Justification of the Reduction

The reduction is based on the observation that a state-sequence σ satisfies the compassion requirement (p_i, q_i) if either σ contains only finitely many p_i-states, or it contains infinitely many q_i-states.

The boolean variable $nevermore_i$ is intended to be set to 1 at a point, beyond which, there will be no further p_i-states. Thus, $nevermore_i$ predicts the absence of p_i-states. If this prediction is correct, then the newly introduced justice requirement $nevermore_i \lor q_i$ is equivalent to the original compassion requirement.

In the revised FDS $\mathcal{D}_{\mathcal{J}}$, the prediction by $nevermore_i$ is implemented as a non-deterministic assignment of 1 to $nevermore_i$. Therefore, the correctness of the prediction cannot be guaranteed.

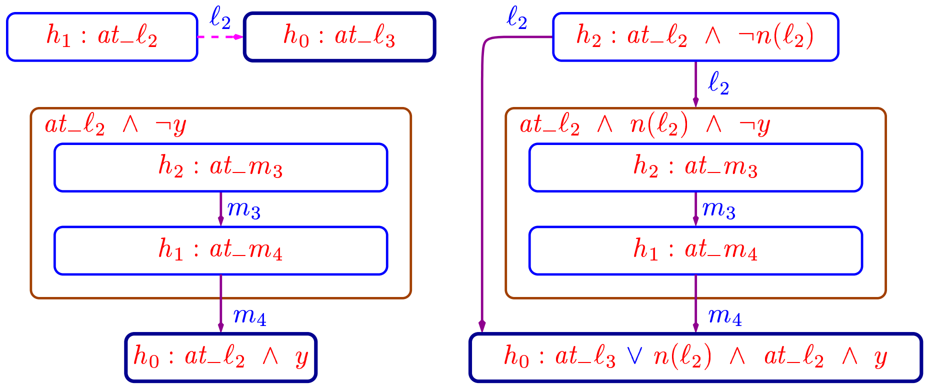
To counter this difficulty, we modify the response property which we aim to prove. The revised property claims that any φ-state in which no mis-prediction has been detected yet, must be followed by a goal state which, either satisfies ψ, or detects a mis-prediction. Mis-prediction is identified as a state in which $nevermore_i$ and p_i are both true.

Comparing General Rule RESP to the *nevermore* Reduction

y: **natural initially** $y = 1$

$P_1 ::$ [ℓ_0: **loop forever do** [ℓ_1: **Non-critical**; ℓ_2: **request** y; ℓ_3: **Critical**; ℓ_4: **release** y]] $\parallel$ $P_2 ::$ [m_0: **loop forever do** [m_1: **Non-critical**; m_2: **request** y; m_3: **Critical**; m_4: **release** y]]

Following are verification diagrams for the two approaches:



Example: MUX-SEM

Reconsider program MUX-SEM:

$$
\begin{array}{lll}
\textbf{in} & N & : \textbf{integer where } N > 1 \\
\textbf{local} & y & : \{0,1\} \textbf{ where } y = 1 \\
\overset{N}{\underset{p=1}{\|}} P[p] :: & & \left[\begin{array}{l} \ell_0: \textbf{loop forever do} \\ \quad \left[\begin{array}{ll} \ell_1: & \textbf{noncritical} \\ \ell_2: & \textbf{request } y \\ \ell_3: & \textbf{critical} \\ \ell_4: & \textbf{release } y \end{array}\right] \end{array}\right]
\end{array}
$$

For which we wish to prove the response property

$$at\_\ell_2[z] \Rightarrow \Diamond\, at\_\ell_3[z]$$

We start by establishing the following invariants:

$$
\begin{array}{lll}
\varphi_1: & \forall i: at\_\ell_{3,4}[i] \;\rightarrow\; y = 0 & \\
\varphi_2: & \forall i \neq j: at\_\ell_{0..2}[i] \;\vee\; at\_\ell_{0..2}[j] & \text{—— Mutual Exclusion} \\
\varphi_3: & y = 0 \;\rightarrow\; \exists i: at\_\ell_{3,4}[i] &
\end{array}
$$

MUX-SEM Continued

Applying the compassion→justice reduction, we introduce the boolean variables $n[i]$, $i = 1, \ldots, N$ (abbreviations for $nevermore[i]$). The added justice requirements are $J_2[i] : n[i] \vee \neg at\_\ell_2[i]$. The mis-prediction predicate is given by:

$$misprediction: \bigvee_{i=1}^{N} at\_\ell_2[i] \;\wedge\; y \;\wedge\; n[i]$$

The helpful justice requirements for this proof are $J_2[z]$ and $\{J_{3,4}[i] \mid i \in [1..N]\}$. The helpful conditions and ranking functions for these transitions are given in the following table:

Id. p	Requirement	$h(p)$	$\delta(p)$
$J_2[z]$	$n[z] \vee \neg at\_\ell_2[z]$	$at\_\ell_2[z] \wedge \neg n[z]$	$\neg n[z]$
$J_3[i]$	$\neg at\_\ell_3[i]$	$at\_\ell_2[z] \wedge n[z] \wedge at\_\ell_3[i]$	$\neg n[z] \vee at\_\ell_3[i]$
$J_4[i]$	$\neg at\_\ell_4[i]$	$at\_\ell_2[z] \wedge n[z] \wedge at\_\ell_4[i]$	$\neg n[z] \vee at\_\ell_{3,4}[i]$

The ranking functions range over the domain $\{0,1\}$. The assertion $\delta(p)$ is true at a state if the corresponding ranking of $J(p)$ is 1. Usually, this is the case if requirement p may still become helpful. If $\delta(p)$ is false, then the corresponding ranking is 0.

Example: Dining Philosophers with One Contrary Philosopher

local f : **array** $[1..n]$ **of natural initially** $f = 1$

$\overset{n-1}{\underset{j=1}{\|}} P[j] ::$
$\left[\begin{array}{l} \ell_0 : \textbf{loop forever do} \\ \quad \left[\begin{array}{ll} \ell_1 : & \textbf{Non-Critical} \\ \ell_2 : & \textbf{request } f[j] \\ \ell_3 : & \textbf{request } f[j{+}1] \\ \ell_4 : & \textbf{Critical} \\ \ell_5 : & \textbf{release } f[j] \\ \ell_6 : & \textbf{release } f[j{+}1] \end{array}\right] \end{array}\right]$

$\|$

$P[n] ::$
$\left[\begin{array}{l} \ell_0 : \textbf{loop forever do} \\ \quad \left[\begin{array}{ll} \ell_1 : & \textbf{Non-Critical} \\ \ell_2 : & \textbf{request } f[1] \\ \ell_3 : & \textbf{request } f[n] \\ \ell_4 : & \textbf{Critical} \\ \ell_5 : & \textbf{release } f[1] \\ \ell_6 : & \textbf{release } f[n] \end{array}\right] \end{array}\right]$

We wish to establish part of accessibility, expressible by

$$\psi_{acc}: \quad \square\ (at\_\ell_3[z] \ \rightarrow\ \diamondsuit\ (at\_\ell_4[z]))$$

for $z \in [2..N-1]$.

Dining Philosophers Continued

Applying the compassion→justice reduction, we introduce two arrays of *nevermore* variables, $n_2[i]$ and $n_3[i]$ corresponding to locations $\ell_2[i]$ and $\ell_3[i]$.

The helpful justice requirements are $J_{3..6}[z-1]$, $J_{2,3}[z]$, $\{J_{3..5}[i] \mid i \in [z{+}1..N{-}1]\}$ and $J_{4..6}[N]$. The helpful conditions for these transitions are given in the following table:

Id. p	$h(p)$
$J_4[z-1]$	$at\_\ell_4[z-1] \ \wedge\ at\_\ell_2[z] \ \wedge\ n_2[z]$
$J_5[z-1]$	$at\_\ell_5[z-1] \ \wedge\ at\_\ell_2[z] \ \wedge\ n_2[z]$
$J_6[z-1]$	$at\_\ell_6[z-1] \ \wedge\ at\_\ell_2[z] \ \wedge\ n_2[z]$
$J_2[z]$	$at\_\ell_2[z] \ \wedge\ \neg n_2[z]$
$J_3[z]$	$at\_\ell_3[z] \ \wedge\ \neg n_3[z]$
$J_3[i] : i \in [z{+}1..N{-}1]$	$at\_\ell_3[z] \ \wedge\ at\_\ell_3[i] \ \wedge\ \neg n_3[i] \ \wedge\ at\_\ell_3[i-1] \ \wedge\ n_3[i-1]$
$J_4[i] : i \in [z{+}1..N{-}1]$	$at\_\ell_3[z] \ \wedge\ at\_\ell_4[i] \ \wedge\ at\_\ell_3[i-1] \ \wedge\ n_3[i-1]$
$J_5[i] : i \in [z{+}1..N{-}1]$	$at\_\ell_3[z] \ \wedge\ at\_\ell_5[i] \ \wedge\ at\_\ell_3[i-1] \ \wedge\ n_3[i-1]$
$J_4[N]$	$at\_\ell_3[z] \ \wedge\ at\_\ell_4[N] \ \wedge\ at\_\ell_3[N-1] \ \wedge\ n_3[N-1]$
$J_5[N]$	$at\_\ell_3[z] \ \wedge\ at\_\ell_5[N] \ \wedge\ at\_\ell_3[N-1] \ \wedge\ n_3[N-1]$
$J_6[N]$	$at\_\ell_3[z] \ \wedge\ at\_\ell_6[N] \ \wedge\ at\_\ell_3[N-1] \ \wedge\ n_3[N-1]$

Dining Philosophers: Ranking Functions

The following table presents the distributed ranking functions $\delta(p)$ for each of the helpful requirements $J(p)$. The ranking functions range over $\{0,1\}$, and the assertion $\delta(p)$ tells us when the ranking of $J(p)$ is 1.

$\delta_4[z{-}1]$	$n_2[z] \rightarrow at\_\ell_{0..4}[z{-}1]$
$\delta_5[z{-}1]$	$n_2[z] \rightarrow at\_\ell_{0..5}[z{-}1]$
$\delta_6[z{-}1]$	1
$\delta_2[z]$	$at\_\ell_2[z] \;\wedge\; \neg n_2[z]$
$\delta_3[z]$	$\neg n_3[z]$
$\delta_3[i] : i \in [z{+}1..N{-}1]$	$\neg n_3[i] \;\wedge\; (at\_\ell_3[i{-}1] \;\wedge\; n_3[i{-}1] \rightarrow at\_\ell_{0..3,6}[i])$
$\delta_4[i] : i \in [z{+}1..N{-}1]$	$at\_\ell_3[i{-}1] \;\wedge\; n_3[i{-}1] \rightarrow at\_\ell_{0..4,6}[i]$
$\delta_5[i] : i \in [z{+}1..N{-}1]$	1
$\delta_4[N]$	$at\_\ell_3[N{-}1] \;\wedge\; n_3[N{-}1] \rightarrow at\_\ell_{0..4}[N]$
$\delta_5[N]$	$at\_\ell_3[N{-}1] \;\wedge\; n_3[N{-}1] \rightarrow at\_\ell_{0..5}[N]$
$\delta_6[N]$	1

Assignment 1. *Draw a verification diagram for the proof of accessibility for the dining-philosophers system.*

The Centralized vs. Distributed Versions of Rule Well

The premises of the centralized version are:

$$D1. \quad p \quad \rightarrow \quad \bigvee_{j=0}^{m} h_j$$

$$D2. \quad h_i \;\wedge\; \rho \quad \rightarrow \quad (h_i' \;\wedge\; \delta_i = \delta_i' \;\wedge\; \neg J_i') \;\vee\; \left(\bigvee_{j=0}^{m} h_j' \;\wedge\; \delta_i \succ \delta_j'\right)$$

In the distributed version, premise $D2$ is replaced by:

$$D2. \quad h_i \;\wedge\; \rho \quad \rightarrow \quad (h_i' \;\wedge\; \neg J_i') \;\vee\; \left(\delta_i > \delta_i' \;\wedge\; \bigvee_{j=0}^{m} h_j'\right)$$

$$D3. \quad h_i \;\wedge\; \rho \quad \rightarrow \quad q' \;\vee\; \delta_j \geq \delta_j'$$

Thus, in both versions, we have to identify for each requirement J_i, the helpful assertion h_i characterizing the states at which progress is guaranteed by satsisfaction of J_i.

The versions differ in the type of the ranking functions δ_i and the heuristics for their identification.

Identifying the Ranking Functions

In the centralized version, the ranking functions are determined to identify global progress. They often are lexicographic tuples of well-founded domains.

For the distributed version, the ranking functions are often binary (range over $\{0,1\}$). We can represent them by an assertion δ_i true whenever $\delta_i = 1$. There are essentially to heuristics for determining δ.

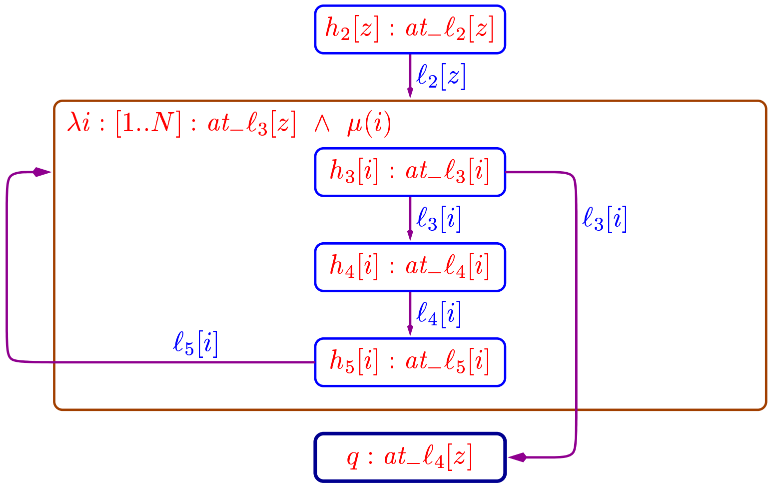
- δ_i should characterize the states from which J_i may still become helpful.
- $\neg\delta_i$ should characterize all J_i-states immediately following an h_i-state, and their descendants. More generally, $\neg\delta_i$ should characterize all states at which J_i is not helpful and can never become helpful in the future.

We will illusrtate this on two of our running examples.

The BAKERY Algorithm

y : **array**$[1..N]$ **of natural where** $y = 0$

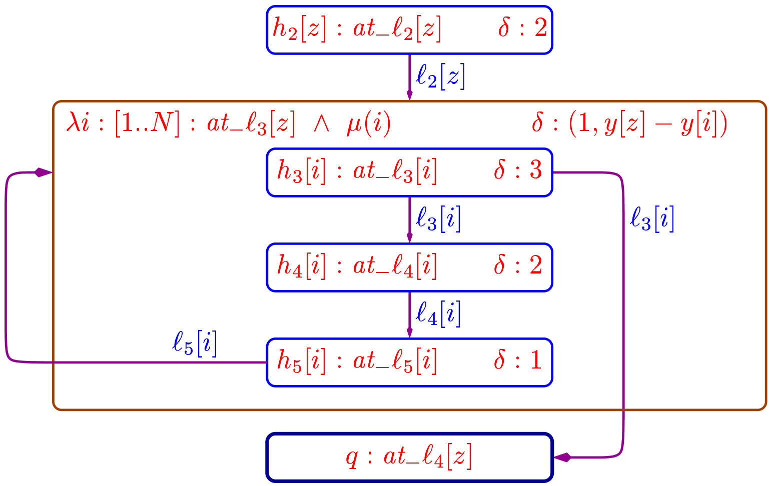
$\overset{N}{\underset{i=1}{\|}} P[i] ::$
ℓ_0: **loop forever do**
- ℓ_1: **Non-critical**
- ℓ_2: $y[i] := \max(y[1], \ldots, y[N]) + 1$
- ℓ_3: **await** $\underbrace{\forall j \neq i : y[j] = 0 \ \vee\ y[i] < y[j]}_{\mu(i)}$
- ℓ_4: **Critical**
- ℓ_5: $y[i] := 0$



With Centralized Ranking Functions

y : **array**$[1..N]$ **of natural where** $y = 0$

$$\overset{N}{\underset{i=1}{\|}} P[i] :: \left[\begin{array}{l} \ell_0\text{: } \textbf{loop forever do} \\ \quad \left[\begin{array}{ll} \ell_1\text{:} & \textbf{Non-critical} \\ \ell_2\text{:} & y[i] := \max(y[1], \ldots, y[N]) + 1 \\ \ell_3\text{:} & \textbf{await } \underbrace{\forall j \neq i : y[j] = 0 \ \vee \ y[i] < y[j]}_{\mu(i)} \\ \ell_4\text{:} & \textbf{Critical} \\ \ell_5\text{:} & y[i] := 0 \end{array}\right] \end{array}\right]$$



With Distributed Ranking Functions

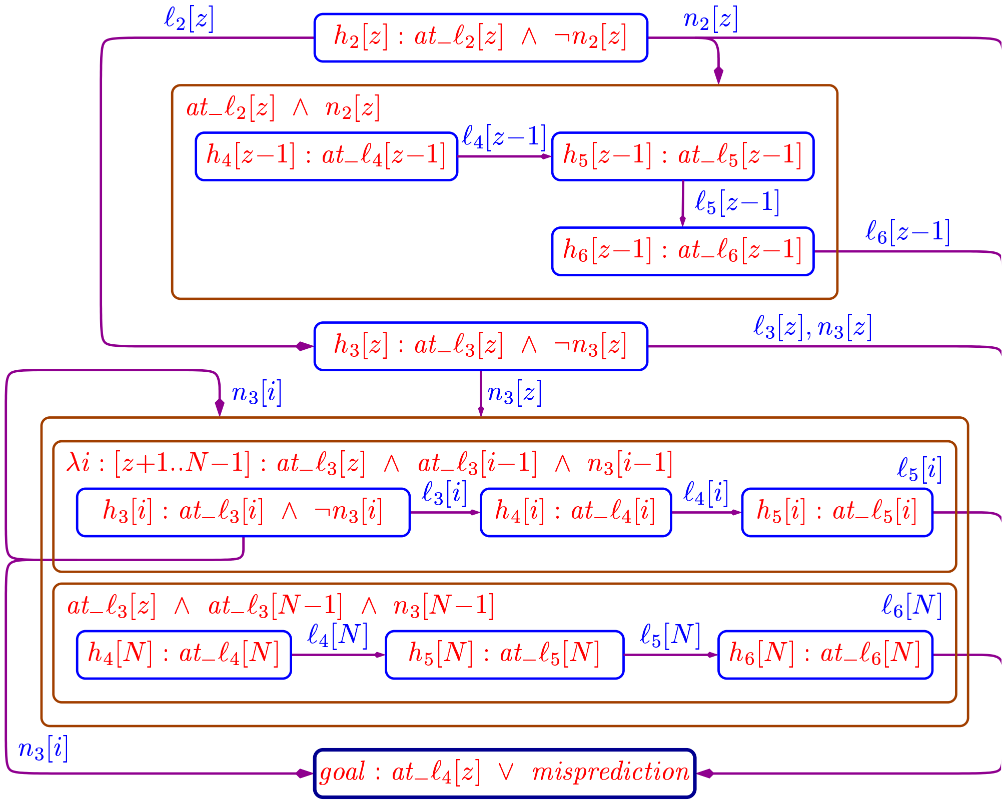
y : **array**$[1..N]$ **of natural where** $y = 0$

$$\overset{N}{\underset{i=1}{\|}} P[i] :: \left[\begin{array}{l} \ell_0\text{: } \textbf{loop forever do} \\ \quad \left[\begin{array}{ll} \ell_1\text{:} & \textbf{Non-critical} \\ \ell_2\text{:} & y[i] := \max(y[1], \ldots, y[N]) + 1 \\ \ell_3\text{:} & \textbf{await } \underbrace{\forall j \neq i : y[j] = 0 \ \vee \ y[i] < y[j]}_{\mu(i)} \\ \ell_4\text{:} & \textbf{Critical} \\ \ell_5\text{:} & y[i] := 0 \end{array}\right] \end{array}\right]$$

The ranking functions are given by:

$\delta_2[z]$:	$at\_\ell_2[z]$
$\delta_3[i]$:	$at\_\ell_2[z] \quad \vee \quad at\_\ell_3[z] \ \wedge \ y[i] \leq y[z] \ \wedge \ at\_\ell_3[i]$
$\delta_4[i]$:	$at\_\ell_2[z] \quad \vee \quad at\_\ell_3[z] \ \wedge \ y[i] \leq y[z] \ \wedge \ at\_\ell_{3,4}[i]$
$\delta_5[i]$:	$at\_\ell_2[z] \quad \vee \quad at\_\ell_3[z] \ \wedge \ y[i] \leq y[z] \ \wedge \ at\_\ell_{3..5}[i]$

Verification Diagram for Dining



Distributed Ranking Functions

The useful heuristic here is to identify $\neg h_i$-states following an h_i-state:

$\delta_4[z-1]$	$\neg(n_2[z] \wedge at\_\ell_{5,6}[z-1])$
$\delta_5[z-1]$	$\neg(n_2[z] \wedge at\_\ell_6[z-1])$
$\delta_6[z-1]$	1
$\delta_2[z]$	$at\_\ell_2[z] \wedge \neg n_2[z]$
$\delta_3[z]$	$\neg n_3[z]$
$\delta_3[i] : i \in [z+1..N-1]$	$\neg n_3[i] \wedge \neg(at\_\ell_3[i-1] \wedge n_3[i-1] \wedge at\_\ell_{4,5}[i])$
$\delta_4[i] : i \in [z+1..N-1]$	$\neg(at\_\ell_3[i-1] \wedge n_3[i-1] \wedge at\_\ell_5[i])$
$\delta_5[i] : i \in [z+1..N-1]$	1
$\delta_4[N]$	$\neg(at\_\ell_3[N-1] \wedge n_3[N-1] \wedge at\_\ell_{5,6}[N])$
$\delta_5[N]$	$\neg(at\_\ell_3[N-1] \wedge n_3[N-1] \wedge at\_\ell_6[N])$
$\delta_6[N]$	1

A centralized ranking function can be obtained by counting how many requirements J_i currently satisfy δ_i.