BCS2213 – Formal methods

Teaching assignment 3. TLA specification of Asynchronous Interface.

1. Run TLA+ Toolbox.

2. Develop TLA specification of AsynchInterface, as shown below (please use as module name Lab_3_<Your_ID>)

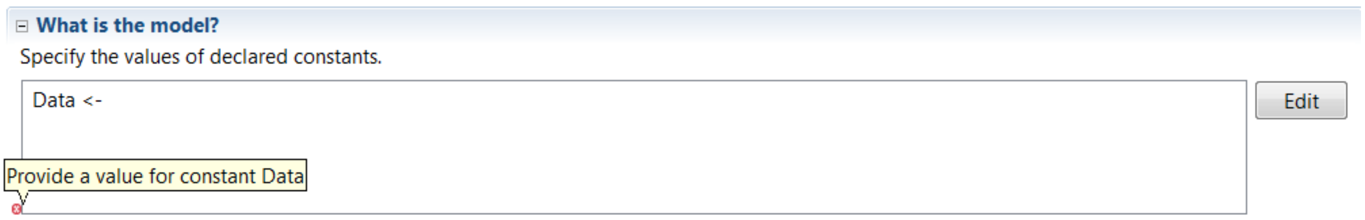
MODULE *AsynchInterface*

$$\begin{array}{l}
\textsc{extends}\ \mathit{Naturals} \\
\textsc{constant}\ \mathit{Data} \\
\textsc{variables}\ \mathit{val}, \mathit{rdy}, \mathit{ack} \\
\mathit{TypeInvariant} \triangleq \begin{array}[t]{l} \land\ \mathit{val} \in \mathit{Data} \\ \land\ \mathit{rdy} \in \{0, 1\} \\ \land\ \mathit{ack} \in \{0, 1\} \end{array}
\end{array}$$

$$\begin{array}{lcl}
\mathit{Init} & \triangleq & \land\ \mathit{val} \in \mathit{Data} \\
 & & \land\ \mathit{rdy} \in \{0, 1\} \\
 & & \land\ \mathit{ack} = \mathit{rdy} \\
\mathit{Send} & \triangleq & \land\ \mathit{rdy} = \mathit{ack} \\
 & & \land\ \mathit{val}' \in \mathit{Data} \\
 & & \land\ \mathit{rdy}' = 1 - \mathit{rdy} \\
 & & \land\ \textsc{unchanged}\ \mathit{ack} \\
\mathit{Rcv} & \triangleq & \land\ \mathit{rdy} \neq \mathit{ack} \\
 & & \land\ \mathit{ack}' = 1 - \mathit{ack} \\
 & & \land\ \textsc{unchanged}\ \langle \mathit{val}, \mathit{rdy} \rangle \\
\mathit{Next} & \triangleq & \mathit{Send} \lor \mathit{Rcv} \\
\mathit{Spec} & \triangleq & \mathit{Init} \land \Box[\mathit{Next}]_{\langle \mathit{val}, \mathit{rdy}, \mathit{ack} \rangle}
\end{array}$$

$$\textsc{theorem}\ \mathit{Spec} \Rightarrow \Box \mathit{TypeInvariant}$$

3. Create new model. To run the model you need provide a value for the **CONSTANT Data**. Find in the Model Overview the next window and press "Edit" button.

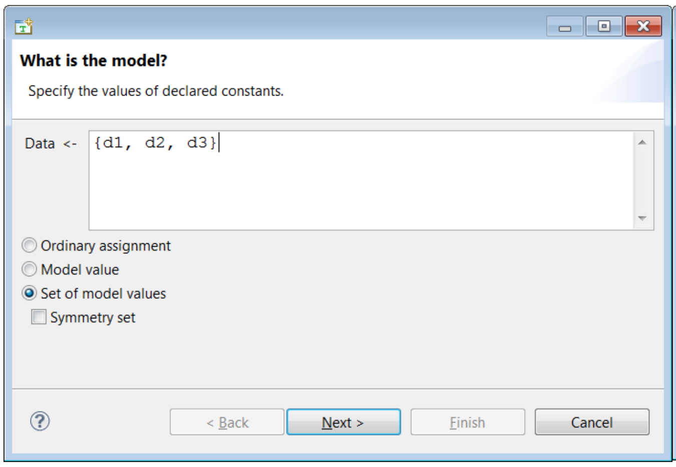

What is the model?

Specify the values of declared constants.

Data <-

Provide a value for constant Data

- Specify the values of constant Data as {d1, d2, d3} and press "Next >"
- Leave values untyped and press "Finish".

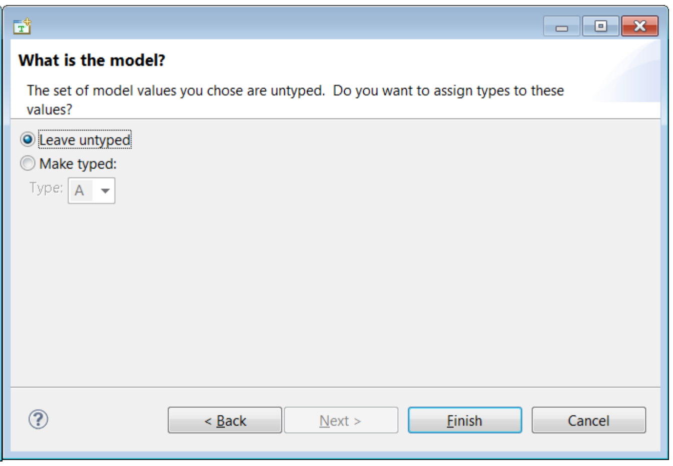


What is the model?

Specify the values of declared constants.

```
Data <- {d1, d2, d3}
```

Ordinary assignment
Model value
[✓ Set of model values]
Symmetry set

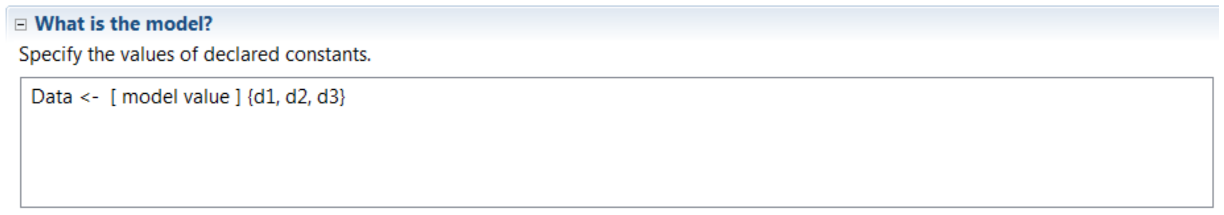


What is the model?

The set of model values you chose are untyped. Do you want to assign types to these values?

[✓ Leave untyped]
Make typed:
Type: A

As result you will see



What is the model?

Specify the values of declared constants.

Data <- [model value] {d1, d2, d3}

where {d1, d2, d3} is a set of messages to be send via asynchronous channel.

4. Define **Send** action with parameter d, showing exact data to be sent by **Send(d)**

Modify the **Next** action, showing that exists *d* in *Data*, such that the step satisfies **Send(d)**

5. Analyze the amount of *distinct* states, generated by TLC.

For it change the size of the Data set (add/delete one element), run TLC and see results.

Comment your finings inside module.

6. What happens if you will specify ***AND*** operation in the next state predicate, linking ***Send*** and ***Rcv*** actions?

Is such the behavior correct? Check it by TLC. Write corresponding comment in your code.

7. TLC allows printing values during module checking.

Operator **Print** is defined in the standard module **TLC**, you need include it by the **EXTENDS** keyword.

TLA definition of Print is

Print(exp1, exp2) == exp2

i.e. the return value of Print(exp1, exp2) is the expression exp2

To use Print in formulas as ***true*** *assumption* we can specify

Print(exp, TRUE)

To print more, than one expression we can use tuple

Print(<<id, exp>>, TRUE)

Modify Send and Rcv actions by adding corresponding print statements:

```
/\ Print(<<"Send ", val>>, TRUE)
```

```
/\ Print(<<"Rcv ", val>>, TRUE)
```

Analyze the printed output.

Copy part of output into your lab works and give explanations.

8. Simplify the AsyncInterface protocol in order it has *only one synchronization line* – **ack** (please do it on the base of the initial specification by commenting not needed parts).

Conditions:

Sender sends a **val** if **ack** is 0 (and next set **ack** into 1 - “work done”).

Receiver receives a **val**, if **ack** is 1 (and next set **ack** into 0 - “new request for send”).

Note. **rdy** variable no longer needed. Please comment it.

9. Submit the TLA specification into Kalam for evaluation. Name the file Lab_3_<Your ID>.tla

This assignment will be evaluated in maximum 2.5% of the total marks.