq do(a, s_2) \equiv Poss(a, s_2) \wedge s_1 < s_2. \tag{8}$$

4 Propositions

$$(\forall s)S_0 \leq s, \tag{9}$$

$$(\forall s)\neg s \leq s, \tag{10}$$

$$(\forall s)\neg s \leq S_0, \tag{11}$$

$$(\forall s).s \neq S_0 \rightarrow (\exists a, s_2)s = do(a, s_2), \tag{12}$$

$$(\forall s_1, s_2, s_3).s_1 < s_2 \wedge s_2 < s_3 \rightarrow s_1 < s_3, \tag{13}$$

$$(\forall a, s)s < do(a, s), \tag{14}$$

$$(\forall a_1, a_2, s_1, s_2).do(a_1, s_1) < do(a_2, s_2) \rightarrow s_1 < s_2, \tag{15}$$

$$(\forall a, s_1, s_2).do(a, s_1) \leq s_2 \rightarrow s_1 < s_2, \tag{16}$$

$$(\forall a, s_1, s_2).do(a, s_1) = do(a, s_2) \rightarrow s_1 = s_2, \tag{17}$$

$$(\forall a, s_1, s_2).\neg(s_1 < s_2 < do(a, s_1)), \tag{18}$$

$$(\forall s_1, s_2).s_1 \prec s_2 \rightarrow s_1 < s_2, \tag{19}$$

$$\begin{aligned}(\forall f, s_1, s_2).s_1 < s_2 \wedge holds(f, s_1) \wedge \neg holds(f, s_2) \rightarrow \\ (\exists s, a)(s_1 < do(a, s) \leq s_2) \\ \wedge \neg[holds(f, do(a, s)) \equiv holds(f, s)].\end{aligned} \tag{20}$$

5 Mapping of the Reaction RuleML KR dialect into the Reified Situation Calculus Profile

Given an Reaction RuleML 1.0 document *.rrml* modelled in the Knowledge Represenation (KR) dialect of Reaction RuleML (KR Reaction RuleML 1.0) the following translation $\tau(\cdot)$ from elements of $RR1$ to sentences of the situation calculus profile logic is defined:

- for each $< Fluent >$ **expr**ession **fun**ction with n **arg**uments in *.rrml*, $\tau(< Fluent >)$ is a fluent $F_i \in \overline{F}$ of $arity = n$ with the optional attribute value of $@type$ (or the default sort $ruleml : Fluent$) denoting the sort of F_i in F^T,
- for each $< Action >$ **expr**ession **fun**ction with n **arg**uments in *.rrml*, $\tau(< Action >)$ is an action $A_i \in \overline{A}$ of $arity = n$ with the optional attribute value of $@type$ (or the default sort $ruleml : SimpleAction$) denoting the sort of A_i in A^T,
- for each $< Situation >$ **expr**ession **fun**ction with n **arg**uments in *.rrml*, $\tau(< Situation >)$ is a situation $S_i \in \overline{S}$ of $arity = n$ with the optional attribute value of $@type$ (or the default sort $ruleml : History$) denoting the sort of S_i in S^T,
- for each $< Do >$ function in *.rrml* with first **arg**ument an $< Action >$ and second **arg**ument a $< Situation >$, $\tau(< Do >)$ is a 2-ary function do,
- for each predicate $< Atom >$ in *.rrml* with **rel**ation name $< Rel > poss < /Rel >$ and first **arg**ument an $< Action >$ and second **arg**ument a $< Situation >$, $\tau(< Atom >)$ is a 2-ary predicate $Poss$,
- for each predicate $< Holds >$ in *.rrml* with first **arg**ument an $< Fluent >$ and second **arg**ument a $< Situation >$, $\tau(< Holds >)$ is a 2-ary predicate $holds$,
- for each individual $< Ind > s0 < /Ind >$ in *.rrml*, $\tau(< Ind > s0 < /Ind >)$ is the situation constant S_0,
- for all other (Reaction) RuleML elements t and r in *.rrml*, $\tau(t)$ and $\tau(r)$ is defined inductively as follows:
  - $\tau(t) = \phi_t$, where t is a RuleML **term** and ϕ_t is its first-order logic translation into a logic term (constans, functions, variables),

- $\tau(r) = \phi_r$, where r is a RuleML **formula** and ϕ_r is its first-order logic translation into a logic formula,
- $\tau(\neg r) = \neg\phi_r$,
- $\tau(r_1 \circ r_2) = \tau(r_1) \circ \tau(r_2)$, $\circ \in \wedge, \vee, \neg, \rightarrow$, where r_1 and r_2 are formulas,

with $\tau(\cdot)$ maintaining the same arity and quantification and the optional value of the attribute @*type* (or a default sort from the RuleML metamodel) denoting the sort of $\cdot$ in D^T.

6 Profile Usage in Reaction RuleML 1.0

The profile can be defined as intended semantics by the < **Profile** > element in the < **evaluation** > role.

- the **@type** attribute defines the name/type of the used semantics profile
- the **@iri** attribute points to IRI of the semantics profile

The profile IRI is:

```
http://reaction.ruleml.org/1.0/profiles/ReifiedClassicalSituationCalculusProfile.rrml
```

The profile type defined in the RuleML vocabulary is:

ReifiedClassicalSituationCalculus

This profile specifies the semantics for the reified situation calculus in terms of a RuleML formalization of the axioms and propositions given in the section 3 and 4. It can be imported by XML Inclusions (**XInclude**) or the Reaction RuleML < **Consult** > into a KR Reaction RuleML Reified Situation Calculus program.

7 Example

This example shows how to use the profile be defining it as a profile type from the RuleML vocabulary.

```
<evaluation>
  <Profile type="ruleml:ReifiedClassicalSituationCalculus"/>
</evaluation>
```

This example shows how to use the profile be referencing it's IRI.

```
<evaluation>
  <Profile iri="http://reaction.ruleml.org/1.0/profiles/ReifiedClassicalSituationCalculusProfile.rrml"/>
</evaluation>
```

This example show how to include the profile with all its axioms and rules into a Reaction RuleML document.

```
<!-- include the axioms from profile "SituationCalculusProfile.rrml" -->
<xi:include href="http://reaction.ruleml.org/1.0/profiles/ReifiedClassicalSituationCalculusProfile.rrml"
xpointer="xpointer(/RuleML/*)"/>

<!-- use the included "ruleml:ReifiedClassicalSituationCalculus" semantic profile for interpretation -->
<evaluation>
   <Profile keyref="ruleml:ReifiedClassicalSituationCalculus" direction="backward"/>
</evaluation>
```

Further examples can be found in the Reaction RuleML 1.0 examples directory:
`http://reaction.ruleml.org/1.0/exa/kr/`

8 Acknowledgement

Many thanks to Mikhail Soutchanski for his proof reading and input on this profile.

References

1. Adrian Paschke, Harold Boley, Zhili Zhao, Kia Teymourian, and Tara Athan. Reaction ruleml 1.0: Standardized semantic reaction rules. In *RuleML*, pages 100–119, 2012.
2. Javier Andres Pinto. *Temporal Reasoning in the Situation Calculus.* PhD thesis, Toronto, Ont., Canada, Canada, 1994. AAINN92616.
3. Raymond Reiter. The frame problem in situation the calculus: A simple solution (sometimes) and a completeness result for goal regression. pages 359–380. Academic Press Professional, Inc., San Diego, CA, USA, 1991.