1 Modeling

We present the specification of a mutual exclusion protocol.

1.1 Rigid data types

Example 1 (Labels).
```
spec! LABEL
sort Label .
ops re wt cs : -> Label [ctor].
op _~_ : Label Label -> Bool [comm].
var  L : Label .
eq (re ~ wt) = false .
eq (re ~ cs) = false .
eq (wt ~ cs) = false .
---
ceq true = false if re = wt .
ceq true = false if re = cs .
ceq true = false if wt = cs .
```

Example 2 (Process identifiers).
```
spec* PID
inc BOOL .
sort Pid .
op _~_ : Pid Pid -> Bool [comm].
vars I J : Pid .
eq I ~ I = true .
ceq I = J if I ~ J [nonexec].
```

Example 3 (lists of process identifiers).
```
spec! SEQUENCE{X :: PID}
sort Sequence .
subsorts X$Pid < Sequence .
--- constructors
op empty : -> Sequence [ctor] .
op _,_ : Sequence Sequence -> Sequence [ctor id: empty assoc].
vars Q Q' : Sequence . var I : X$Pid .
---
op top : Sequence -> X$Pid .
eq top(empty) = empty .
eq top(I,Q) = I .
---
op get : Sequence -> Sequence .
eq get(empty) = empty .
eq get(I,Q) = Q .
---
ceq true = false if Q,I,Q' := empty .
ceq [lemma-top]: top(Q,I) = top(Q) if top(Q) :: X$Pid .
```

1.2 Nominals

Example 4 (Agents).
```
spec* AGENT
sort Agent
```

Example 5 (Nominals).
```
spec! NOMINAL{Y :: AGENT}
sorts Sys.
--- actions
op init : -> Sys [ctor].
ops want try exit : Sys Y$Agent -> Sys [ctor].
```

1.3 Flexible data types

Example 6 (Mutual exclusion protocol).

```
spec* QLOCK{X ::  PID, Y ::  AGENT}

pr SEQUENCE{X} .  pr NOMINAL{Y} .  pr LABEL .

--- observers

op pid :→ X$Pid --- extract pid from agents

op sq :→ Sequence --- gives the waiting queue for each state

op pc :→ Label --- indicates the label of each agent at a given state

--- variables

vars S S₁ S₂ : Sys

vars I J K : X$Pid

vars A B C : Y$Agent

var Q : Sequence

  --- restrictions ---
```

(1) $\forall A, S_1, S_2 \cdot @_{S_1} @_A \, pid = @_{S_2} @_A \, pid$ --- pid depends only of the agent

(2) $\forall A, B, S \cdot @_S @_A \, sq = @_S @_B \, sq$ --- sq depends only of the current state

     --- init ---

(3) $\forall A \cdot @_{init} @_A \, pc = re$

(4) $@_{init} \, sq = empty$

     --- want ---

(5) $\forall S, A, B \cdot @_{want(S,A)} @_B \, pc = wt$ if $@_S @_A \, pc = re \; \bigwedge \; A = B$

(6) $\forall S, A, B \cdot @_{want(S,A)} @_B \, pc = @_S @_B \, pc$ if $\neg(A = B)$

(7) $\forall S, A, B \cdot @_{want(S,A)} @_B \, pc = @_S @_B \, pc$ if $\neg(@_A @_S \, pc = re)$

(8) $\forall S, A \cdot @_{want(S,A)} \, sq = (@_S \, sq), (@_A \, pid)$ if $@_S @_A \, pc = re$

(9) $\forall S, A \cdot @_{want(S,A)} \, sq = @_S \, sq$ if $\neg(@_S @_A \, pc = re)$

     --- try ---

(10) $\forall S, A, B \cdot @_{try(S,A)} @_B \, pc = cs$ if $@_S @_A \, pc = wt \; \bigwedge \; (@_A \, pid), Q := @_S \, sq \; \bigwedge \; A = B$

(11) $\forall S, A, B \cdot @_{try(S,A)} @_B \, pc = @_S @_B \, pc$ if $\neg(A = B)$

(12) $\forall S, A, B \cdot @_{\text{try}(S,A)} @_B \, pc = @_S \, @_B \, pc$ if $\neg(@_S \, @_A \, pc = wt)$

(13) $\forall S, A, B \cdot @_B \, @_{\text{try}(S,A)} \, pc = @_B \, @_S \, pc$ if $\neg(\text{top}(@_S \, sq) = @_A \, pid)$

(14) $\forall S, A \cdot @_{\text{try}(S,A)} \, sq = @_S \, sq$

    `--- exit ---`

(15) $\forall S, A, B \cdot @_{\text{exit}(S,A)} @_B \, pc = re$ if $@_S \, @_A \, pc = cs \ \bigwedge \ A = B$

(16) $\forall S, A, B \cdot @_{\text{exit}(S,A)} @_B \, pc = @_A \, @_B \, pc$ if $\neg(A = B)$

(17) $\forall S, A, B \cdot @_{\text{exit}(S,A)} @_B \, pc = @_A \, @_B \, pc$ if $\neg(@_S \, @_A \, pc = cs)$

(18) $\forall S, A \cdot @_{\text{exit}(S,A)} \, sq = \text{get}(@_S \, sq)$ if $@_S \, @_A \, pc = cs$

(19) $\forall S, A \cdot @_{\text{exit}(S,A)} \, sq = @_S \, sq$ if $\neg(@_S \, @_A \, pc = cs)$

2 Formal verification

We are interested in proving formally $\text{QLOCK} \vdash \forall S, A \cdot \text{top}(@_S \, sq) = @_A \, pid$ if $@_S \, @_A \, pc = cs$.

  `==================================================`

  `spec QLOCK`$_I$

  `pr QLOCK .`

  `op s : -> Sys .`

(20) $\forall A \cdot \text{top}(@_s \, sq) = @_A \, pid$ if $@_s \, @_A \, pc = cs$ [induction hypothesis].

  `==================================================`

  `spec QLOCK`$_{TC}$

  `pr QLOCK`$_I$ `.`

  `op a b : -> Y$Agent .`

  `==================================================`

Apply induction on S:

[init] In this case,

1	$\text{QLOCK} \vdash \forall A \cdot \text{top}(@_{\text{init}} \, sq) = @_A \, pid$ if $@_{\text{init}} \, @_A \, pc = cs$	
2	$\text{QLOCK} \vdash \forall A \cdot \text{top}(@_{\text{init}} \, sq) = @_A \, pid$ if $re = cs$	by sentence (3)
3	discharged	since $\text{QLOCK} \vdash true = false$ if $re = cs$

[want]

1	$\text{QLOCK}_I \vdash \forall A, B \cdot \text{top}(@_{\text{want}(s,B)} \, sq) = @_A \, pid$ if $@_{\text{want}(s,B)} \, @_A \, pc = cs$	
2	$\text{QLOCK}_{TC} \vdash \text{top}(@_{\text{want}(s,b)} \, sq) = @_a \, pid$ if $@_{\text{want}(s,b)} \, @_a \, pc = cs$	

    [$b = a$, $@_s \, @_b \, pc = re$]

1	$\text{QLOCK}_{TC} + \{b = a, @_s \, @_b \, pc = re\} \vdash$ $\text{top}(@_{\text{want}(s,b)} \, sq) = @_a \, pid$ if $@_{\text{want}(s,b)} \, @_a \, pc = cs$	by case analysis
2	$\text{QLOCK}_{TC} + \{a = b, @_s \, @_b \, pc = re\} \vdash$ $\text{top}(@_s \, sq, @_b \, pid) = @_b \, pid$ if $wt = cs$	by rew
3	discharged	since $\text{QLOCK} \vdash true = false$ if $wt = cs$

[¬(b = a)]

1	$\text{QLOCK}_{TC} + \{\texttt{b} \sim \texttt{a} = \texttt{false}\} \vdash$	by case analysis
	$\text{top}(@_{\texttt{want(s,b)}} \,\texttt{sq}) = @_\texttt{a}\,\texttt{pid if } @_{\texttt{want(s,b)}}\,@_\texttt{a}\,\texttt{pc} = \texttt{cs}$	
2	$\text{QLOCK}_{TC} + \{\texttt{a} \sim \texttt{b} = \texttt{false}\} \vdash$	by rew
	$\text{top}(@_\texttt{s}\,\texttt{sq}) = @_\texttt{a}\,\texttt{pid if } @_\texttt{s}\,@_\texttt{a}\,\texttt{pc} = \texttt{cs}$	
3	discharged	by the induction hypothesis

[¬(@ₛ @ᵦ pc) = re]

1	$\text{QLOCK}_{TC} + \{@_\texttt{s}\,@_\texttt{b}\,\texttt{pc} \sim \texttt{re} = \texttt{false}\} \vdash$	by case analysis
	$\text{top}(@_{\texttt{want(s,b)}} \,\texttt{sq}) = @_\texttt{a}\,\texttt{pid if } @_{\texttt{want(s,b)}}\,@_\texttt{a}\,\texttt{pc} = \texttt{cs}$	
2	$\text{QLOCK}_{TC} + \{@_\texttt{s}\,@_\texttt{b}\,\texttt{pc} \sim \texttt{re} = \texttt{false}\} \vdash$	by rew
	$\text{top}(@_\texttt{s}\,\texttt{sq}) = @_\texttt{a}\,\texttt{pid if } @_\texttt{s}\,@_\texttt{a}\,\texttt{pc} = \texttt{cs}$	
3	discharged	by the induction hypothesis

[try]

[exit]