The Dwarf Signal

A Railway Signalling Device

Simon Foster Jim Woodcock
University of York

20th August 2022

Overview

Overview

1. Introduction: a Dwarf Railway Signal
2. Formal Specification of the Dwarf Signal
3. Formalising the Safety Requirements
4. Specifying the Interface
5. A Use Case: Installation then Set to Warning

Dwarf Railway Signals

- Dwarf signals serve the same purpose as typical high signals.
- They are small, short, wayside or trackside signals.
- Commonly found where there is restricted room between the tracks.
- Short enough to fit notch at bottom of normal clearance envelope.
- Low position limits distant visibility: only relatively slow-speed track.
- High-speed, poor clearance: signal bridge is more expensive solution.
- Frequently used for:
  - Transition tracks: sidings, spurs, branch lines, turn-outs, diamonds.
  - Signalling limited clearances, reduced speeds.
  - Alternative to standard high signals to reduce costs and maintenance.

Dwarf Railway Signals

- Dwarf signals serve the same purpose as typical high signals.
- They are small, short, wayside or trackside signals.
- Commonly found where there is restricted room between the tracks.
- Short enough to fit notch at bottom of normal clearance envelope.
- Low position limits distant visibility: only relatively slow-speed track.
- High-speed, poor clearance: signal bridge is more expensive solution.
- Frequently used for:
  - Transition tracks: sidings, spurs, branch lines, turn-outs, diamonds.
  - Signalling limited clearances, reduced speeds.
  - Alternative to standard high signals to reduce costs and maintenance.

Dwarf Railway Signals

- Dwarf signals serve the same purpose as typical high signals.
- They are small, short, wayside or trackside signals.
- Commonly found where there is restricted room between the tracks.
- Short enough to fit notch at bottom of normal clearance envelope.
- Low position limits distant visibility: only relatively slow-speed track.
- High-speed, poor clearance: signal bridge is more expensive solution.
- Frequently used for:
  - Transition tracks: sidings, spurs, branch lines, turn-outs, diamonds.
  - Signalling limited clearances, reduced speeds.
  - Alternative to standard high signals to reduce costs and maintenance.

Dwarf Railway Signals

- Dwarf signals serve the same purpose as typical high signals.
- They are small, short, wayside or trackside signals.
- Commonly found where there is restricted room between the tracks.
- Short enough to fit notch at bottom of normal clearance envelope.
- Low position limits distant visibility: only relatively slow-speed track.
- High-speed, poor clearance: signal bridge is more expensive solution.
- Frequently used for:
  - Transition tracks: sidings, spurs, branch lines, turn-outs, diamonds.
  - Signalling limited clearances, reduced speeds.
  - Alternative to standard high signals to reduce costs and maintenance.

Dwarf Railway Signals

- Dwarf signals serve the same purpose as typical high signals.
- They are small, short, wayside or trackside signals.
- Commonly found where there is restricted room between the tracks.
- Short enough to fit notch at bottom of normal clearance envelope.
- Low position limits distant visibility: only relatively slow-speed track.
- High-speed, poor clearance: signal bridge is more expensive solution.
- Frequently used for:
  - Transition tracks: sidings, spurs, branch lines, turn-outs, diamonds.
  - Signalling limited clearances, reduced speeds.
  - Alternative to standard high signals to reduce costs and maintenance.

Dwarf Railway Signals

- Dwarf signals serve the same purpose as typical high signals.
- They are small, short, wayside or trackside signals.
- Commonly found where there is restricted room between the tracks.
- Short enough to fit notch at bottom of normal clearance envelope.
- Low position limits distant visibility: only relatively slow-speed track.
- High-speed, poor clearance: signal bridge is more expensive solution.
- Frequently used for:
  - Transition tracks: sidings, spurs, branch lines, turn-outs, diamonds.
  - Signalling limited clearances, reduced speeds.
  - Alternative to standard high signals to reduce costs and maintenance.

Dwarf Railway Signals

- Dwarf signals serve the same purpose as typical high signals.
- They are small, short, wayside or trackside signals.
- Commonly found where there is restricted room between the tracks.
- Short enough to fit notch at bottom of normal clearance envelope.
- Low position limits distant visibility: only relatively slow-speed track.
- High-speed, poor clearance: signal bridge is more expensive solution.
- Frequently used for:
  - Transition tracks: sidings, spurs, branch lines, turn-outs, diamonds.
  - Signalling limited clearances, reduced speeds.
  - Alternative to standard high signals to reduce costs and maintenance.

Dwarf Railway Signals

- Dwarf signals serve the same purpose as typical high signals.
- They are small, short, wayside or trackside signals.
- Commonly found where there is restricted room between the tracks.
- Short enough to fit notch at bottom of normal clearance envelope.
- Low position limits distant visibility: only relatively slow-speed track.
- High-speed, poor clearance: signal bridge is more expensive solution.
- Frequently used for:
  - Transition tracks: sidings, spurs, branch lines, turn-outs, diamonds.
  - Signalling limited clearances, reduced speeds.
  - Alternative to standard high signals to reduce costs and maintenance.

Dwarf Railway Signals

- Dwarf signals serve the same purpose as typical high signals.
- They are small, short, wayside or trackside signals.
- Commonly found where there is restricted room between the tracks.
- Short enough to fit notch at bottom of normal clearance envelope.
- Low position limits distant visibility: only relatively slow-speed track.
- High-speed, poor clearance: signal bridge is more expensive solution.
- Frequently used for:
  - Transition tracks: sidings, spurs, branch lines, turn-outs, diamonds.
  - Signalling limited clearances, reduced speeds.
  - Alternative to standard high signals to reduce costs and maintenance.

Dwarf Railway Signals

- Dwarf signals serve the same purpose as typical high signals.
- They are small, short, wayside or trackside signals.
- Commonly found where there is restricted room between the tracks.
- Short enough to fit notch at bottom of normal clearance envelope.
- Low position limits distant visibility: only relatively slow-speed track.
- High-speed, poor clearance: signal bridge is more expensive solution.
- Frequently used for:
  - Transition tracks: sidings, spurs, branch lines, turn-outs, diamonds.
  - Signalling limited clearances, reduced speeds.
  - Alternative to standard high signals to reduce costs and maintenance.

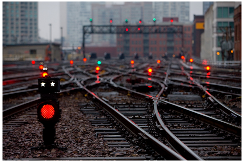
Dwarf Railway Signals

- Dwarf signals serve the same purpose as typical high signals.
- They are small, short, wayside or trackside signals.
- Commonly found where there is restricted room between the tracks.
- Short enough to fit notch at bottom of normal clearance envelope.
- Low position limits distant visibility: only relatively slow-speed track.
- High-speed, poor clearance: signal bridge is more expensive solution.
- Frequently used for:
  - Transition tracks: sidings, spurs, branch lines, turn-outs, diamonds.
  - Signalling limited clearances, reduced speeds.
  - Alternative to standard high signals to reduce costs and maintenance.

Dwarf Railway Signals

Dwarf Railway Signals








Our 3-Lamp Dwarf Signal

Our 3-Lamp Dwarf Signal

Different Aspects of the Dwarf Signal

- Our dwarf signal consists of three white lamps.
- Lamps can show two aspects: on (burning) and off (not burning) .
- If all equipment is physically intact, there are four different valid aspects:

Different Aspects of the Dwarf Signal

- Our dwarf signal consists of three white lamps.
- Lamps can show two aspects: on (burning) and off (not burning).
- If all equipment is physically intact, there are four different valid aspects:

Different Aspects of the Dwarf Signal

- Our dwarf signal consists of three white lamps.
- Lamps can show two aspects: on (burning) and off (not burning).
- If all equipment is physically intact, there are four different valid aspects:

Lamps

Dark

Stop

Warning

Drive

Different Aspects of the Dwarf Signal

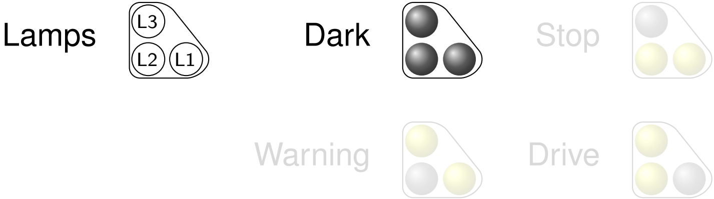
- Our dwarf signal consists of three white lamps.
- Lamps can show two aspects: on (burning) and off (not burning).
- If all equipment is physically intact, there are four different valid aspects:

Different Aspects of the Dwarf Signal

- Our dwarf signal consists of three white lamps.
- Lamps can show two aspects: on (burning) and off (not burning).
- If all equipment is physically intact, there are four different valid aspects:

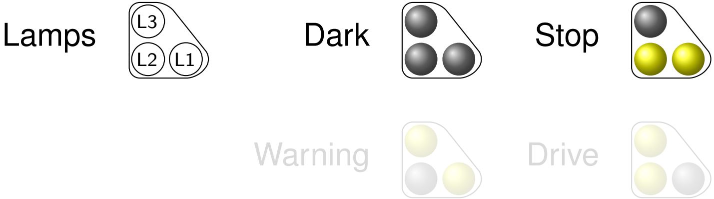
Different Aspects of the Dwarf Signal

- Our dwarf signal consists of three white lamps.
- Lamps can show two aspects: on (burning) and off (not burning) .
- If all equipment is physically intact, there are four different valid aspects:



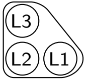
Different Aspects of the Dwarf Signal

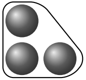
- Our dwarf signal consists of three white lamps.
- Lamps can show two aspects: on (burning) and off (not burning) .
- If all equipment is physically intact, there are four different valid aspects:

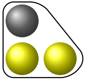


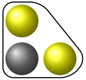
Different Aspects of the Dwarf Signal

- Our dwarf signal consists of three white lamps.
- Lamps can show two aspects: on (burning) and off (not burning).
- If all equipment is physically intact, there are four different valid aspects:

Lamps 

Dark 

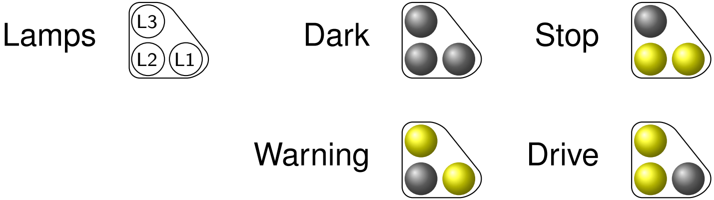
Stop 

Warning 

Drive

Different Aspects of the Dwarf Signal

- Our dwarf signal consists of three white lamps.
- Lamps can show two aspects: on (burning) and off (not burning).
- If all equipment is physically intact, there are four different valid aspects:



Interpreting the Different Aspects

- The safest aspect is stop.
- The dark aspect is used only to prolong the lifetime of the lamps.
- The dark aspect should never be seen by a driver in normal conditions.
- If drive can't be shown, warning should be shown.
- If warning can't be shown, stop should be shown instead.
- Drivers are assumed to interpret an invalid aspect in a safe way:
  - dark is interpreted as stop.
  - L1 burning on its own is interpreted as stop.
  - L2 burning on its own is interpreted as stop.
  - L3 burning on its own is interpreted as warning.

Interpreting the Different Aspects

- The safest aspect is stop.
- The dark aspect is used only to prolong the lifetime of the lamps.
- The dark aspect should never be seen by a driver in normal conditions.
- If drive can't be shown, warning should be shown.
- If warning can't be shown, stop should be shown instead.
- Drivers are assumed to interpret an invalid aspect in a safe way:
  - dark is interpreted as stop.
  - L1 burning on its own is interpreted as stop.
  - L2 burning on its own is interpreted as stop.
  - L3 burning on its own is interpreted as warning.

Interpreting the Different Aspects

- The safest aspect is stop.
- The dark aspect is used only to prolong the lifetime of the lamps.
- The dark aspect should never be seen by a driver in normal conditions.
- If drive can't be shown, warning should be shown.
- If warning can't be shown, stop should be shown instead.
- Drivers are assumed to interpret an invalid aspect in a safe way:
  - dark is interpreted as stop.
  - L1 burning on its own is interpreted as stop.
  - L2 burning on its own is interpreted as stop.
  - L3 burning on its own is interpreted as warning.

Interpreting the Different Aspects

- The safest aspect is stop.
- The dark aspect is used only to prolong the lifetime of the lamps.
- The dark aspect should never be seen by a driver in normal conditions.
- If drive can't be shown, warning should be shown.
- If warning can't be shown, stop should be shown instead.
- Drivers are assumed to interpret an invalid aspect in a safe way:
  - dark is interpreted as stop.
  - L1 burning on its own is interpreted as stop.
  - L2 burning on its own is interpreted as stop.
  - L3 burning on its own is interpreted as warning.

Interpreting the Different Aspects

- The safest aspect is stop.
- The dark aspect is used only to prolong the lifetime of the lamps.
- The dark aspect should never be seen by a driver in normal conditions.
- If drive can't be shown, warning should be shown.
- If warning can't be shown, stop should be shown instead.
- Drivers are assumed to interpret an invalid aspect in a safe way:
  - dark is interpreted as stop.
  - L1 burning on its own is interpreted as stop.
  - L2 burning on its own is interpreted as stop.
  - L3 burning on its own is interpreted as warning.

Interpreting the Different Aspects

- The safest aspect is stop.
- The dark aspect is used only to prolong the lifetime of the lamps.
- The dark aspect should never be seen by a driver in normal conditions.
- If drive can't be shown, warning should be shown.
- If warning can't be shown, stop should be shown instead.
- Drivers are assumed to interpret an invalid aspect in a safe way:
  - dark is interpreted as stop.
  - L1 burning on its own is interpreted as stop.
  - L2 burning on its own is interpreted as stop.
  - L3 burning on its own is interpreted as warning.

Interpreting the Different Aspects

- The safest aspect is stop.
- The dark aspect is used only to prolong the lifetime of the lamps.
- The dark aspect should never be seen by a driver in normal conditions.
- If drive can't be shown, warning should be shown.
- If warning can't be shown, stop should be shown instead.
- Drivers are assumed to interpret an invalid aspect in a safe way:
  - dark is interpreted as stop.
  - L1 burning on its own is interpreted as stop.
  - L2 burning on its own is interpreted as stop.
  - L3 burning on its own is interpreted as warning.

Interpreting the Different Aspects

- The safest aspect is stop.
- The dark aspect is used only to prolong the lifetime of the lamps.
- The dark aspect should never be seen by a driver in normal conditions.
- If drive can't be shown, warning should be shown.
- If warning can't be shown, stop should be shown instead.
- Drivers are assumed to interpret an invalid aspect in a safe way:
  - dark is interpreted as stop.
  - L1 burning on its own is interpreted as stop.
  - L2 burning on its own is interpreted as stop.
  - L3 burning on its own is interpreted as warning.

Interpreting the Different Aspects

- The safest aspect is stop.
- The dark aspect is used only to prolong the lifetime of the lamps.
- The dark aspect should never be seen by a driver in normal conditions.
- If drive can't be shown, warning should be shown.
- If warning can't be shown, stop should be shown instead.
- Drivers are assumed to interpret an invalid aspect in a safe way:
  - dark is interpreted as stop.
  - L1 burning on its own is interpreted as stop.
  - L2 burning on its own is interpreted as stop.
  - L3 burning on its own is interpreted as warning.

Interpreting the Different Aspects

- The safest aspect is stop.
- The dark aspect is used only to prolong the lifetime of the lamps.
- The dark aspect should never be seen by a driver in normal conditions.
- If drive can't be shown, warning should be shown.
- If warning can't be shown, stop should be shown instead.
- Drivers are assumed to interpret an invalid aspect in a safe way:
  - dark is interpreted as stop.
  - L1 burning on its own is interpreted as stop.
  - L2 burning on its own is interpreted as stop.
  - L3 burning on its own is interpreted as warning.

Interpreting the Different Aspects

- The safest aspect is stop.
- The dark aspect is used only to prolong the lifetime of the lamps.
- The dark aspect should never be seen by a driver in normal conditions.
- If drive can't be shown, warning should be shown.
- If warning can't be shown, stop should be shown instead.
- Drivers are assumed to interpret an invalid aspect in a safe way:
  - dark is interpreted as stop.
  - L1 burning on its own is interpreted as stop.
  - L2 burning on its own is interpreted as stop.
  - L3 burning on its own is interpreted as warning.

Rules for Switching between Aspects

- Only one lamp aspect may be changed at one time.
- The three lamps must never be on simultaneously.
- The signal must never be dark unless:
  - The dark aspect has to be shown or there is a lamp failure.
- The change to and from dark is allowed only to and from stop.
- Initial aspect and aspect in powerless state of actors is stop.

Rules for Switching between Aspects

- Only one lamp aspect may be changed at one time.
- The three lamps must never be on simultaneously.
- The signal must never be dark unless:
  - The dark aspect has to be shown or there is a lamp failure.
- The change to and from dark is allowed only to and from stop.
- Initial aspect and aspect in powerless state of actors is stop.

Rules for Switching between Aspects

- Only one lamp aspect may be changed at one time.
- The three lamps must never be on simultaneously.
- The signal must never be dark unless:
  - The dark aspect has to be shown or there is a lamp failure.
- The change to and from dark is allowed only to and from stop.
- Initial aspect and aspect in powerless state of actors is stop.

Rules for Switching between Aspects

- Only one lamp aspect may be changed at one time.
- The three lamps must never be on simultaneously.
- The signal must never be dark unless:
  - The dark aspect has to be shown or there is a lamp failure.
- The change to and from dark is allowed only to and from stop.
- Initial aspect and aspect in powerless state of actors is stop.

Rules for Switching between Aspects

- Only one lamp aspect may be changed at one time.
- The three lamps must never be on simultaneously.
- The signal must never be dark unless:
  - The dark aspect has to be shown or there is a lamp failure.
- The change to and from dark is allowed only to and from stop.
- Initial aspect and aspect in powerless state of actors is stop.

Rules for Switching between Aspects

- Only one lamp aspect may be changed at one time.
- The three lamps must never be on simultaneously.
- The signal must never be dark unless:
  - The dark aspect has to be shown or there is a lamp failure.
- The change to and from dark is allowed only to and from stop.
- Initial aspect and aspect in powerless state of actors is stop.

Rules for Switching between Aspects

- Only one lamp aspect may be changed at one time.
- The three lamps must never be on simultaneously.
- The signal must never be dark unless:
  - The dark aspect has to be shown or there is a lamp failure.
- The change to and from dark is allowed only to and from stop.
- Initial aspect and aspect in powerless state of actors is stop.

Overview

1. Introduction: a Dwarf Railway Signal
2. Formal Specification of the Dwarf Signal
3. Formalising the Safety Requirements
4. Specifying the Interface
5. A Use Case: Installation then Set to Warning

Formal Specification of the Dwarf Signal

- There are three lamp identifiers: $LampId ::= L1 \mid L2 \mid L3$.
- There are four aspects of the signal:

$$
\begin{array}{|l}
dark, stop, warning, drive : \mathbb{F}\ LampId \\
\hline
dark = \emptyset \\
stop = \{L1, L2\} \\
warning = \{L1, L3\} \\
drive = \{L2, L3\}
\end{array}
$$

- These are proper aspects of the signal.
- Transient aspects: $ProperState == \{dark, stop, warning, drive\}$.
- $Signal$ is the type of lamp identifier sets: $Signal == \mathbb{F}\ LampId$.

Formal Specification of the Dwarf Signal

- There are three lamp identifiers: $LampId ::= L1 \mid L2 \mid L3$.
- There are four aspects of the signal:

$$
\begin{array}{|l}
dark, stop, warning, drive : \mathbb{F}\ LampId \\
\hline
dark = \emptyset \\
stop = \{L1, L2\} \\
warning = \{L1, L3\} \\
drive = \{L2, L3\}
\end{array}
$$

- These are proper aspects of the signal.
- Transient aspects: $ProperState == \{dark, stop, warning, drive\}$.
- $Signal$ is the type of lamp identifier sets: $Signal == \mathbb{F}\ LampId$.

Formal Specification of the Dwarf Signal

- There are three lamp identifiers: $LampId ::= L1 \mid L2 \mid L3$.
- There are four aspects of the signal:

$$
\begin{array}{|l}
dark, stop, warning, drive : \mathbb{F}\, LampId \\
\hline
dark = \emptyset \\
stop = \{L1, L2\} \\
warning = \{L1, L3\} \\
drive = \{L2, L3\}
\end{array}
$$

- These are proper aspects of the signal.
- Transient aspects: $ProperState == \{dark, stop, warning, drive\}$.
- $Signal$ is the type of lamp identifier sets: $Signal == \mathbb{F}\, LampId$.

Formal Specification of the Dwarf Signal

- There are three lamp identifiers: $LampId ::= L1 \mid L2 \mid L3$.
- There are four aspects of the signal:

$$
\begin{array}{|l}
dark, stop, warning, drive : \mathbb{F}\, LampId \\
\hline
dark = \emptyset \\
stop = \{L1, L2\} \\
warning = \{L1, L3\} \\
drive = \{L2, L3\}
\end{array}
$$

- These are proper aspects of the signal.
- Transient aspects: $ProperState == \{dark, stop, warning, drive\}$.
- $Signal$ is the type of lamp identifier sets: $Signal == \mathbb{F}\, LampId$.

Formal Specification of the Dwarf Signal

- There are three lamp identifiers: $LampId ::= L1 \mid L2 \mid L3$.
- There are four aspects of the signal:

$$
\begin{array}{|l}
dark, stop, warning, drive : \mathbb{F}\, LampId \\
\hline
dark = \emptyset \\
stop = \{L1, L2\} \\
warning = \{L1, L3\} \\
drive = \{L2, L3\}
\end{array}
$$

- These are proper aspects of the signal.
- Transient aspects: $ProperState == \{dark, stop, warning, drive\}$.
- $Signal$ is the type of lamp identifier sets: $Signal == \mathbb{F}\, LampId$.

Formal Specification of the Dwarf Signal

- There are three lamp identifiers: $LampId ::= L1 \mid L2 \mid L3$.
- There are four aspects of the signal:

$$
\begin{array}{|l}
dark, stop, warning, drive : \mathbb{F}\, LampId \\
\hline
dark = \emptyset \\
stop = \{L1, L2\} \\
warning = \{L1, L3\} \\
drive = \{L2, L3\}
\end{array}
$$

- These are proper aspects of the signal.
- Transient aspects: $ProperState == \{dark, stop, warning, drive\}$.
- $Signal$ is the type of lamp identifier sets: $Signal == \mathbb{F}\, LampId$.

State Model for Dwarf Signal

We model the dwarf signal using six state variables:

1. $last\_proper\_state$: constrains signal transitions between proper aspects.
2. $turn\_off$: lamps that must be extinguished to reach desired state.
3. $turn\_on$: lamps that must be lit to reach desired state.
4. $last\_state$: constrains sequence of signal states (proper or transient).
   - Ensures that only one lamp may be changed at a time.
5. $current\_state$: set of lamps identifiers currently burning.
6. $desired\_proper\_state$: the next desired proper aspect of the signal.

State Model for Dwarf Signal

We model the dwarf signal using six state variables:

1. $last\_proper\_state$: constrains signal transitions between proper aspects.
2. $turn\_off$: lamps that must be extinguished to reach desired state.
3. $turn\_on$: lamps that must be lit to reach desired state.
4. $last\_state$: constrains sequence of signal states (proper or transient).
   - Ensures that only one lamp may be changed at a time.
5. $current\_state$: set of lamps identifiers currently burning.
6. $desired\_proper\_state$: the next desired proper aspect of the signal.

State Model for Dwarf Signal

We model the dwarf signal using six state variables:

1. $last\_proper\_state$: constrains signal transitions between proper aspects.
2. $turn\_off$: lamps that must be extinguished to reach desired state.
3. $turn\_on$: lamps that must be lit to reach desired state.
4. $last\_state$: constrains sequence of signal states (proper or transient).
   - Ensures that only one lamp may be changed at a time.
5. $current\_state$: set of lamps identifiers currently burning.
6. $desired\_proper\_state$: the next desired proper aspect of the signal.

State Model for Dwarf Signal

We model the dwarf signal using six state variables:

1. $last\_proper\_state$: constrains signal transitions between proper aspects.
2. $turn\_off$: lamps that must be extinguished to reach desired state.
3. $turn\_on$: lamps that must be lit to reach desired state.
4. $last\_state$: constrains sequence of signal states (proper or transient).
   - Ensures that only one lamp may be changed at a time.
5. $current\_state$: set of lamps identifiers currently burning.
6. $desired\_proper\_state$: the next desired proper aspect of the signal.

State Model for Dwarf Signal

We model the dwarf signal using six state variables:

1. $last\_proper\_state$: constrains signal transitions between proper aspects.
2. $turn\_off$: lamps that must be extinguished to reach desired state.
3. $turn\_on$: lamps that must be lit to reach desired state.
4. $last\_state$: constrains sequence of signal states (proper or transient).
   - Ensures that only one lamp may be changed at a time.
5. $current\_state$: set of lamps identifiers currently burning.
6. $desired\_proper\_state$: the next desired proper aspect of the signal.

State Model for Dwarf Signal

We model the dwarf signal using six state variables:

1. *last_proper_state*: constrains signal transitions between proper aspects.
2. *turn_off*: lamps that must be extinguished to reach desired state.
3. *turn_on*: lamps that must be lit to reach desired state.
4. *last_state*: constrains sequence of signal states (proper or transient).
   - Ensures that only one lamp may be changed at a time.
5. *current_state*: set of lamps identifiers currently burning.
6. *desired_proper_state*: the next desired proper aspect of the signal.

State Model for Dwarf Signal

We model the dwarf signal using six state variables:

1. $last\_proper\_state$: constrains signal transitions between proper aspects.
2. $turn\_off$: lamps that must be extinguished to reach desired state.
3. $turn\_on$: lamps that must be lit to reach desired state.
4. $last\_state$: constrains sequence of signal states (proper or transient).
   - Ensures that only one lamp may be changed at a time.
5. $current\_state$: set of lamps identifiers currently burning.
6. $desired\_proper\_state$: the next desired proper aspect of the signal.

State Model for Dwarf Signal

We model the dwarf signal using six state variables:

1. $last\_proper\_state$: constrains signal transitions between proper aspects.
2. $turn\_off$: lamps that must be extinguished to reach desired state.
3. $turn\_on$: lamps that must be lit to reach desired state.
4. $last\_state$: constrains sequence of signal states (proper or transient).
   - Ensures that only one lamp may be changed at a time.
5. $current\_state$: set of lamps identifiers currently burning.
6. $desired\_proper\_state$: the next desired proper aspect of the signal.

State Model for Dwarf Signal

We model the dwarf signal using six state variables:

1. $last\_proper\_state$: constrains signal transitions between proper aspects.
2. $turn\_off$: lamps that must be extinguished to reach desired state.
3. $turn\_on$: lamps that must be lit to reach desired state.
4. $last\_state$: constrains sequence of signal states (proper or transient).
   - Ensures that only one lamp may be changed at a time.
5. $current\_state$: set of lamps identifiers currently burning.
6. $desired\_proper\_state$: the next desired proper aspect of the signal.

State Invariants

Two invariants link the state components:

Invariant 1 The changes indicated by the two sets $turn\_off$ and $turn\_on$ will transform the current state into the desired state.

Invariant 2 It can never be the case that a lamp needs to be both turned on and turned off to reach the desired proper state.

A consequence of these two invariants:

Desired state reached when there are no more lamps to turn off or on.

State Invariants

Two invariants link the state components:

Invariant 1 The changes indicated by the two sets $turn\_off$ and $turn\_on$ will transform the current state into the desired state.

Invariant 2 It can never be the case that a lamp needs to be both turned on and turned off to reach the desired proper state.

A consequence of these two invariants:

Desired state reached when there are no more lamps to turn off or on.

State Invariants

Two invariants link the state components:

Invariant 1 The changes indicated by the two sets $turn\_off$ and $turn\_on$ will transform the current state into the desired state.

Invariant 2 It can never be the case that a lamp needs to be both turned on and turned off to reach the desired proper state.

A consequence of these two invariants:

Desired state reached when there are no more lamps to turn off or on.

State Invariants

Two invariants link the state components:

Invariant 1 The changes indicated by the two sets $turn\_off$ and $turn\_on$ will transform the current state into the desired state.

Invariant 2 It can never be the case that a lamp needs to be both turned on and turned off to reach the desired proper state.

A consequence of these two invariants:

Desired state reached when there are no more lamps to turn off or on.

State Invariants

Two invariants link the state components:

Invariant 1 The changes indicated by the two sets $turn\_off$ and $turn\_on$ will transform the current state into the desired state.

Invariant 2 It can never be the case that a lamp needs to be both turned on and turned off to reach the desired proper state.

A consequence of these two invariants:

Desired state reached when there are no more lamps to turn off or on.

State Invariants

Two invariants link the state components:

Invariant 1 The changes indicated by the two sets $turn\_off$ and $turn\_on$ will transform the current state into the desired state.

Invariant 2 It can never be the case that a lamp needs to be both turned on and turned off to reach the desired proper state.

A consequence of these two invariants:

Desired state reached when there are no more lamps to turn off or on.

Formal Specification of the State of the Dwarf Signal

Dwarf

$$last\_proper\_state : ProperState$$
$$turn\_off, turn\_on : \mathbb{F}\ LampId$$
$$last\_state, current\_state : Signal$$
$$desired\_proper\_state : ProperState$$

$$(current\_state \setminus turn\_off) \cup turn\_on = desired\_proper\_state$$
$$turn\_off \cap turn\_on = \emptyset$$

Formal Specification of the State of the Dwarf Signal

Dwarf

$last\_proper\_state : ProperState$
$turn\_off, turn\_on : \mathbb{F}\ LampId$
$last\_state, current\_state : Signal$
$desired\_proper\_state : ProperState$

$(current\_state \setminus turn\_off) \cup turn\_on = desired\_proper\_state$
$turn\_off \cap turn\_on = \emptyset$

Formal Specification of the State of the Dwarf Signal

Dwarf

$$last\_proper\_state : ProperState$$
$$turn\_off, turn\_on : \mathbb{F}\ LampId$$
$$last\_state, current\_state : Signal$$
$$desired\_proper\_state : ProperState$$

$$(current\_state \setminus turn\_off) \cup turn\_on = desired\_proper\_state$$
$$turn\_off \cap turn\_on = \emptyset$$

Formal Specification of the State of the Dwarf Signal

$Dwarf$

$last\_proper\_state : ProperState$
$turn\_off, turn\_on : \mathbb{F}\ LampId$
$last\_state, current\_state : Signal$
$desired\_proper\_state : ProperState$

$(current\_state \setminus turn\_off) \cup turn\_on = desired\_proper\_state$
$turn\_off \cap turn\_on = \emptyset$

Formal Specification of the State of the Dwarf Signal

$$
\begin{array}{l}
\textit{Dwarf} \\
\hline
last\_proper\_state : ProperState \\
turn\_off, turn\_on : \mathbb{F}\ LampId \\
last\_state, current\_state : Signal \\
desired\_proper\_state : ProperState \\
\hline
(current\_state \setminus turn\_off) \cup turn\_on = desired\_proper\_state \\
turn\_off \cap turn\_on = \emptyset \\
\hline
\end{array}
$$

Formal Specification of the State of the Dwarf Signal

$$
\begin{array}{l}
\textit{Dwarf} \\
\hline
\textit{last\_proper\_state} : \textit{ProperState} \\
\textit{turn\_off}, \textit{turn\_on} : \mathbb{F}\ \textit{LampId} \\
\textit{last\_state}, \textit{current\_state} : \textit{Signal} \\
\textit{desired\_proper\_state} : \textit{ProperState} \\
\hline
(\textit{current\_state} \setminus \textit{turn\_off}) \cup \textit{turn\_on} = \textit{desired\_proper\_state} \\
\textit{turn\_off} \cap \textit{turn\_on} = \emptyset
\end{array}
$$

Formal Specification of the State of the Dwarf Signal

$Dwarf$

$last\_proper\_state : ProperState$
$turn\_off, turn\_on : \mathbb{F}\, LampId$
$last\_state, current\_state : Signal$
$desired\_proper\_state : ProperState$

$(current\_state \setminus turn\_off) \cup turn\_on = desired\_proper\_state$
$turn\_off \cap turn\_on = \emptyset$

Formal Specification of the State of the Dwarf Signal

$$
\begin{array}{l}
\textit{Dwarf} \\
\hline
\textit{last\_proper\_state} : \textit{ProperState} \\
\textit{turn\_off}, \textit{turn\_on} : \mathbb{F}\, \textit{LampId} \\
\textit{last\_state}, \textit{current\_state} : \textit{Signal} \\
\textit{desired\_proper\_state} : \textit{ProperState} \\
\hline
(\textit{current\_state} \setminus \textit{turn\_off}) \cup \textit{turn\_on} = \textit{desired\_proper\_state} \\
\textit{turn\_off} \cap \textit{turn\_on} = \emptyset
\end{array}
$$

Formal Specification of the State of the Dwarf Signal

$$
\begin{array}{l}
\textit{Dwarf} \\
\hline
\mathit{last\_proper\_state} : \mathit{ProperState} \\
\mathit{turn\_off}, \mathit{turn\_on} : \mathbb{F}\, \mathit{LampId} \\
\mathit{last\_state}, \mathit{current\_state} : \mathit{Signal} \\
\mathit{desired\_proper\_state} : \mathit{ProperState} \\
\hline
(\mathit{current\_state} \setminus \mathit{turn\_off}) \cup \mathit{turn\_on} = \mathit{desired\_proper\_state} \\
\mathit{turn\_off} \cap \mathit{turn\_on} = \emptyset \\
\hline
\end{array}
$$

Formal Specification of the State of the Dwarf Signal

$Dwarf$

$last\_proper\_state : ProperState$
$turn\_off, turn\_on : \mathbb{F}\ LampId$
$last\_state, current\_state : Signal$
$desired\_proper\_state : ProperState$

$(current\_state \setminus turn\_off) \cup turn\_on = desired\_proper\_state$
$turn\_off \cap turn\_on = \emptyset$

Formal Specification of the State of the Dwarf Signal

$$
\begin{array}{l}
\textit{Dwarf} \\
\hline
last\_proper\_state : ProperState \\
turn\_off, turn\_on : \mathbb{F}\ LampId \\
last\_state, current\_state : Signal \\
desired\_proper\_state : ProperState \\
\hline
(current\_state \setminus turn\_off) \cup turn\_on = desired\_proper\_state \\
turn\_off \cap turn\_on = \emptyset
\end{array}
$$

Formal Specification of the State of the Dwarf Signal

$Dwarf$

$last\_proper\_state : ProperState$
$turn\_off, turn\_on : \mathbb{F}\ LampId$
$last\_state, current\_state : Signal$
$desired\_proper\_state : ProperState$

$(current\_state \setminus turn\_off) \cup turn\_on = desired\_proper\_state$
$turn\_off \cap turn\_on = \emptyset$

Formal Specification of the State of the Dwarf Signal

$$
\begin{array}{l}
\textit{Dwarf} \\
\hline
last\_proper\_state : ProperState \\
turn\_off, turn\_on : \mathbb{F}\ LampId \\
last\_state, current\_state : Signal \\
desired\_proper\_state : ProperState \\
\hline
(current\_state \setminus turn\_off) \cup turn\_on = desired\_proper\_state \\
turn\_off \cap turn\_on = \emptyset \\
\hline
\end{array}
$$

Formal Specification of the State of the Dwarf Signal

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{Dwarf}} \\
\hline
last\_proper\_state : ProperState \\
turn\_off, turn\_on : \mathbb{F}\ LampId \\
last\_state, current\_state : Signal \\
desired\_proper\_state : ProperState \\
\hline
(current\_state \setminus turn\_off) \cup turn\_on = desired\_proper\_state \\
turn\_off \cap turn\_on = \emptyset \\
\hline
\end{array}
$$

Overview

Formalising the Safety Requirements

- There are five safety requirements for the dwarf signal.
- We specify each safety requirement separately.
- This principle is known as separation of concerns.
- Each safety requirement is a constraint on the dwarf signal state.

Formalising the Safety Requirements

- There are five safety requirements for the dwarf signal.
- We specify each safety requirement separately.
- This principle is known as separation of concerns.
- Each safety requirement is a constraint on the dwarf signal state.

Formalising the Safety Requirements

- There are five safety requirements for the dwarf signal.
- We specify each safety requirement separately.
- This principle is known as separation of concerns.
- Each safety requirement is a constraint on the dwarf signal state.

Formalising the Safety Requirements

- There are five safety requirements for the dwarf signal.
- We specify each safety requirement separately.
- This principle is known as separation of concerns.
- Each safety requirement is a constraint on the dwarf signal state.

Formalising the Safety Requirements

- There are five safety requirements for the dwarf signal.
- We specify each safety requirement separately.
- This principle is known as separation of concerns.
- Each safety requirement is a constraint on the dwarf signal state.

Safety 1: The signal should never light all its lamps

NeverShowAll

Dwarf

$current\_state \neq \{L1, L2, L3\}$

Safety 1: The signal should never light all its lamps

$NeverShowAll$

$Dwarf$

$current\_state \neq \{L1, L2, L3\}$

Safety 1: The signal should never light all its lamps

$NeverShowAll$

$Dwarf$

$current\_state \neq \{L1, L2, L3\}$

Safety 1: The signal should never light all its lamps

$$\begin{array}{|l} \hline \textit{NeverShowAll} \\ \textit{Dwarf} \\ \hline current\_state \neq \{L1, L2, L3\} \\ \hline \end{array}$$

Safety 2: Only one lamp should be changed at a time

$$
\begin{array}{l}
\textit{MaxOneLampChange} \\
\hline
\textit{Dwarf} \\
\hline
\exists\, l : \textit{LampId} \bullet \\
\quad \textit{current\_state} \setminus \textit{last\_state} = \{l\} \\
\quad \vee \\
\quad \textit{last\_state} \setminus \textit{current\_state} = \{l\} \\
\quad \vee \\
\quad \textit{last\_state} = \textit{current\_state}
\end{array}
$$

Safety 2: Only one lamp should be changed at a time

MaxOneLampChange

Dwarf

$$\exists\, l : LampId \bullet$$
$$current\_state \setminus last\_state = \{l\}$$
$$\vee$$
$$last\_state \setminus current\_state = \{l\}$$
$$\vee$$
$$last\_state = current\_state$$

Safety 2: Only one lamp should be changed at a time

$$
\begin{array}{l}
\textit{MaxOneLampChange} \\
\hline
\textit{Dwarf} \\
\hline
\exists\, l : \textit{LampId} \bullet \\
\qquad current\_state \setminus last\_state = \{l\} \\
\qquad \lor \\
\qquad last\_state \setminus current\_state = \{l\} \\
\qquad \lor \\
\qquad last\_state = current\_state
\end{array}
$$

Safety 2: Only one lamp should be changed at a time

$MaxOneLampChange$

$Dwarf$

$\exists\, l : LampId \bullet$

$current\_state \setminus last\_state = \{l\}$

$\vee$

$last\_state \setminus current\_state = \{l\}$

$\vee$

$last\_state = current\_state$

Safety 2: Only one lamp should be changed at a time

$$
\begin{array}{l}
\textit{MaxOneLampChange} \\
\hline
\textit{Dwarf} \\
\hline
\exists\, l : \textit{LampId} \bullet \\
\qquad \textit{current\_state} \setminus \textit{last\_state} = \{l\} \\
\qquad \lor \\
\qquad \textit{last\_state} \setminus \textit{current\_state} = \{l\} \\
\qquad \lor \\
\qquad \textit{last\_state} = \textit{current\_state}
\end{array}
$$

Safety 2: Only one lamp should be changed at a time

$$
\begin{array}{l}
\textit{MaxOneLampChange} \\
\hline
\textit{Dwarf} \\
\hline
\exists\, l : \textit{LampId} \bullet \\
\quad \textit{current\_state} \setminus \textit{last\_state} = \{l\} \\
\quad \vee \\
\quad \textit{last\_state} \setminus \textit{current\_state} = \{l\} \\
\quad \vee \\
\quad \textit{last\_state} = \textit{current\_state}
\end{array}
$$

Safety 2: Only one lamp should be changed at a time

$$
\begin{array}{l}
\textit{MaxOneLampChange} \\
\hline
\textit{Dwarf} \\
\hline
\exists\, l : \textit{LampId} \bullet \\
\qquad \textit{current\_state} \setminus \textit{last\_state} = \{l\} \\
\qquad \vee \\
\qquad \textit{last\_state} \setminus \textit{current\_state} = \{l\} \\
\qquad \vee \\
\qquad \textit{last\_state} = \textit{current\_state}
\end{array}
$$

Safety 2: Only one lamp should be changed at a time

$$
\begin{array}{l}
\textit{MaxOneLampChange} \\
\hline
\textit{Dwarf} \\
\hline
\exists\, l : \textit{LampId} \bullet \\
\quad \textit{current\_state} \setminus \textit{last\_state} = \{l\} \\
\quad \lor \\
\quad \textit{last\_state} \setminus \textit{current\_state} = \{l\} \\
\quad \lor \\
\quad \textit{last\_state} = \textit{current\_state}
\end{array}
$$

Safety 2: Only one lamp should be changed at a time

$$
\begin{array}{|l}
\textit{MaxOneLampChange} \\
\hline
\textit{Dwarf} \\
\hline
\exists\, l : \textit{LampId} \bullet \\
\qquad \textit{current\_state} \setminus \textit{last\_state} = \{l\} \\
\qquad \lor \\
\qquad \textit{last\_state} \setminus \textit{current\_state} = \{l\} \\
\qquad \lor \\
\qquad \textit{last\_state} = \textit{current\_state} \\
\hline
\end{array}
$$

Safety 3: It is forbidden to change signal from stop to drive

$$\begin{array}{|l} \multicolumn{1}{l}{\textit{ForbidStopToDrive}} \\ \hline \textit{Dwarf} \\ \hline last\_proper\_state = stop \Rightarrow desired\_proper\_state \neq drive \\ \hline \end{array}$$

Safety 3: It is forbidden to change signal from stop to drive

$ForbidStopToDrive$

$Dwarf$

$last\_proper\_state = stop \Rightarrow desired\_proper\_state \neq drive$

Safety 3: It is forbidden to change signal from stop to drive

ForbidStopToDrive

Dwarf

$$last\_proper\_state = stop \Rightarrow desired\_proper\_state \neq drive$$

Safety 3: It is forbidden to change signal from stop to drive

$$
\begin{array}{l}
\textit{ForbidStopToDrive} \\
\hline
\textit{Dwarf} \\
\hline
\textit{last\_proper\_state} = \textit{stop} \Rightarrow \textit{desired\_proper\_state} \neq \textit{drive} \\
\hline
\end{array}
$$

Safety 4: The only proper aspect following dark is stop

$DarkOnlyToStop$

$Dwarf$

$last\_proper\_state = dark \Rightarrow desired\_proper\_state = stop$

Safety 4: The only proper aspect following dark is stop

$DarkOnlyToStop$

$Dwarf$

$last\_proper\_state = dark \Rightarrow desired\_proper\_state = stop$

Safety 4: The only proper aspect following dark is stop

$DarkOnlyToStop$

$Dwarf$

$last\_proper\_state = dark \Rightarrow desired\_proper\_state = stop$

Safety 4: The only proper aspect following dark is stop

$DarkOnlyToStop$

$Dwarf$

$$last\_proper\_state = dark \Rightarrow desired\_proper\_state = stop$$

Safety 5: The only proper aspect preceding dark is stop

$$
\begin{array}{|l}
\hline
\textit{DarkOnlyFromStop} \\
\textit{Dwarf} \\
\hline
\textit{desired\_proper\_state} = \textit{dark} \Rightarrow \textit{last\_proper\_state} = \textit{stop} \\
\hline
\end{array}
$$

Safety 5: The only proper aspect preceding dark is stop

$DarkOnlyFromStop$

$Dwarf$

$desired\_proper\_state = dark \Rightarrow last\_proper\_state = stop$

Safety 5: The only proper aspect preceding dark is stop

$$
\begin{array}{|l}
\hline
\textit{DarkOnlyFromStop} \\
\textit{Dwarf} \\
\hline
desired\_proper\_state = dark \Rightarrow last\_proper\_state = stop \\
\hline
\end{array}
$$

Safety 5: The only proper aspect preceding dark is stop

$$
\begin{array}{l}
\textit{DarkOnlyFromStop} \\
\hline
\textit{Dwarf} \\
\hline
\textit{desired\_proper\_state} = \textit{dark} \Rightarrow \textit{last\_proper\_state} = \textit{stop} \\
\hline
\end{array}
$$

Safety Specification for the Dwarf Signal

- Each safety requirement is a constraint on the state $Dwarf$.
- The dwarf signal must satisfy all five safety requirements.
- The $DwarfSignal$ state is the safe restriction of the $Dwarf$.

$$
\begin{array}{|l}
\hline
\textit{DwarfSignal} \\
\hline
NeverShowAll \\
MaxOneLampChange \\
ForbidStopToDrive \\
DarkOnlyToStop \\
DarkOnlyFromStop \\
\hline
\end{array}
$$

Safety Specification for the Dwarf Signal

- Each safety requirement is a constraint on the state $Dwarf$.
- The dwarf signal must satisfy all five safety requirements.
- The $DwarfSignal$ state is the safe restriction of the $Dwarf$.

$$
\begin{array}{|l}
\hline
\textbf{DwarfSignal} \\
\hline
NeverShowAll \\
MaxOneLampChange \\
ForbidStopToDrive \\
DarkOnlyToStop \\
DarkOnlyFromStop \\
\hline
\end{array}
$$

Safety Specification for the Dwarf Signal

- Each safety requirement is a constraint on the state $Dwarf$.
- The dwarf signal must satisfy all five safety requirements.
- The $DwarfSignal$ state is the safe restriction of the $Dwarf$.

$$
\begin{array}{|l}
\hline
\textit{DwarfSignal} \\
\hline
NeverShowAll \\
MaxOneLampChange \\
ForbidStopToDrive \\
DarkOnlyToStop \\
DarkOnlyFromStop \\
\hline
\end{array}
$$

Safety Specification for the Dwarf Signal

- Each safety requirement is a constraint on the state $Dwarf$.
- The dwarf signal must satisfy all five safety requirements.
- The $DwarfSignal$ state is the safe restriction of the $Dwarf$.

$DwarfSignal$

$NeverShowAll$
$MaxOneLampChange$
$ForbidStopToDrive$
$DarkOnlyToStop$
$DarkOnlyFromStop$

Safety Specification for the Dwarf Signal

- Each safety requirement is a constraint on the state $Dwarf$.
- The dwarf signal must satisfy all five safety requirements.
- The $DwarfSignal$ state is the safe restriction of the $Dwarf$.

$$
\begin{array}{|l}
\hline
\multicolumn{1}{l}{DwarfSignal} \\
NeverShowAll \\
MaxOneLampChange \\
ForbidStopToDrive \\
DarkOnlyToStop \\
DarkOnlyFromStop \\
\hline
\end{array}
$$

Safety Specification for the Dwarf Signal

- Each safety requirement is a constraint on the state $Dwarf$.
- The dwarf signal must satisfy all five safety requirements.
- The $DwarfSignal$ state is the safe restriction of the $Dwarf$.

$$
\begin{array}{|l}
\hline
\textit{DwarfSignal} \\
\hline
\textit{NeverShowAll} \\
\textit{MaxOneLampChange} \\
\textit{ForbidStopToDrive} \\
\textit{DarkOnlyToStop} \\
\textit{DarkOnlyFromStop} \\
\hline
\end{array}
$$

Safety Specification for the Dwarf Signal

- Each safety requirement is a constraint on the state $Dwarf$.
- The dwarf signal must satisfy all five safety requirements.
- The $DwarfSignal$ state is the safe restriction of the $Dwarf$.

$$
\begin{array}{|l}
\hline
\multicolumn{1}{l}{DwarfSignal} \\
NeverShowAll \\
MaxOneLampChange \\
ForbidStopToDrive \\
DarkOnlyToStop \\
DarkOnlyFromStop \\
\hline
\end{array}
$$

Safety Specification for the Dwarf Signal

- Each safety requirement is a constraint on the state $Dwarf$.
- The dwarf signal must satisfy all five safety requirements.
- The $DwarfSignal$ state is the safe restriction of the $Dwarf$.

$$
\begin{array}{|l}
\hline
\textit{DwarfSignal} \\
\hline
\textit{NeverShowAll} \\
\textit{MaxOneLampChange} \\
\textit{ForbidStopToDrive} \\
\textit{DarkOnlyToStop} \\
\textit{DarkOnlyFromStop} \\
\hline
\end{array}
$$

Safety Specification for the Dwarf Signal

- Each safety requirement is a constraint on the state $Dwarf$.
- The dwarf signal must satisfy all five safety requirements.
- The $DwarfSignal$ state is the safe restriction of the $Dwarf$.

$$
\begin{array}{|l}
\hline
\textit{DwarfSignal} \\
\hline
\textit{NeverShowAll} \\
\textit{MaxOneLampChange} \\
\textit{ForbidStopToDrive} \\
\textit{DarkOnlyToStop} \\
\textit{DarkOnlyFromStop} \\
\hline
\end{array}
$$

Safety Specification for the Dwarf Signal

- Each safety requirement is a constraint on the state $Dwarf$.
- The dwarf signal must satisfy all five safety requirements.
- The $DwarfSignal$ state is the safe restriction of the $Dwarf$.

$$
\begin{array}{|l}
\hline
DwarfSignal \\
NeverShowAll \\
MaxOneLampChange \\
ForbidStopToDrive \\
DarkOnlyToStop \\
DarkOnlyFromStop \\
\hline
\end{array}
$$

Overview

Initialisation for the Dwarf Signal

Initially, the dwarf signal shows the stop aspect:

$$
\begin{array}{|l}
\hline
\textit{Init} \\
\hline
\mathit{DwarfSignal}' \\
\hline
\mathit{last\_proper\_state}' = \mathit{stop} \\
\mathit{turn\_off}' = \emptyset \\
\mathit{turn\_on}' = \emptyset \\
\mathit{last\_state}' = \mathit{stop} \\
\mathit{current\_state}' = \mathit{stop} \\
\mathit{desired\_proper\_state}' = \mathit{stop} \\
\hline
\end{array}
$$

Initialisation for the Dwarf Signal

Initially, the dwarf signal shows the stop aspect:

$$
\begin{array}{|l}
\hline
\textit{Init} \\
\textit{DwarfSignal}' \\
\hline
\textit{last\_proper\_state}' = \textit{stop} \\
\textit{turn\_off}' = \emptyset \\
\textit{turn\_on}' = \emptyset \\
\textit{last\_state}' = \textit{stop} \\
\textit{current\_state}' = \textit{stop} \\
\textit{desired\_proper\_state}' = \textit{stop} \\
\hline
\end{array}
$$

Initialisation for the Dwarf Signal

Initially, the dwarf signal shows the stop aspect:

$$
\begin{array}{|l}
\hline
\textit{Init} \\
\hline
\textit{DwarfSignal}' \\
\hline
\textit{last\_proper\_state}' = \textit{stop} \\
\textit{turn\_off}' = \emptyset \\
\textit{turn\_on}' = \emptyset \\
\textit{last\_state}' = \textit{stop} \\
\textit{current\_state}' = \textit{stop} \\
\textit{desired\_proper\_state}' = \textit{stop} \\
\hline
\end{array}
$$

Initialisation for the Dwarf Signal

Initially, the dwarf signal shows the stop aspect:

$$
\begin{array}{l}
\textbf{Init} \\
\hline
DwarfSignal' \\
\hline
last\_proper\_state' = stop \\
turn\_off' = \emptyset \\
turn\_on' = \emptyset \\
last\_state' = stop \\
current\_state' = stop \\
desired\_proper\_state' = stop
\end{array}
$$

Initialisation for the Dwarf Signal

Initially, the dwarf signal shows the stop aspect:

$$
\begin{array}{l}
\mathit{Init} \\
\hline
\mathit{DwarfSignal}' \\
\hline
\mathit{last\_proper\_state}' = \mathit{stop} \\
\mathit{turn\_off}' = \emptyset \\
\mathit{turn\_on}' = \emptyset \\
\mathit{last\_state}' = \mathit{stop} \\
\mathit{current\_state}' = \mathit{stop} \\
\mathit{desired\_proper\_state}' = \mathit{stop}
\end{array}
$$

Initialisation for the Dwarf Signal

Initially, the dwarf signal shows the stop aspect:

$$
\begin{array}{|l}
\hline
\textit{Init} \\
\textit{DwarfSignal}' \\
\hline
\textit{last\_proper\_state}' = \textit{stop} \\
\textit{turn\_off}' = \emptyset \\
\textit{turn\_on}' = \emptyset \\
\textit{last\_state}' = \textit{stop} \\
\textit{current\_state}' = \textit{stop} \\
\textit{desired\_proper\_state}' = \textit{stop} \\
\hline
\end{array}
$$

Initialisation for the Dwarf Signal

Initially, the dwarf signal shows the stop aspect:

$$
\begin{array}{|l}
\hline
\textit{Init} \\
\textit{DwarfSignal}' \\
\hline
\textit{last\_proper\_state}' = \textit{stop} \\
\textit{turn\_off}' = \emptyset \\
\textit{turn\_on}' = \emptyset \\
\textit{last\_state}' = \textit{stop} \\
\textit{current\_state}' = \textit{stop} \\
\textit{desired\_proper\_state}' = \textit{stop} \\
\hline
\end{array}
$$

Initialisation for the Dwarf Signal

Initially, the dwarf signal shows the stop aspect:

$$
\begin{array}{l}
\textit{Init} \\
\hline
\textit{DwarfSignal}' \\
\hline
\textit{last\_proper\_state}' = \textit{stop} \\
\textit{turn\_off}' = \emptyset \\
\textit{turn\_on}' = \emptyset \\
\textit{last\_state}' = \textit{stop} \\
\textit{current\_state}' = \textit{stop} \\
\textit{desired\_proper\_state}' = \textit{stop}
\end{array}
$$

Initialisation for the Dwarf Signal

Initially, the dwarf signal shows the stop aspect:

$$
\begin{array}{l}
\textit{Init} \\
\hline
\textit{DwarfSignal}' \\
\hline
\textit{last\_proper\_state}' = \textit{stop} \\
\textit{turn\_off}' = \emptyset \\
\textit{turn\_on}' = \emptyset \\
\textit{last\_state}' = \textit{stop} \\
\textit{current\_state}' = \textit{stop} \\
\textit{desired\_proper\_state}' = \textit{stop} \\
\hline
\end{array}
$$

Initialisation for the Dwarf Signal

Initially, the dwarf signal shows the stop aspect:

$$
\begin{array}{|l}
\hline
\textit{Init} \\
\textit{DwarfSignal}' \\
\hline
\textit{last\_proper\_state}' = \textit{stop} \\
\textit{turn\_off}' = \emptyset \\
\textit{turn\_on}' = \emptyset \\
\textit{last\_state}' = \textit{stop} \\
\textit{current\_state}' = \textit{stop} \\
\textit{desired\_proper\_state}' = \textit{stop} \\
\hline
\end{array}
$$

Operations on the Dwarf Signal

There are three operations to change the state of the dwarf signal:

SetNewProperState The operator (the railway signaller) sets the new desired proper state. This is the only external operation.

TurnOff An internal operation that turns off one of the lamps needed to make progress towards the new desired state.

TurnOn Another internal operation that turns on one of the lamps needed to make progress towards the new desired state.

Operations on the Dwarf Signal

There are three operations to change the state of the dwarf signal:

$SetNewProperState$ The operator (the railway signaller) sets the new desired proper state. This is the only external operation.

$TurnOff$ An internal operation that turns off one of the lamps needed to make progress towards the new desired state.

$TurnOn$ Another internal operation that turns on one of the lamps needed to make progress towards the new desired state.

Operations on the Dwarf Signal

There are three operations to change the state of the dwarf signal:

$SetNewProperState$ The operator (the railway signaller) sets the new desired proper state. This is the only external operation.

$TurnOff$ An internal operation that turns off one of the lamps needed to make progress towards the new desired state.

$TurnOn$ Another internal operation that turns on one of the lamps needed to make progress towards the new desired state.

Operations on the Dwarf Signal

There are three operations to change the state of the dwarf signal:

$SetNewProperState$ The operator (the railway signaller) sets the new desired proper state. This is the only external operation.

$TurnOff$ An internal operation that turns off one of the lamps needed to make progress towards the new desired state.

$TurnOn$ Another internal operation that turns on one of the lamps needed to make progress towards the new desired state.

Operations on the Dwarf Signal

There are three operations to change the state of the dwarf signal:

$SetNewProperState$ The operator (the railway signaller) sets the new desired proper state. This is the only external operation.

$TurnOff$ An internal operation that turns off one of the lamps needed to make progress towards the new desired state.

$TurnOn$ Another internal operation that turns on one of the lamps needed to make progress towards the new desired state.

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
\textit{SetNewProperState} \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st? \\
\hline
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{|l}
\textbf{SetNewProperState} \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st? \\
\hline
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
\textbf{SetNewProperState} \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st? \\
\hline
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
\textbf{SetNewProperState} \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st?
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
\textbf{SetNewProperState} \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st? \\
\hline
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
SetNewProperState \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st?
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
SetNewProperState \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st?
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
\textit{SetNewProperState} \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st? \\
\hline
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
SetNewProperState \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st?
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
\textbf{SetNewProperState} \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st? \\
\hline
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$SetNewProperState$$

$$\Delta DwarfSignal$$
$$st? : ProperState$$

$$current\_state = desired\_proper\_state$$
$$st? \neq current\_state$$
$$last\_proper\_state' = current\_state$$
$$turn\_off' = current\_state \setminus st?$$
$$turn\_on' = st? \setminus current\_state$$
$$last\_state' = current\_state$$
$$current\_state' = current\_state$$
$$desired\_proper\_state' = st?$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
\textit{SetNewProperState} \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st? \\
\hline
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
SetNewProperState \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st?
\end{array}
$$

Setting the New Desired Proper State

- Operation on the state $DwarfSignal$ with one input: $st? : ProperState$.
- Preconditions: no outstanding actions and a genuinely new proper state.

$$
\begin{array}{l}
SetNewProperState \\
\hline
\Delta DwarfSignal \\
st? : ProperState \\
\hline
current\_state = desired\_proper\_state \\
st? \neq current\_state \\
last\_proper\_state' = current\_state \\
turn\_off' = current\_state \setminus st? \\
turn\_on' = st? \setminus current\_state \\
last\_state' = current\_state \\
current\_state' = current\_state \\
desired\_proper\_state' = st? \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$$
\begin{array}{|l}
\multicolumn{1}{l}{TurnOff} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_off \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \setminus \{l?\} \\
turn\_on' = turn\_on \\
last\_state' = current\_state \\
current\_state' = current\_state \setminus \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$$
\begin{array}{|l}
\multicolumn{1}{l}{TurnOff} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_off \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \setminus \{l?\} \\
turn\_on' = turn\_on \\
last\_state' = current\_state \\
current\_state' = current\_state \setminus \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$TurnOff$

$$\Delta DwarfSignal$$
$$l? : LampId$$

$$l? \in turn\_off$$
$$last\_proper\_state' = last\_proper\_state$$
$$turn\_off' = turn\_off \setminus \{l?\}$$
$$turn\_on' = turn\_on$$
$$last\_state' = current\_state$$
$$current\_state' = current\_state \setminus \{l?\}$$
$$desired\_proper\_state' = desired\_proper\_state$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$$
\begin{array}{l}
\textit{TurnOff} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_off \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \setminus \{l?\} \\
turn\_on' = turn\_on \\
last\_state' = current\_state \\
current\_state' = current\_state \setminus \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$$
\begin{array}{l}
TurnOff \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_off \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \setminus \{l?\} \\
turn\_on' = turn\_on \\
last\_state' = current\_state \\
current\_state' = current\_state \setminus \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$$
\begin{array}{|l}
\multicolumn{1}{l}{TurnOff} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_off \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \setminus \{l?\} \\
turn\_on' = turn\_on \\
last\_state' = current\_state \\
current\_state' = current\_state \setminus \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$$
\begin{array}{l}
\textit{TurnOff} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_off \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \setminus \{l?\} \\
turn\_on' = turn\_on \\
last\_state' = current\_state \\
current\_state' = current\_state \setminus \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$$
\begin{array}{l}
\textit{TurnOff} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_off \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \setminus \{l?\} \\
turn\_on' = turn\_on \\
last\_state' = current\_state \\
current\_state' = current\_state \setminus \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$$
\begin{array}{l}
\textit{TurnOff} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_off \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \setminus \{l?\} \\
turn\_on' = turn\_on \\
last\_state' = current\_state \\
current\_state' = current\_state \setminus \{l?\} \\
desired\_proper\_state' = desired\_proper\_state
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$$
\begin{array}{l}
\textit{TurnOff} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_off \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \setminus \{l?\} \\
turn\_on' = turn\_on \\
last\_state' = current\_state \\
current\_state' = current\_state \setminus \{l?\} \\
desired\_proper\_state' = desired\_proper\_state
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$$
\begin{array}{|l}
\multicolumn{1}{l}{TurnOff} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_off \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \setminus \{l?\} \\
turn\_on' = turn\_on \\
last\_state' = current\_state \\
current\_state' = current\_state \setminus \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ off.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{TurnOff}} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_off \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \setminus \{l?\} \\
turn\_on' = turn\_on \\
last\_state' = current\_state \\
current\_state' = current\_state \setminus \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$$
\begin{array}{l}
\textbf{TurnOn} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$TurnOn$

$$
\begin{array}{|l}
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$TurnOn$

$$
\begin{array}{l}
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$$
\begin{array}{l}
\underline{\quad TurnOn \quad} \\
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$$
\begin{array}{l}
\textit{TurnOn} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$$
\begin{array}{l}
TurnOn \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\_\_ TurnOn \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$$
\begin{array}{l}
\textit{TurnOn} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$$
\begin{array}{l}
\underline{\quad TurnOn \quad} \\
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$$
\begin{array}{l}
\textit{TurnOn} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$$
\begin{array}{l}
\textit{TurnOn} \\
\hline
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$$
\begin{array}{l}
\underline{\quad TurnOn \quad} \\
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state
\end{array}
$$

Turning off a Lamp

- Operation on the state $DwarfSignal$ with one input: $l? : LampId$.
- Precondition: we must have a requirement to turn $l?$ on.

$$
\begin{array}{|l}
\hline
\textit{TurnOn} \\
\Delta DwarfSignal \\
l? : LampId \\
\hline
l? \in turn\_on \\
last\_proper\_state' = last\_proper\_state \\
turn\_off' = turn\_off \\
turn\_on' = turn\_on \setminus \{l?\} \\
last\_state' = current\_state \\
current\_state' = current\_state \cup \{l?\} \\
desired\_proper\_state' = desired\_proper\_state \\
\hline
\end{array}
$$

Overview

Use Case: $Init$

Use Case: $Init$

$Init$

Use Case: $Init$

$$\begin{array}{|l}\hline \textit{Init} \\ \textit{DwarfSignal}' \\ \hline \\ \hline\end{array}$$

Use Case: Init

$$
\begin{array}{|l}
\hline
\textit{Init} \\
\mathit{DwarfSignal}' \\
\hline
\mathit{last\_proper\_state}' = \mathit{stop} \\
\mathit{turn\_off}' = \emptyset \\
\mathit{turn\_on}' = \emptyset \\
\mathit{last\_state}' = \mathit{stop} \\
\mathit{current\_state}' = \mathit{stop} \\
\mathit{desired\_proper\_state}' = \mathit{stop} \\
\hline
\end{array}
$$

Use Case: $Init$

$$
\begin{array}{|l}
\hline
\textit{Init} \\
\textit{DwarfSignal}' \\
\hline
\textit{last\_proper\_state}' = \textit{stop} \\
\textit{turn\_off}' = \emptyset \\
\textit{turn\_on}' = \emptyset \\
\textit{last\_state}' = \textit{stop} \\
\textit{current\_state}' = \textit{stop} \\
\textit{desired\_proper\_state}' = \textit{stop} \\
\hline
\end{array}
$$

After $Init$, the state looks like this:

Use Case: $Init$

$$
\begin{array}{l}
\underline{\quad Init \quad\qquad\qquad\qquad\qquad\qquad\qquad} \\
DwarfSignal' \\
\hline
last\_proper\_state' = stop \\
turn\_off' = \emptyset \\
turn\_on' = \emptyset \\
last\_state' = stop \\
current\_state' = stop \\
desired\_proper\_state' = stop \\
\hline
\end{array}
$$

After $Init$, the state looks like this:

$$
\begin{aligned}
last\_proper\_state &= stop \\
turn\_off &= \emptyset \\
turn\_on &= \emptyset \\
last\_state &= stop \\
current\_state &= stop \\
desired\_proper\_state &= stop
\end{aligned}
$$

Use Case: $Init \mathbin{;} SetNewProperState$

$\mathbin{;}$

Use Case: $Init \fatsemi SetNewProperState$

$$\left\{\begin{array}{rcl} last\_proper\_state' &=& stop \\ turn\_off' &=& \emptyset \\ turn\_on' &=& \emptyset \\ last\_state' &=& stop \\ current\_state' &=& stop \\ desired\_proper\_state' &=& stop \end{array}\right\} \fatsemi$$

Use Case: $Init \fatsemi SetNewProperState$

$$\left\{ \begin{array}{rcl} last\_proper\_state' & = & stop \\ turn\_off' & = & \emptyset \\ turn\_on' & = & \emptyset \\ last\_state' & = & stop \\ current\_state' & = & stop \\ desired\_proper\_state' & = & stop \end{array} \right\} \fatsemi$$

$SetNewProperState$

$\Delta DwarfSignal$
$st? : ProperState$

$current\_state = desired\_proper\_state$
$st? \neq current\_state$
$last\_proper\_state' = current\_state$
$turn\_off' = current\_state \setminus st?$
$turn\_on' = st? \setminus current\_state$
$last\_state' = current\_state$
$current\_state' = current\_state$
$desired\_proper\_state' = st?$

Use Case: $Init \fatsemi SetNewProperState$

$$\left\{ \begin{array}{rcl} last\_proper\_state' & = & stop \\ turn\_off' & = & \emptyset \\ turn\_on' & = & \emptyset \\ last\_state' & = & stop \\ current\_state' & = & stop \\ desired\_proper\_state' & = & stop \end{array} \right\} \fatsemi$$

$SetNewProperState$

$\Delta DwarfSignal$
$st? : ProperState$

$current\_state = desired\_proper\_state$
$warning \neq current\_state$
$last\_proper\_state' = current\_state$
$turn\_off' = current\_state \setminus warning$
$turn\_on' = warning \setminus current\_state$
$last\_state' = current\_state$
$current\_state' = current\_state$
$desired\_proper\_state' = warning$

Use Case: $Init \mathbin{\text{;}} SetNewProperState$

$$
\left\{
\begin{array}{rcl}
last\_proper\_state' & = & stop \\
turn\_off' & = & \emptyset \\
turn\_on' & = & \emptyset \\
last\_state' & = & stop \\
current\_state' & = & stop \\
desired\_proper\_state' & = & stop
\end{array}
\right\} \mathbin{\text{;}}
$$

SetNewProperState

$\Delta DwarfSignal$
$st? : ProperState$

$stop = desired\_proper\_state$
$warning \neq stop$
$last\_proper\_state' = stop$
$turn\_off' = stop \setminus warning$
$turn\_on' = warning \setminus stop$
$last\_state' = stop$
$current\_state' = stop$
$desired\_proper\_state' = warning$

Use Case: $Init \mathbin{;} SetNewProperState$

$$\left\{\begin{array}{rcl} last\_proper\_state' &=& stop \\ turn\_off' &=& \emptyset \\ turn\_on' &=& \emptyset \\ last\_state' &=& stop \\ current\_state' &=& stop \\ desired\_proper\_state' &=& stop \end{array}\right\} \mathbin{;}$$

SetNewProperState

$\Delta DwarfSignal$
$st? : ProperState$

$stop = stop$
$warning \neq stop$
$last\_proper\_state' = stop$
$turn\_off' = stop \setminus warning$
$turn\_on' = warning \setminus stop$
$last\_state' = stop$
$current\_state' = stop$
$desired\_proper\_state' = warning$

Use Case: $Init \fatsemi SetNewProperState$

$$
\left\{
\begin{array}{rcl}
last\_proper\_state' & = & stop \\
turn\_off' & = & \emptyset \\
turn\_on' & = & \emptyset \\
last\_state' & = & stop \\
current\_state' & = & stop \\
desired\_proper\_state' & = & stop
\end{array}
\right\} \fatsemi
$$

$$
\begin{array}{l}
\underline{SetNewProperState} \\
\Delta DwarfSignal \\
st? : ProperState \\
\hline
stop = stop \\
warning \neq stop \\
last\_proper\_state' = stop \\
turn\_off' = stop \setminus warning \\
turn\_on' = warning \setminus stop \\
last\_state' = stop \\
current\_state' = stop \\
desired\_proper\_state' = warning
\end{array}
$$

Use Case: $Init \fatsemi SetNewProperState$

$$\left\{\begin{array}{rcl} last\_proper\_state' & = & stop \\ turn\_off' & = & \emptyset \\ turn\_on' & = & \emptyset \\ last\_state' & = & stop \\ current\_state' & = & stop \\ desired\_proper\_state' & = & stop \end{array}\right\} \fatsemi$$

$SetNewProperState$

$\Delta DwarfSignal$

$st? : ProperState$

$true$

$true$

$last\_proper\_state' = stop$

$turn\_off' = stop \setminus warning$

$turn\_on' = warning \setminus stop$

$last\_state' = stop$

$current\_state' = stop$

$desired\_proper\_state' = warning$

Use Case: $Init \fatsemi SetNewProperState$

$$\left\{\begin{array}{rcl} last\_proper\_state' & = & stop \\ turn\_off' & = & \emptyset \\ turn\_on' & = & \emptyset \\ last\_state' & = & stop \\ current\_state' & = & stop \\ desired\_proper\_state' & = & stop \end{array}\right\} \fatsemi$$

$$\begin{array}{|l} \multicolumn{1}{l}{SetNewProperState} \\ \Delta DwarfSignal \\ st? : ProperState \\ \hline last\_proper\_state' = stop \\ turn\_off' = stop \setminus warning \\ turn\_on' = warning \setminus stop \\ last\_state' = stop \\ current\_state' = stop \\ desired\_proper\_state' = warning \\ \hline \end{array}$$

Use Case: $Init \mathbin{\S} SetNewProperState$

$$\left\{ \begin{array}{rcl} last\_proper\_state' & = & stop \\ turn\_off' & = & \emptyset \\ turn\_on' & = & \emptyset \\ last\_state' & = & stop \\ current\_state' & = & stop \\ desired\_proper\_state' & = & stop \end{array} \right\} \mathbin{\S}$$

SetNewProperState

$\Delta DwarfSignal$
$st? : ProperState$

$last\_proper\_state' = stop$
$turn\_off' = \{L1, L2\} \setminus \{L1, L3\}$
$turn\_on' = \{L1, L3\} \setminus \{L1, L2\}$
$last\_state' = stop$
$current\_state' = stop$
$desired\_proper\_state' = warning$

Use Case: $Init \mathbin{\text{;}} SetNewProperState$

$$\left\{\begin{array}{rcl} last\_proper\_state' &=& stop \\ turn\_off' &=& \emptyset \\ turn\_on' &=& \emptyset \\ last\_state' &=& stop \\ current\_state' &=& stop \\ desired\_proper\_state' &=& stop \end{array}\right\} \mathbin{\text{;}}$$

SetNewProperState

$\Delta DwarfSignal$
$st? : ProperState$

$last\_proper\_state' = stop$
$turn\_off' = \{L2\}$
$turn\_on' = \{L3\}$
$last\_state' = stop$
$current\_state' = stop$
$desired\_proper\_state' = warning$

Use Case: Installation then Set to Warning

1. ***Init***:
$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \emptyset \\ turn\_on &= \emptyset \\ last\_state &= stop \\ current\_state &= stop \\ desired\_proper\_state &= stop \end{aligned}$$
2. ***SetNewProperState warning***:
$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= stop \setminus warning \\ &= \{L1, L2\} \setminus \{L1, L3\} \\ &= \{L2\} \\ turn\_on &= warning \setminus stop \\ &= \{L1, L3\} \setminus \{L1, L2\} \\ &= \{L3\} \\ last\_state &= stop \\ current\_state &= stop \\ desired\_proper\_state &= warning \end{aligned}$$
3. ***TurnOff L2***:
$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \{L2\} \setminus \{L2\} \\ &= \emptyset \\ turn\_on &= \{L3\} \\ last\_state &= stop \\ current\_state &= stop \setminus \{L2\} \\ &= \{L1, L2\} \setminus \{L2\} \\ &= \{L1\} \\ desired\_proper\_state &= warning \end{aligned}$$
4. ***TurnOn L3***:
$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \emptyset \\ turn\_on &= \{L3\} \setminus \{L3\} \\ &= \emptyset \\ last\_state &= \{L1\} \\ current\_state &= \{L1\} \cup \{L3\} \\ &= \{L1, L3\} \\ &= warning \\ desired\_proper\_state &= warning \end{aligned}$$

Use Case: Installation then Set to Warning

1. ***Init***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \emptyset \\ turn\_on &= \emptyset \\ last\_state &= stop \\ current\_state &= stop \\ desired\_proper\_state &= stop \end{aligned}$$

2. ***SetNewProperState warning***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= stop \setminus warning \\ &= \{L1, L2\} \setminus \{L1, L3\} \\ &= \{L2\} \\ turn\_on &= warning \setminus stop \\ &= \{L1, L3\} \setminus \{L1, L2\} \\ &= \{L3\} \\ last\_state &= stop \\ current\_state &= stop \\ desired\_proper\_state &= warning \end{aligned}$$

3. ***TurnOff L2***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \{L2\} \setminus \{L2\} \\ &= \emptyset \\ turn\_on &= \{L3\} \\ last\_state &= stop \\ current\_state &= stop \setminus \{L2\} \\ &= \{L1, L2\} \setminus \{L2\} \\ &= \{L1\} \\ desired\_proper\_state &= warning \end{aligned}$$

4. ***TurnOn L3***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \emptyset \\ turn\_on &= \{L3\} \setminus \{L3\} \\ &= \emptyset \\ last\_state &= \{L1\} \\ current\_state &= \{L1\} \cup \{L3\} \\ &= \{L1, L3\} \\ &= warning \\ desired\_proper\_state &= warning \end{aligned}$$

Use Case: Installation then Set to Warning

1. ***Init***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \emptyset \\ turn\_on &= \emptyset \\ last\_state &= stop \\ current\_state &= stop \\ desired\_proper\_state &= stop \end{aligned}$$

2. ***SetNewProperState warning***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= stop \setminus warning \\ &= \{L1, L2\} \setminus \{L1, L3\} \\ &= \{L2\} \\ turn\_on &= warning \setminus stop \\ &= \{L1, L3\} \setminus \{L1, L2\} \\ &= \{L3\} \\ last\_state &= stop \\ current\_state &= stop \\ desired\_proper\_state &= warning \end{aligned}$$

3. ***TurnOff L2***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \{L2\} \setminus \{L2\} \\ &= \emptyset \\ turn\_on &= \{L3\} \\ last\_state &= stop \\ current\_state &= stop \setminus \{L2\} \\ &= \{L1, L2\} \setminus \{L2\} \\ &= \{L1\} \\ desired\_proper\_state &= warning \end{aligned}$$

4. ***TurnOn L3***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \emptyset \\ turn\_on &= \{L3\} \setminus \{L3\} \\ &= \emptyset \\ last\_state &= \{L1\} \\ current\_state &= \{L1\} \cup \{L3\} \\ &= \{L1, L3\} \\ &= warning \\ desired\_proper\_state &= warning \end{aligned}$$

Use Case: Installation then Set to Warning

1. ***Init***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \emptyset \\ turn\_on &= \emptyset \\ last\_state &= stop \\ current\_state &= stop \\ desired\_proper\_state &= stop \end{aligned}$$

2. ***SetNewProperState warning***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= stop \setminus warning \\ &= \{L1, L2\} \setminus \{L1, L3\} \\ &= \{L2\} \\ turn\_on &= warning \setminus stop \\ &= \{L1, L3\} \setminus \{L1, L2\} \\ &= \{L3\} \\ last\_state &= stop \\ current\_state &= stop \\ desired\_proper\_state &= warning \end{aligned}$$

3. ***TurnOff L2***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \{L2\} \setminus \{L2\} \\ &= \emptyset \\ turn\_on &= \{L3\} \\ last\_state &= stop \\ current\_state &= stop \setminus \{L2\} \\ &= \{L1, L2\} \setminus \{L2\} \\ &= \{L1\} \\ desired\_proper\_state &= warning \end{aligned}$$

4. ***TurnOn L3***:

$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \emptyset \\ turn\_on &= \{L3\} \setminus \{L3\} \\ &= \emptyset \\ last\_state &= \{L1\} \\ current\_state &= \{L1\} \cup \{L3\} \\ &= \{L1, L3\} \\ &= warning \\ desired\_proper\_state &= warning \end{aligned}$$

Use Case: Installation then Set to Warning

1. ***Init***:
$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \emptyset \\ turn\_on &= \emptyset \\ last\_state &= stop \\ current\_state &= stop \\ desired\_proper\_state &= stop \end{aligned}$$
2. ***SetNewProperState warning***:
$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= stop \setminus warning \\ &= \{L1, L2\} \setminus \{L1, L3\} \\ &= \{L2\} \\ turn\_on &= warning \setminus stop \\ &= \{L1, L3\} \setminus \{L1, L2\} \\ &= \{L3\} \\ last\_state &= stop \\ current\_state &= stop \\ desired\_proper\_state &= warning \end{aligned}$$
3. ***TurnOff L2***:
$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \{L2\} \setminus \{L2\} \\ &= \emptyset \\ turn\_on &= \{L3\} \\ last\_state &= stop \\ current\_state &= stop \setminus \{L2\} \\ &= \{L1, L2\} \setminus \{L2\} \\ &= \{L1\} \\ desired\_proper\_state &= warning \end{aligned}$$
4. ***TurnOn L3***:
$$\begin{aligned} last\_proper\_state &= stop \\ turn\_off &= \emptyset \\ turn\_on &= \{L3\} \setminus \{L3\} \\ &= \emptyset \\ last\_state &= \{L1\} \\ current\_state &= \{L1\} \cup \{L3\} \\ &= \{L1, L3\} \\ &= warning \\ desired\_proper\_state &= warning \end{aligned}$$

Conclusion

- Described a real-world railway signalling device: the dwarf signal.
- Specified the behaviour in the Z notation, starting with the state.
- Formalised the safety requirements.
- Specified the user interface: one operation plus two daemon operations.
- Described a use case.
- Final lecture: how to automate the specification animation.

Conclusion

- Described a real-world railway signalling device: the dwarf signal.
- Specified the behaviour in the Z notation, starting with the state.
- Formalised the safety requirements.
- Specified the user interface: one operation plus two daemon operations.
- Described a use case.
- Final lecture: how to automate the specification animation.

Conclusion

- Described a real-world railway signalling device: the dwarf signal.
- Specified the behaviour in the Z notation, starting with the state.
- Formalised the safety requirements.
- Specified the user interface: one operation plus two daemon operations.
- Described a use case.
- Final lecture: how to automate the specification animation.

Conclusion

- Described a real-world railway signalling device: the dwarf signal.
- Specified the behaviour in the Z notation, starting with the state.
- Formalised the safety requirements.
- Specified the user interface: one operation plus two daemon operations.
- Described a use case.
- Final lecture: how to automate the specification animation.

Conclusion

- Described a real-world railway signalling device: the dwarf signal.
- Specified the behaviour in the Z notation, starting with the state.
- Formalised the safety requirements.
- Specified the user interface: one operation plus two daemon operations.
- Described a use case.
- Final lecture: how to automate the specification animation.

Conclusion

- Described a real-world railway signalling device: the dwarf signal.
- Specified the behaviour in the Z notation, starting with the state.
- Formalised the safety requirements.
- Specified the user interface: one operation plus two daemon operations.
- Described a use case.
- Final lecture: how to automate the specification animation.

Conclusion

- Described a real-world railway signalling device: the dwarf signal.
- Specified the behaviour in the Z notation, starting with the state.
- Formalised the safety requirements.
- Specified the user interface: one operation plus two daemon operations.
- Described a use case.
- Final lecture: how to automate the specification animation.