Maude natural4 internal documentation

Joe Watt

March 15, 2023

This is a work in progress.

Contents

1 Syntax

1.1 Grammar reference

Here we describe the AST used in Maude. We use the metavariables s and n to refer to strings and natural numbers respectively.

```
                 ⟨RuleSet⟩ → [{⟨Rule⟩ “,”} Rule]
                    ⟨Rule⟩ → “START” ⟨Identifier⟩
                           | ⟨Regulative⟩
                           | ⟨Identifier⟩ “MEANS” ⟨Identifiers⟩
              ⟨Identifier⟩ → “qid (‘‘s’’)”
             ⟨Identifiers⟩ → {⟨Identifier⟩ “AND”} ⟨Identifiers⟩
              ⟨Regulative⟩ → “RULE” ⟨Identifier⟩ ⟨ActorDeonticActionDeadline⟩ ⟨HenceLest⟩
⟨ActorDeonticActionDeadline⟩ → ⟨RKeyword⟩ ⟨Aka⟩ ⟨Deontic⟩ ⟨Deadline⟩ ⟨Action⟩
                           | ⟨RKeyword⟩ ⟨Aka⟩ ⟨Deadline⟩ ⟨Deontic⟩ ⟨Action⟩
                           | ⟨Deadline⟩ ⟨RKeyword⟩ ⟨Aka⟩ ⟨Deontic⟩ ⟨Action⟩
                           | ⟨RKeyword⟩ ⟨Aka⟩ ⟨Deontic⟩ ⟨Action⟩ [⟨Deadline⟩]
                ⟨RKeyword⟩ → “EVERY” | “PARTY”
                     ⟨Aka⟩ → ⟨Identifier⟩ [“AKA” ⟨Identifier⟩]
                 ⟨Deontic⟩ → “MUST” | “MAY” | “SHANT”
                  ⟨Action⟩ → [“DO”] ⟨Identifier⟩
                ⟨Deadline⟩ → (“ON” | “WITHIN”) n “DAY”
               ⟨HenceLest⟩ → [“HENCE” ⟨MaybeIdentifiers⟩] [“LEST” ⟨MaybeIdentifiers⟩]
                           | [“LEST” ⟨MaybeIdentifiers⟩] [“HENCE” ⟨MaybeIdentifiers⟩]
         ⟨MaybeIdentifier⟩ → “NOTHING” | ⟨Identifier⟩
        ⟨MaybeIdentifiers⟩ → {⟨MaybeIdentifier⟩ “AND”} ⟨MaybeIdentifier⟩
```

1.1.1 Some notes about the syntax

- Each nonterminal on the LHS is modelled by an algebraic datatype in Maude, with constructors modelling the RHS of the production rules. In other words, we have designed our abstract syntax to look very much like the concrete syntax, which is in turn a form of Controlled Natural Language (CNL).

- `qid` is a Maude function meaning “quoted identifier”, which takes a string and turns it into an identifier.

- There is no explicit syntax for `HENCE/LEST FULFILLED/BREACHED` as those are inferred automatically from the deontic in the rule. Missing `HENCE/LEST` clauses are interpreted as `HENCE/LEST NOTHING`, which are in turn interpreted as `HENCE/LEST Fulfilled/Breached` depending on the deontic used in the rule, eg:

```
  PARTY actor MAY action
= PARTY actor MAY action HENCE NOTHING LEST NOTHING
= PARTY actor MAY action HENCE FULFILLED LEST FULFILLED
  PARTY actor MUST action
= PARTY actor MUST action HENCE NOTHING LEST NOTHING
= PARTY actor MUST action HENCE FULFILLED LEST BREACHED
  PARTY actor SHANT action
= PARTY actor SHANT action HENCE NOTHING LEST NOTHING
= PARTY actor SHANT action HENCE BREACHED LEST FULFILLED
```

1.2 PDPA example

Here is an example of our AST in Maude.

```
START qid("Assessment"),

RULE qid("Assessment")
EVERY qid("Organisation") AKA qid("You")
MUST WITHIN 30 DAY
DO qid("assess if it is a Notifiable Data Breach")
HENCE qid("Notification")
LEST qid("PDPC query with demand"),

RULE qid("PDPC query with demand")
PARTY qid("the PDPC")
MAY qid("query You")
HENCE qid("Respond to PDPC"),

RULE qid("Respond to PDPC")
PARTY qid("You")
MUST DO qid("respond to the PDPC"),

qid("Notification") MEANS (qid("Notify PDPC") AND qid("Notify Individuals")),

RULE qid("Notify PDPC")
PARTY qid("You")
MUST qid("NOTIFY the PDPC")
WITHIN 3 DAY
HENCE qid("PDPC prohibit notify individuals"),

RULE qid("PDPC prohibit notify individuals")
PARTY qid("the PDPC")
MAY qid("NOTIFY you")
HENCE qid("Cannot notify individuals"),
RULE qid("Cannot notify individuals")
PARTY qid("You")
SHANT qid("NOTIFY each of the Notifiable Individuals"),

RULE qid("Notify Individuals")
WITHIN 3 DAY
PARTY qid("You")
MUST qid("NOTIFY each of the Notifiable Individuals")
LEST qid("Notify and explain"),

RULE qid("Notify and explain")
PARTY qid("You")
MUST qid("notify each of the Notifiable Individuals")
```