1 Modeling

Formal specification of a buffer with an infinite number of states.

1.1 Rigid data types

Example 1. *Specification of lists of arbitrary elements:*

```
spec! LISTS
pr BOOL
sorts Elt List .
op err : -> Elt .
op empty : -> List .
op _⨾_ : Elt List -> List .
op _in_ : Elt List -> Bool .
```

eq-1 $\forall \mathsf{L} \cdot \mathsf{L} \,⨾\, \mathtt{err} = \mathsf{L}$
eq-2 $\forall \mathsf{E} \cdot \mathsf{E}\ \mathtt{in}\ \mathtt{empty} = \mathtt{false}$
eq-3 $\forall \mathsf{E}, \mathsf{E}' \cdot (\mathsf{E}\ \mathtt{in}\ \mathsf{E}' \,⨾\, \mathsf{L})\mathtt{if}\ \ \mathsf{E} = \mathsf{E}'$
eq-4 $\forall \mathsf{E}, \mathsf{E}' \cdot \mathsf{E}\ \mathtt{in}\ \mathsf{E}' \,⨾\, \mathsf{L} = \mathsf{E}\ \mathtt{in}\ \mathsf{L}\ \mathtt{if} \neg(\mathsf{E} = \mathsf{E}')$

1.2 Nominals

Example 2. *Specification of nominals:*

```
spec! NOMINAL
sort Nominal .
op init : -> Nominal .
op next : Nominal -> Nominal .
```

1.3 Flexible data types

Example 3. *Specification of the attributes* `read` *and* `del`*:*

```
spec BUFFER[LISTS,NOMINAL]
op read : List -> Elt .
op del : List -> List .
```

eq-5 $\forall \mathsf{Z} \cdot @_{\mathsf{Z}}\, \mathtt{read}(\mathtt{empty}) = \mathtt{err}$
eq-6 $\forall \mathsf{Z} \cdot @_{\mathsf{Z}}\, \mathtt{del}(\mathtt{empty}) = \mathtt{empty}$
eq-7 $\forall \mathsf{E}, \mathsf{L} \cdot @_{\mathtt{init}}\, \mathtt{read}(\mathsf{E} \,⨾\, \mathsf{L}) = \mathsf{E}$
eq-8 $\forall \mathsf{E}, \mathsf{L} \cdot @_{\mathtt{init}}\, \mathtt{del}(\mathsf{E} \,⨾\, \mathsf{L}) = \mathsf{L}$
eq-9 $\forall \mathsf{Z}, \mathsf{E}, \mathsf{L} \cdot @_{\mathtt{next(Z)}}\, \mathtt{read}(\mathsf{E} \,⨾\, \mathsf{L}) = @_{\mathsf{Z}}\, \mathtt{read}(\mathsf{L})$
eq-10 $\forall \mathsf{Z}, \mathsf{E}, \mathsf{L} \cdot @_{\mathtt{next(Z)}}\, \mathtt{del}(\mathsf{E} \,⨾\, \mathsf{L}) = \mathsf{E}; \mathtt{del}(\mathsf{Z}, \mathsf{L})$

2 Formal verification

The property we are interested in proving formally is $\Gamma \vdash \forall \mathsf{L}, \mathsf{E} \cdot \exists \mathsf{Z} \cdot \mathtt{read}(\mathsf{Z}, \mathsf{L}) = \mathsf{E}\ \mathtt{if}(\mathsf{E}\ \mathtt{in}\ \mathsf{L}) = \mathtt{true}$, where Γ consists of all equations in BUFFER We proceed by induction on the structure of L.

- induction base: $\Gamma \vdash \forall \mathsf{E} \cdot \exists \mathsf{Z} \cdot \mathtt{read}(\mathsf{Z}, \mathtt{empty}) = \mathsf{E}\ \mathtt{if}(\mathsf{E}\ \mathtt{in}\ \mathtt{empty}) = \mathtt{true}$
- induction step: