Leverage the power of the First Order... Logic: Introduction to TLA+

Thomas Bracher

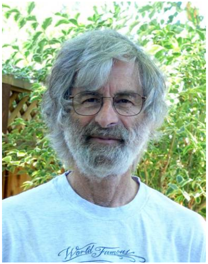
March 11, 2021

Different Paradigms, same goal

- C – procedural
- Java – object oriented
- Haskell – functional
- Built for code execution

Enters TLA+

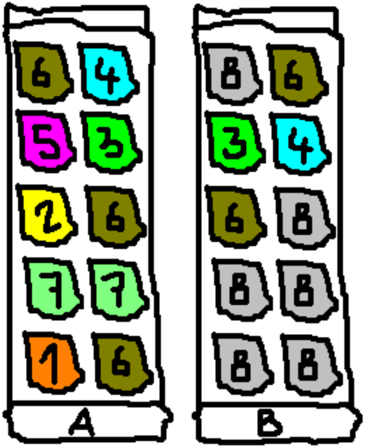
Specification language by Leslie Lamport



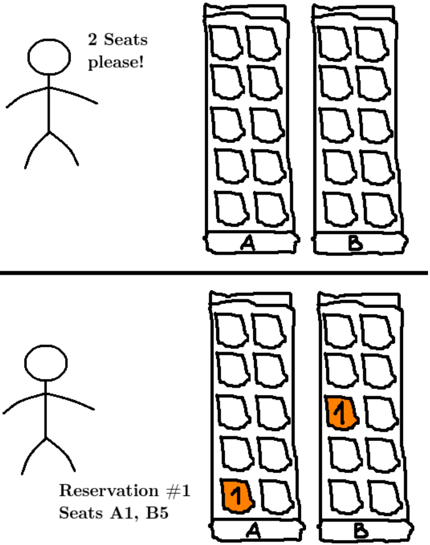
TLA+: Temporal Logic Action

- Temporal = intuitive time
- Logical = first order logic
- Action = why not?

Reservation Train Kata

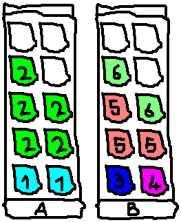


The Rules



The Rules

Max 70% occupation



Enters the First Order...

First Order Logic

$$
\begin{aligned}
&\mathit{Coaches} \triangleq \{''A'', ''B''\} \\
&\mathit{SeatNumbers} \triangleq \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\
&\mathit{Seats} \triangleq \mathit{Coaches} \times \mathit{SeatNumbers}
\end{aligned}
$$

First Order Logic

$$
\begin{aligned}
&Coaches \triangleq \{''A'', ''B''\}\\
&SeatNumbers \triangleq \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}\\
&Seats \triangleq \{\langle ''A'', 1\rangle, \langle ''B'', 1\rangle,\\
&\langle ''A'', 2\rangle, \langle ''B'', 2\rangle, \langle ''A'', 3\rangle, \langle ''B'', 3\rangle,\\
&\langle ''A'', 4\rangle, \langle ''B'', 4\rangle, \langle ''A'', 5\rangle, \langle ''B'', 5\rangle,\\
&\langle ''A'', 6\rangle, \langle ''B'', 6\rangle, \langle ''A'', 7\rangle, \langle ''B'', 7\rangle,\\
&\langle ''A'', 8\rangle, \langle ''B'', 8\rangle, \langle ''A'', 9\rangle, \langle ''B'', 9\rangle,\\
&\langle ''A'', 10\rangle, \langle ''B'', 10\rangle\}
\end{aligned}
$$

First Order... Logic

$$Predicate \triangleq i \in \{1, 2, 3, 4\}$$

First Order Logic

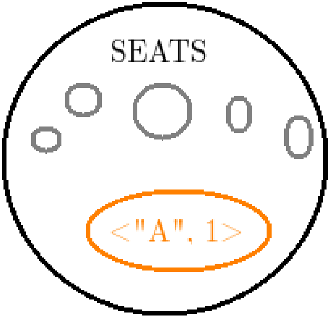
$$Implies \triangleq i \in \{1, 2\} \Rightarrow i \in \{1, 2, 3\}$$

First Order Logic

$$ConjuctionOp(seat) \triangleq seat[1] \in \{"A", "B"\} \land seat[2] \in 1..10$$

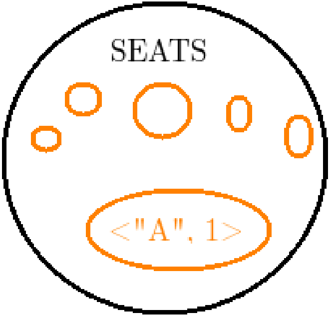
First Order Logic

$$Existence \triangleq \exists\ seat \in Seats : seat = \langle "A", 1\rangle$$



First Order Logic

$$Universal \triangleq \forall\ seat \in Seats : ConjuctionOp(seat)$$

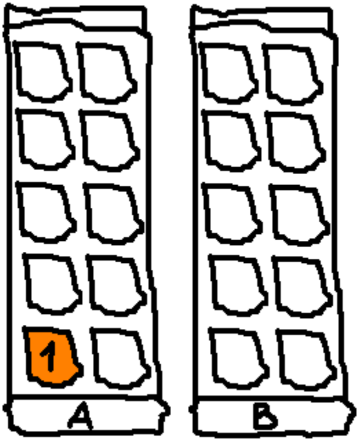


First Specification

First Specification

$$\mathit{Union} \triangleq \{1, 2\} \cup \{3\} = \{1, 2, 3\}$$

First Specification



First Specification

MODULE *FirstSpecification*

EXTENDS *Naturals*

VARIABLE *reservations*

$$
\begin{aligned}
Coaches &\triangleq \{\text{"A"}, \text{"B"}\} \\
SeatNumbers &\triangleq 1 \,..\, 10 \\
Seats &\triangleq Coaches \times SeatNumbers
\end{aligned}
$$

$$Reserve \triangleq reservations' = reservations \cup \{\{\langle \text{"A"}, 1 \rangle\}\}$$

$$
\begin{aligned}
Init &\triangleq reservations = \{\} \\
Next &\triangleq Reserve
\end{aligned}
$$

First Specification

VARIABLE *reservations*

First Specification

$$Reserve \triangleq reservations' = reservations \cup \{\langle "A", 1\rangle\}$$

First Specification

$$Init \triangleq reservations = \{\}$$
$$Next \triangleq Reserve$$

Toolbox

Reserving a seat at a time

$$Reserve \triangleq \exists\, seat \in Seats : reservations' = reservations \cup \{\{seat\}\}$$

Toolbox

Enforcing an invariant

At most 70% of the train is reserved

Enforcing an invariant

$$\mathit{Union} \triangleq \text{UNION } \{\{1, 2, 3\}, \{1, 4\}\} = \{1, 2, 3, 4\}$$

Enforcing an invariant

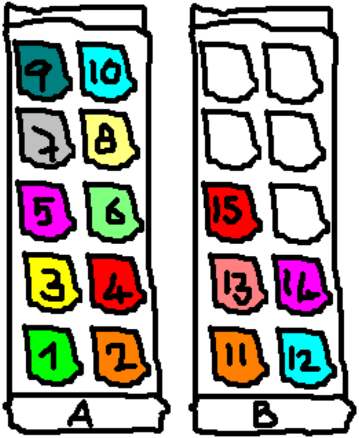
$\mathit{Cardinal} \triangleq \mathit{Cardinality}(\{1, 2, 3\}) = 3$

Enforcing an invariant

$70 \mathit{PercentTrainOccupation} \triangleq (70 * \mathit{Cardinality}(\mathit{Seats})) \div 100$
$\mathit{AtMost}70\mathit{PercentTrainOccupation} \triangleq$
$\mathit{Cardinality}(\mathrm{UNION}\ \mathit{reservations}) \leq 70\mathit{PercentTrainOccupation}$

Toolbox

Counter Example



Inforcing the invariant

$$
\begin{aligned}
&70PercentTrainOccupation \triangleq (70 * Cardinality(Seats)) \div 100\\
&ReservedSeats \triangleq \mathrm{UNION}\ reservations\\
&Reserve \triangleq\\
&\land Cardinality(ReservedSeats) < 70PercentTrainOccupation\\
&\land \exists\ seat \in Seats : reservation' = reservations \cup \{\{seat\}\}
\end{aligned}
$$

Reserving multiple seats

Reserving multiple seats

$Subset \triangleq \text{SUBSET } \{1,2,3\} =$
$\{\{\},\{1\},\{2\},\{3\},\{1,2\},\{2,3\},\{1,3\},\{1,2,3\}\}$

Reserving multiple seats

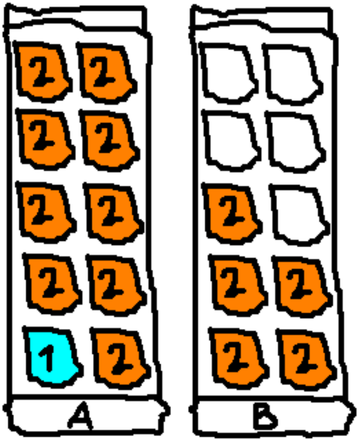
$$
\begin{aligned}
&\mathit{Reserve}(\mathit{count}) \triangleq \\
&\land \mathit{Cardinality}(\mathit{ReservedSeats}) < 70\mathit{PercentTrainOccupation} \\
&\land \exists\ \mathit{seats} \in \mathrm{SUBSET}\ \mathit{Seats} : \\
&\land \mathit{Cardinality}(\mathit{seats}) = \mathit{count} \\
&\land \mathit{reservation}' = \mathit{reservations} \cup \{\mathit{seats}\}
\end{aligned}
$$

Reserving multiple seats

$$Next \triangleq \exists\ seatCount \in 1..Cardinality(Seats) : Reserve(seatCount)$$

Toolbox

Counter Example



Reserving multiple seats

$$
\begin{aligned}
&Reserve(count) \triangleq \\
&\land Cardinality(ReservedSeats) < 70PercentTrainOccupation \\
&\ldots
\end{aligned}
$$

Fixing the specification

$$
\begin{aligned}
&\mathit{Reserve}(\mathit{count}) \triangleq \\
&\wedge \mathit{Cardinality}(\mathit{ReservedSeats}) + \mathit{count} \leq 70\mathit{PercentTrainOccupation} \\
&\wedge \exists\ \mathit{seats} \in \mathrm{SUBSET}\ \mathit{Seats} : \\
&\wedge \mathit{Cardinality}(\mathit{seats}) = \mathit{count} \\
&\wedge \mathit{reservation}' = \mathit{reservations} \cup \{\mathit{seats}\}
\end{aligned}
$$

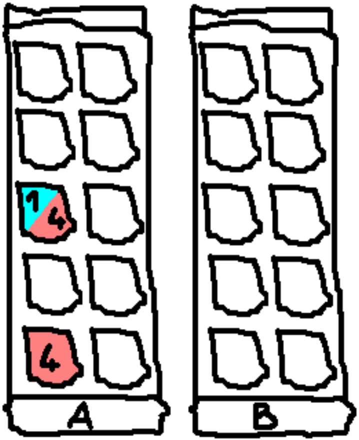
Overlapping reservations

Overlapping reservations

$$
\begin{aligned}
&SeatsReservedOnce \triangleq \\
&\forall\ seat \in Seats : \forall\ r1 \in reservations : \forall\ r2 \in reservations : \\
&(seat \in r1 \land seat \in r2) \Rightarrow r1 = r2
\end{aligned}
$$

Toolbox

Counter Example



Overlapping reservations

$$\mathit{SetDifference} \triangleq \{1, 2, 3\} \setminus \{3, 4\} = \{1, 2\}$$

Toolbox

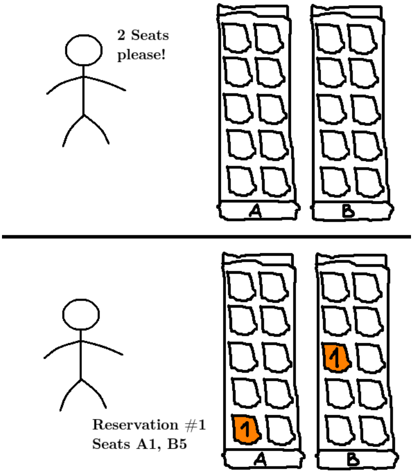
First Order Logic (FOL)

- Intuitive
- Powerful
- Most problems can be expressed with FOL

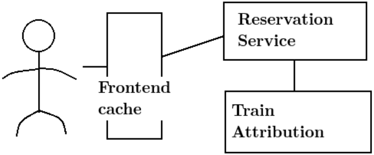
TLA+

- Yields the power of FOL
- Easy incremental modelling
- Built for distributed systems

Single node reservation



Distributed reservation



What's next?

- Download the toolbox
- Play with some tutorials
- Ask your questions to the community
- Read the book
- Have fun!

Thank you