

Z 言語による仕様——例題 3

染 谷 誠

1. 設計の方針

- ・実際の昇降機の制御系を実現することは断念し、リフト運行の模擬系を設計する。
- ・設計案は状態遷移のトレースモデルによって提示する。
- ・Z 言語は状態遷移モデルを記述するための言語で、状態遷移のトレースモデルを記述するには適していない。そのため Z 言語を少々拡張した。
- ・トレースモデルの記述には CSP などの記述の枠組みが適しており、それを推奨するのが当方の基本的立場だが、ここでは、記述実験の意味で、Z 言語を自然に拡張してトレースモデルを記述することを試みた。この拡張 Z 言語を、分散協調型の仕様言語として広く推奨する意図はないが、この拡張は、Z 言語の使い勝手の向上に寄与すると信ずる。

2. 課題の概要

「課題の提示」の「コンピュータ制御エレベータ」に付随して、非形式的な課題説明がある。乞う参照。同説明文中にある緊急処理に関する部分は無視する。

仕様の最初の版では、フロアからのリフトの呼び出しボタンの操作には誤りがない、との前提で設計した。実際には、フロアからの呼び出しボタンの操作には誤りがあり得るが、おそらくは課題を単純化するため、課題説明中にも誤操作には言及されていない。しかし、実装の途上で、呼び出し要求を適当に見捨てることにより、誤操作に対処した。

3. 基本データ

$$\left| \begin{array}{l} \text{groundFloorNo, topFloorNo} : \mathbb{N} \\ \hline 0 < \text{groundFloorNo} \\ \text{groundFloorNo} < \text{topFloorNo} \end{array} \right.$$

$FLOORNO ::= \text{groundFloorNo}..topFloorNo$
[LIFTID]

$SIGNAL ::= \text{pushLiftButtonSig}\langle LIFTID \times FLOORNO \rangle$
| $\text{pushUpButtonSig}\langle FLOORNO \rangle$
| $\text{pushDownButtonSig}\langle FLOORNO \rangle$
| $\text{arrivingSig}\langle LIFTID \rangle$
| $\text{canLeaveSig}\langle LIFTID \rangle$

$COMKIND ::= \text{leaveUp} \mid \text{arriveUp} \mid \text{leaveDown} \mid \text{arriveDown} \mid \text{noOperation}$
 $COMMAND ::= \text{noCommand} \mid \text{liftCommand}\langle LIFTID \times COMKIND \rangle$

信号の説明

- $pushLiftButtonSig(l, f)$ $l: LIFTID, f: FLOORNO$
リフト l のリフトボタンパネルの行先フロア f へのボタンが押下された。
- $pushUpButtonSig(f)$ $f: FLOORNO$
フロア f の上昇ボタンが押下された。
- $pushDownButtonSig(f)$ $f: FLOORNO$
フロア f の下降ボタンが押下された。
- $canLeaveSig(l)$ $l: LIFTID$
リフト l は、ドアが閉まり、現在のフロアから離床可能。
- $arrivingSig(l)$ $l: LIFTID$
リフト l がフロアに接近中。これを受信して、直ちに $arrivUp$ や $arrivDown$ などの到着指令を出せば、そのフロアに到着できるものとする。何も指令を出さなければ、そのフロアは通過する。

リフト制動装置への指令の種類 ($COMKIND$) と指令 ($COMMAND$)

- $leaveUp$
今停止しているフロアから上昇する。
- $arriveUp$
今接近中のフロアへ、上昇しながら到着する。
- $leaveDown$
今停止しているフロアから下降する。
- $arriveDown$
今接近中のフロアへ、下降しながら到着する。
- $noOperation$
何もしない。
- $noCommand$
何も指令しない。
- $liftCommand(l, cid)$ $l: LIFTID, cid: COMKIND$
リフト l に対して cid を指令する。

4. フロア要求とリフト行先要求

リフトの利用者は、その人のいるフロアから、上昇ボタンや下降ボタンを押して、リフトを呼び出す。この呼び出し要求を**フロア要求**という。上昇ボタンを押すと上昇するリフトを呼び出す。これを**上昇要求**という。下降ボタンを押すと下降するリフトを呼び出す。これを**下降要求**という。上昇要求も、下降要求も、それぞれどのフロアのボタンが押されたのか、そのフロア番号によって特徴づけられる。

リフトにはボタンの配列盤がある。配列盤のボタンはフロアに対応する。行先フロアに対応するボタンを押すことによって、行先を指定する。この行先要求を**リフト行先要求**という。リフト要求は、どのフロアへ行こうとするのか、そのフロア番号によって特徴づけられる。

これらの要求の集合を仕様化するために、フロア番号の集合を仕様化する。

4.1 フロアの集合

4.1.1 フロア集合

<i>FloorSet</i> <i>flrset</i> : $\mathbb{F} \text{ FLOORNO}$

4.1.2 フロア集合に関する操作

4.1.2.1 フロア集合・初期化 フロア集合を初期化する。

<i>FloorSetInit</i> <i>FloorSet'</i> <i>flrset'</i> = \emptyset

4.1.2.2 フロア集合・追加 フロア集合に新しいフロア番号 $f?$ を追加する。

<i>FloorSetAdd</i> $\Delta \text{FloorSet}$ $f? : \text{FLOORNO}$ <i>flrset'</i> = $\text{flrset} \cup \{f?\}$

4.1.2.3 フロア集合・削除 指定されたフロア番号 $f?$ がフロア集合に属していれば, $f?$ をその集合から削除する。

<i>FloorSetRemove</i> $\Delta \text{FloorSet}$ $f? : \text{FLOORNO}$ <i>flrset'</i> = $\text{flrset} \setminus \{f?\}$

4.1.2.4 フロア集合・条件付き削除 指定されたフロア番号 $f?$ がフロア集合に属していれば, $f?$ をその集合から削除する。 $f?$ がそのフロア集合に属していなければ何もしない。

<i>FloorSetMayRemove</i> $\Delta \text{FloorSet}$ $f? : \text{FLOORNO}$ $f? \in \text{flrset} \Rightarrow \text{flrset}' = (\text{flrset} \setminus \{f?\})$ $f? \notin \text{flrset} \Rightarrow \text{flrset}' = \text{flrset}$

4.1.3 フロア集合に関する判断

4.1.3.1 $f?$ がフロア集合に含まれている

<i>FloorSetContainsCurrentFlr</i> <i>FloorSet</i> $f? : \text{FLOORNO}$ $f? \in \text{flrset}$

4.1.3.2 $f?$ より上のフロアがフロア集合に含まれている

$FloorSetContainsAboveFlr$
$FloorSet$
$f? : FLOORNO$
$\exists f : FLOORNO \bullet f? < f \wedge f \in flrset$

4.1.3.3 $f?$ より下のフロアがフロア集合に含まれている

$FloorSetContainsBelowFlr$
$FloorSet$
$f? : FLOORNO$
$\exists f : FLOORNO \bullet f < f? \wedge f \in flrset$

4.2 フロアからの上昇要求

4.2.1 上昇要求の集合

最上階からの上昇要求は存在しない。

$UpRequest$
$upreq : FloorSet$
$topFloorNo \notin upreq.flrset$

4.2.2 上昇要求に関する操作

4.2.2.1 操作の枠

$UpRequestOp$
$\Delta UpRequest$
$\Delta FloorSet$
$f? : FLOORNO$
$\theta FloorSet = upreq$
$upreq' = \theta FloorSet'$

4.2.2.2 諸操作

- 上昇要求の初期化
- $f?$ からの上昇要求の追加
- $f?$ からの上昇要求の削除

$$UpRequestInit \hat{=} [UpRequest'; FloorSetInit \mid upreq' = \theta FloorSet'] \\ \setminus (FloorSet')$$

$$UpRequestAdd \hat{=} (UpRequestOp \wedge FloorSetAdd) \setminus (\Delta FloorSet)$$

$$UpRequestRemove \hat{=} (UpRequestOp \wedge FloorSetRemove) \setminus (\Delta FloorSet)$$

4.2.3 上昇要求に関する判断

4.2.3.1 判断の枠

$UpRequestState$
$UpRequest$
$FloorSet$
$\theta FloorSet = upreq$

4.2.3.2 上昇要求に関する諸判断

- $f?$ から上昇要求がある
- $f?$ より上のフロアから上昇要求がある
- $f?$ より下のフロアから上昇要求がある

$$UpRequestFromCurrentFlr \hat{=} (UpRequestState \wedge FloorSetContainsCurrentFlr) \setminus (FloorSet)$$

$$UpRequestFromAboveFlr \hat{=} (UpRequestState \wedge FloorSetContainsAboveFlr) \setminus (FloorSet)$$

$$UpRequestFromBelowFlr \hat{=} (UpRequestState \wedge FloorSetContainsBelowFlr) \setminus (FloorSet)$$

4.3 フロアからの下降要求

4.3.1 下降要求の集合

$DownRequest$
$downreq : FloorSet$
$groundFloorNo \notin downreq.flrset$

4.3.2 下降要求に関する操作

4.3.2.1 操作の枠

$DownRequestOp$
$\Delta DownRequest$
$\Delta FloorSet$
$f? : FLOORNO$
$\theta FloorSet = downreq$
$downreq' = \theta FloorSet'$

4.3.2.2 諸操作

- 下降要求の初期化
- $f?$ からの下降要求の追加
- $f?$ からの下降要求の削除

$$DownRequestInit \hat{=} [DownRequest'; FloorSetInit \mid downreq' = \theta FloorSet'] \setminus (FloorSet')$$

$$DownRequestAdd \hat{=} (DownRequestOp \wedge FloorSetAdd) \setminus (\Delta FloorSet)$$

$$DownRequestRemove \hat{=} (DownRequestOp \wedge FloorSetRemove) \setminus (\Delta FloorSet)$$

4.3.3 下降要求に関する判断

4.3.3.1 判断の枠

$DownRequestState$ $DownRequest$ $FloorSet$
$\theta FloorSet = downreq$

4.3.3.2 下降要求に関する諸判断

- $f?$ から下降要求がある
- $f?$ より上のフロアから下降要求がある
- $f?$ より下のフロアから下降要求がある

$$DownRequestFromCurrentFlr \hat{=} (DownRequestState \wedge FloorSetContainsCurrentFlr) \setminus (FloorSet)$$

$$DownRequestFromAboveFlr \hat{=} (DownRequestState \wedge FloorSetContainsAboveFlr) \setminus (FloorSet)$$

$$DownRequestFromBelowFlr \hat{=} (DownRequestState \wedge FloorSetContainsBelowFlr) \setminus (FloorSet)$$

4.4 リフトの行先要求

リフトにはリフトボタンのパネルがある。利用者は、そのパネルの、フロア番号に対応するリフトボタンを押下することによって、行先フロアを指定する。これをリフトの行先要求という。リフトの行先要求はフロア番号によって特徴づけられる。

4.4.1 リフト行先要求の集合

$LiftReqSet$ $lftreq : FloorSet$

4.4.1.1 操作の枠

$LiftRequestOp$ $\Delta LiftRequest$ $\Delta FloorSet$ $f? : FLOORNO$
$\theta FloorSet = lftreq$ $lftreq' = \theta FloorSet'$

4.4.2 リフト行先要求に関する操作

4.4.2.1 操作の枠

$LiftRequestOp$ $\Delta LiftRequest$ $\Delta FloorSet$ $f? : FLOORNO$
$\theta FloorSet = lftreq$ $lftreq' = \theta FloorSet'$

4.4.2.2 諸操作

- リフト行先要求の初期化
- $f?$ へのリフト行先要求の追加
- $f?$ へのリフト行先要求の削除

$$\text{LiftRequestInit} \hat{=} [\text{LiftRequest}' ; \text{FloorSetInit} \mid \text{lftreq}' = \theta \text{FloorSet}'] \\ \quad \quad \quad \setminus (\text{FloorSet}')$$

$$\text{LiftRequestAdd} \hat{=} (\text{LiftRequestOp} \wedge \text{FloorSetAdd}) \setminus (\Delta \text{FloorSet})$$

$$\text{LiftRequestRemove} \hat{=} (\text{LiftRequestOp} \wedge \text{FloorSetRemove}) \setminus (\Delta \text{FloorSet})$$

4.4.3 リフト行先要求に関する判断

4.4.3.1 判断の枠

LiftRequestState
LiftRequest
FloorSet
$\theta \text{FloorSet} = \text{lftreq}$

4.4.3.2 リフト行先要求に関する諸判断

- $f?$ へのリフト行先要求がある
- $f?$ より上のフロアへのリフト行先要求がある
- $f?$ より下のフロアへのリフト行先要求がある

$$\text{LiftRequestToCurrentFlr} \hat{=} (\text{LiftRequestState} \wedge \text{FloorSetContainsCurrentFlr}) \setminus (\text{FloorSet})$$

$$\text{LiftRequestToAboveFlr} \hat{=} (\text{LiftRequestState} \wedge \text{FloorSetContainsAboveFlr}) \setminus (\text{FloorSet})$$

$$\text{LiftRequestToBelowFlr} \hat{=} (\text{LiftRequestState} \wedge \text{FloorSetContainsBelowFlr}) \setminus (\text{FloorSet})$$

5. フロアボタンとリフトボタン

リフトを呼び出すため、各フロアには、呼び出し用ボタンが配備されている。これをフロアボタンという。

リフト内にも、その行先を指定するため、ボタンが配備されている。これをリフトボタンという。

5.1 ボタン

ボタンは押すと点灯し、やがてその要求が充たされると消灯する。

5.1.1 ボタンの状態

Button
$\text{isLighted} : \text{Boolean}$

5.1.2 ボタンの操作

5.1.2.1 ボタンを初期化する

ButtonInit
$\text{FloorButton}'$
$\text{isLighted}' = \text{false}$

5.1.2.2 ボタンを点燈する

<i>ButtonLightOn</i>
$\Delta FloorButton$
$isLighted' = true$

5.1.2.3 ボタンを消燈する

<i>ButtonLightOff</i>
$\Delta FloorButton$
$isLighted' = false$

5.1.3 ボタンの状態に関する判断

5.1.3.1 ボタンは消燈している

<i>ButtonIsDark</i>
<i>FloorButton</i>
$isLighted = false$

5.1.3.2 ボタンは点燈している

<i>ButtonIsBright</i>
<i>FloorButton</i>
$isLighted = true$

5.2 ボタンの配列盤

各フロアからリフトを呼び出すためのフロアボタンには、上昇するリフトを呼び出す上昇用ボタンと、下降するリフトを呼び出す下降用ボタンがある。それぞれ、各フロアに一つずつ備えられている。

リフトの行先を指定するリフトボタンも、各フロアごとに一つずつ配置されている。このように、各フロアごとに一つずつボタンを配置したものを、ボタンの配列盤とよぶ。

上昇用ボタンは最上階にはない。下降用ボタンは最下階にはない。このため、配列盤は $FLOORNO$ の上の部分関数とする。

5.2.1 ボタンの配列盤

配列盤のボタンは、各フロア ($FLOORNO$) ごとに、高々一つずつ配置されている。

<i>ButtonPanel</i>
$btnpanel : FLOORNO \rightarrow Button$

5.2.2 ボタン配列盤の操作

5.2.2.1 配列盤の初期化

$\begin{array}{l} \text{ButtonPanelInit} \\ \text{ButtonPanel}' \\ \hline \forall f : \text{dom btnpanel} \bullet \text{btnpanel}'(f) = (\mu \text{ButtonInit} \bullet \theta \text{Button}') \end{array}$
--

5.2.2.2 配列盤の中のボタンに関する操作の枠

配列盤の中のあるボタン（フロア $f?$ に対応するボタン）に関する操作の枠。

$\begin{array}{l} \text{ButtonPanelOp} \\ \Delta \text{ButtonPanel} \\ \Delta \text{Button} \\ f? : \text{FLOORNO} \\ \hline f? \in \text{dom btnpanel} \\ \theta \text{Button} = \text{btnpanel}(f?) \\ \text{btnpanel}' = \text{btnpanel}' \oplus \{f? \mapsto \theta \text{Button}'\} \end{array}$

5.2.2.3 配列盤の中のボタンに関する操作

- 配列盤の中のあるボタン ($f?$ に対応する) を点燈する
- 配列盤の中のあるボタン ($f?$ に対応する) を消燈する

$$\text{ItsButtonLightOn} \hat{=} (\text{ButtonPanelOp} \wedge \text{ButtonLightOn}) \setminus (\Delta \text{Button})$$

$$\text{ItsButtonLightOff} \hat{=} (\text{ButtonPanelOp} \wedge \text{ButtonLightOff}) \setminus (\Delta \text{Button})$$

5.2.3 配列盤の中のボタンに関する判断

配列盤の中のあるボタン（フロア $f?$ に対応するボタン）に関する判断の枠。

5.2.3.1 配列盤の中のボタンに関する判断の枠

$\begin{array}{l} \text{ButtonPanelState} \\ \text{ButtonPanel} \\ \text{Button} \\ f? : \text{FLOORNO} \\ \hline f? \in \text{dom btnpanel} \\ \theta \text{Button} = \text{btnpanel}(f?) \end{array}$

5.2.3.2 配列盤の中のボタンに関する判断

- 配列盤の中のあるボタン ($f?$ に対応する) が消えている
- 配列盤の中のあるボタン ($f?$ に対応する) が点いている

$$\text{ItsButtonIsDark} \hat{=} (\text{ButtonPanelState} \wedge \text{ButtonIsDark}) \setminus (\text{Button})$$

$$\text{ItsButtonIsBright} \hat{=} (\text{ButtonPanelState} \wedge \text{ButtonIsBright}) \setminus (\text{Button})$$

5.3 上昇用フロアボタンの配列盤

上昇するリフトを呼び出すためのボタンが各フロアに一つずつ備えられている。ただし最上階にはない。

5.3.1 上昇用フロアボタンの配置

$UpButtonPanel$ $upbtn : ButtonPanel$ $dom\ upbtn.btnpanel = FLOORNO \setminus \{topFloorNo\}$
--

5.3.2 上昇用フロアボタンの配列盤に関する操作

配列盤の中のあるボタン（フロア $f?$ に対応するボタン）に関する操作の枠。

$UpButtonPanelOp$ $\Delta UpButtonPanel$ $\Delta ButtonPanel$ $f? : FLOORNO$
$f? \in dom\ upbtn.btnpanel$ $\theta ButtonPanel = upbtn$ $upbtn' = \theta ButtonPanel'$

- 上昇用フロアボタン配列盤の初期化
- ある上昇用フロア ($f?$) ボタンを点燈する
- ある上昇用フロア ($f?$) ボタンを消燈する

$$UpButtonPanelInit \hat{=} [UpButtonPanel'; ButtonPanelInit \mid upbtn' = \theta ButtonPanel'] \setminus (ButtonPanel')$$

$$ItsUpButtonLightOn \hat{=} (UpButtonPanelOp \wedge ItsButtonLightOn) \setminus (\Delta ButtonPanel)$$

$$ItsUpButtonLightOff \hat{=} (UpButtonPanelOp \wedge ItsButtonLightOff) \setminus (\Delta ButtonPanel)$$

5.3.3 上昇用ボタン配列盤に関する判断

$UpButtonPanelState$ $UpButtonPanel$ $ButtonPanel$ $f? : FLOORNO$
$f? \in dom\ upbtn.btnpanel$ $\theta ButtonPanel = upbtn$

- ある上昇用フロア ($f?$) ボタンが点いている
- ある上昇用フロア ($f?$) ボタンが消えている

$$ItsUpButtonIsBright \hat{=} (UpButtonPanelState \wedge ItsButtonIsBright) \setminus (ButtonPanel)$$

$$ItsUpButtonIsDark \hat{=} (UpButtonPanelState \wedge ItsButtonIsDark) \setminus (ButtonPanel)$$

5.4 下降用フロアボタンの配列盤

下降するリフトを呼び出すためのボタンが各フロアに一つずつ備えられている。ただし最下階にはない。

5.4.1 下降用フロアボタンの配置

$DownButtonPanel$ $downbtn : ButtonPanel$ $dom\ downbtn.btnpanel = FLOORNO \setminus \{groundFloorNo\}$

5.4.2 下降用フロアボタンの配列盤に関する操作

配列盤の中のあるボタン（フロア $f?$ に対応するボタン）に関する操作の枠．

$$\text{DownButtonPanelOp}$$

$$\Delta \text{DownButtonPanel}$$

$$\Delta \text{ButtonPanel}$$

$$f? : \text{FLOORNO}$$

$$f? \in \text{dom downbtn.btnpanel}$$

$$\theta \text{ButtonPanel} = \text{downbtn}$$

$$\text{downbtn}' = \theta \text{ButtonPanel}'$$

- 下降用フロアボタン配列盤の初期化
- ある下降用フロア ($f?$) ボタンを点燈する
- ある下降用フロア ($f?$) ボタンを消燈する

$$\text{DownButtonPanelInit} \hat{=} [\text{DownButtonPanel}'; \text{ButtonPanelInit} \mid \text{downbtn}' = \theta \text{ButtonPanel}'] \\ \setminus (\text{ButtonPanel}')$$

$$\text{ItsDownButtonLightOn} \hat{=} (\text{DownButtonPanelOp} \wedge \text{ItsButtonLightOn}) \setminus (\Delta \text{ButtonPanel})$$

$$\text{ItsDownButtonLightOff} \hat{=} (\text{DownButtonPanelOp} \wedge \text{ItsButtonLightOff}) \setminus (\Delta \text{ButtonPanel})$$

5.4.3 下降用ボタン配列盤に関する判断

$$\text{DownButtonPanelState}$$

$$\text{DownButtonPanel}$$

$$\text{ButtonPanel}$$

$$f? : \text{FLOORNO}$$

$$f? \in \text{dom downbtn.btnpanel}$$

$$\theta \text{ButtonPanel} = \text{downbtn}$$

- ある下降用フロア ($f?$) ボタンが点いている
- ある下降用フロア ($f?$) ボタンが消えている

$$\text{ItsDownButtonIsBright} \hat{=} (\text{DownButtonPanelState} \wedge \text{ItsButtonIsBright}) \setminus (\text{ButtonPanel})$$

$$\text{ItsDownButtonIsDark} \hat{=} (\text{DownButtonPanelState} \wedge \text{ItsButtonIsDark}) \setminus (\text{ButtonPanel})$$

5.5 リフトボタンの配列盤

リフトには、その行先を指定するためのボタンが配備されている．これをリフトボタンという．そのボタンは、各フロアごとに一つずつ配置されている．従ってリフトボタンは配列盤を構成する．

5.5.1 リフトボタンの配置

$$\text{LiftButtonPanel}$$

$$\text{liftbtn} : \text{ButtonPanel}$$

$$\text{dom liftbtn.btnpanel} = \text{FLOORNO}$$

5.5.2 リフトボタンの配列盤に関する操作

配列盤の中のリフトボタン（フロア $f?$ に対応するボタン）に関する操作の枠．

$$\text{LiftButtonPanelOp}$$

$$\Delta \text{LiftButtonPanel}$$

$$\Delta \text{ButtonPanel}$$

$$f? : \text{FLOORNO}$$

$$f? \in \text{dom liftbtn.btnpanel}$$

$$\theta \text{ButtonPanel} = \text{liftbtn}$$

$$\text{liftbtn}' = \theta \text{ButtonPanel}'$$

- リフトボタン配列盤の初期化
- $f?$ に対応するリフトボタンを点燈する
- $f?$ に対応するリフトボタンを消燈する

$$\text{LiftButtonPanelInit} \hat{=} [\text{LiftButtonPanel}' ; \text{ButtonPanelInit} \mid \text{liftbtn}' = \theta \text{ButtonPanel}' \setminus (\text{ButtonPanel}')$$

$$\text{ItsLiftButtonLightOn} \hat{=} (\text{LiftButtonPanelOp} \wedge \text{ItsButtonLightOn}) \setminus (\Delta \text{ButtonPanel})$$

$$\text{ItsLiftButtonLightOff} \hat{=} (\text{LiftButtonPanelOp} \wedge \text{ItsButtonLightOff}) \setminus (\Delta \text{ButtonPanel})$$

5.5.3 リフトボタン配列盤に関する判断

$$\text{LiftButtonPanelState}$$

$$\text{LiftButtonPanel}$$

$$\text{ButtonPanel}$$

$$f? : \text{FLOORNO}$$

$$f? \in \text{dom liftbtn.btnpanel}$$

$$\theta \text{ButtonPanel} = \text{liftbtn}$$

- $f?$ に対応するリフトボタンが点いている
- $f?$ に対応するリフトボタンが消えている

$$\text{ItsLiftButtonIsBright} \hat{=} (\text{LiftButtonPanelState} \wedge \text{ItsButtonIsBright}) \setminus (\text{ButtonPanel})$$

$$\text{ItsLiftButtonIsDark} \hat{=} (\text{LiftButtonPanelState} \wedge \text{ItsButtonIsDark}) \setminus (\text{ButtonPanel})$$

6. フロアボタンとフロア要求の同期

6.1 上昇用ボタンと上昇要求との同期

6.1.1 上昇用ボタンと上昇要求の同期の枠

$$\text{UpButtonAndUpRequest}$$

$$\Delta \text{UpRequest}$$

$$\Delta \text{UpButtonPanel}$$

$$f? : \text{FLOORNO}$$

6.1.2 上昇用ボタンを押下する ItsUpButtonPush

フロア $f?$ の上昇用ボタンを押下すると、そのボタンが点燈し、上昇要求集合を $f?$ 追加する。

- フロア $f?$ の上昇用ボタンの消燈時 $ItsUpButtonPushWhenDark$
- フロア $f?$ の上昇用ボタンの点燈時 $ItsUpButtonPushWhenBright$
- 一般の場合 $UpButtonPush$

$$\begin{aligned}
 ItsUpButtonPushWhenDark &\hat{=} ItsUpButtonIsDark \wedge ItsUpButtonLightOn \wedge UpRequestAdd \\
 ItsUpButtonPushWhenBright &\hat{=} \\
 & [UpButtonAndUpRequest \wedge ItsUpButtonIsBright \mid \\
 & \quad \theta UpRequest' = \theta UpRequest; \theta UpButtonPanel' = \theta UpButtonPanel] \\
 ItsUpButtonPush &\hat{=} ItsUpButtonPushWhenDark \vee UpButtonPushWhenBright
 \end{aligned}$$

6.1.3 上昇要求を充たす $UpRequestSatisfy$

フロア $f?$ の上昇要求が充たされると、そのフロアボタンを消燈し、上昇要求集合からその要求を削除する。

$$UpRequestSatisfy \hat{=} UpRequestMayRemove \wedge ItsUpButtonLightOff$$

6.2 下降用ボタンと下降要求との同期

6.2.1 下降用ボタンと下降要求の同期の枠

$DownButtonAndDownRequest$ $\Delta DownRequest$ $\Delta DownButtonPanel$ $f? : FLOORNO$
--

6.2.2 下降用ボタンを押下する $ItsDownButtonPush$

フロア $f?$ の下降用ボタンを押下すると、そのボタンが点燈し、下降要求集合を $f?$ 追加する。

- フロア $f?$ の下降用ボタンの消燈時 $ItsDownButtonPushWhenDark$
- フロア $f?$ の下降用ボタンの点燈時 $ItsDownButtonPushWhenBright$
- 一般の場合 $DownButtonPush$

$$\begin{aligned}
 ItsDownButtonPushWhenDark &\hat{=} ItsDownButtonIsDark \wedge ItsDownButtonLightOn \wedge DownRequestAdd \\
 ItsDownButtonPushWhenBright &\hat{=} \\
 & [DownButtonAndDownRequest \wedge ItsDownButtonIsBright \mid \\
 & \quad \theta DownRequest' = \theta DownRequest; \theta DownButtonPanel' = \theta DownButtonPanel] \\
 ItsDownButtonPush &\hat{=} ItsDownButtonPushWhenDark \vee DownButtonPushWhenBright
 \end{aligned}$$

6.2.3 下降要求を充たす $DownRequestSatisfy$

フロア $f?$ の下降要求が充たされると、そのフロアボタンを消燈し、下降要求集合からその要求を削除する。

$$DownRequestSatisfy \hat{=} DownRequestMayRemove \wedge ItsDownButtonLightOff$$

7. リフトボタンとリフト行先要求の同期

7.1 リフトボタンを押下する $LiftButtonPush$

フロア $f?$ に対応するリフトボタンを押下すると、そのボタンが点燈し、リフト行

先要求集合に $f?$ を追加する .

7.1.1 LiftButtonOp

$LiftButtonOp$ $\Delta LiftRequest$ $\Delta LiftButtonPanel$ $f? : FLOORNO$
--

- $f?$ への行先ボタンの消燈時 $LiftButtonPushWhenDark$
- $f?$ への行先ボタンの点燈時 $LiftButtonPushWhenBright$
- 一般の場合 $LiftButtonPush$

$$LiftButtonPushWhenDark \hat{=} ItsLiftButtonIsDark \wedge ItsLiftButtonLightOn \wedge LiftRequestAdd$$

$$LiftButtonPushWhenBright \hat{=} [LiftButtonOp \wedge ItsLiftButtonIsBright \{ \theta LiftButtonPanel' = \theta LiftButtonPanel; \theta LiftRequest' = \theta LiftRequest \}]$$

$$LiftButtonPush \hat{=} LiftButtonPushWhenDark \vee LiftButtonPushWhenBright$$

7.2 行先要求を充たす $LiftRequestSatisfy$

フロア $f?$ への行先要求が充たされると, 行先要求から $f?$ を削除し, その行先ボタンを消燈する .

$$LiftRequestSatisfy \hat{=} LiftRequestMayRemove \wedge LiftButtonLightOff$$

8. リフト

8.1 リフトの状態

$$RUNMODE ::= waiting \mid movingUp \mid waitingUp \mid movingDown \mid waitingDown$$

$Lift$ $cf : FLOORNO$ $act : RUNMODE$

cf : リフトが停止しているときは, その停止しているフロア番号。リフトが動いているときには, その動く向きにある最も近くのフロア番号。

act : リフトの動き方には次のような区別がある。

- 待機中 ($waiting$): あるフロアに停止していて, 次に, 上昇するか下降するか決まっていない。
- 上昇中 ($movingUp$): 現在上昇中である。
- 上昇待機中 ($waitingUp$): あるフロアに停止しているが, 次に上のフロアへ行こうとしている。
- 下降中 ($movingDown$): 現在下降中である。
- 下降待機中 ($waitingDown$): あるフロアに停止しているが, 次に下のフロアへ行こうとしている。

8.2 リフトの行動

8.2.1 リフトを初期化する

リフトが存在する位置（フロア番号）を入力する。

LiftInit

$\Delta Lift'$

$f?$

$cf' = groundFloorNo$

$act' = waiting$

8.2.2 上昇指令を出す

LiftCommandLeavingUp

$\Delta Lift$

$cid! : COMKIND$

$act \in \{waiting, waitingUp, waitingDown\}$

$cf < topFloorNo$

$cid! = leaveUp$

$act' = movingUp$

$cf' = cf + 1$

8.2.3 上昇を継続する

LiftContinueLeavingUp

$\Delta Lift$

$cid! : COMKIND$

$act = movingUp$

$cf < topFloorNo$

$cid! = noOperation$

$act' = act$

$cf' = cf + 1$

8.2.4 上昇到着指令を出す

LiftCommandArrivingUp

$\Delta Lift$

$cid! : COMKIND$

$act = movingUp$

$cid! = arriveUp$

$act' = waitingUp$

$cf' = cf$

8.2.5 下降指令を出す

<i>LiftCommandLeavingDown</i> $\Delta Lift$ <i>cid!</i> : <i>COMKIND</i>
$act \in \{waiting, waitingDown, waitingUp\}$ $groundFloorNo < cf$ <i>cid!</i> = <i>leaveDown</i> <i>act'</i> = <i>movingDown</i> <i>cf'</i> = <i>cf</i> - 1

8.2.6 下降を継続する

<i>LiftContinueLeavingDown</i> $\Delta Lift$ <i>cid!</i> : <i>COMKIND</i>
$act = movingDown$ $groundFloorNo < cf$ <i>cid!</i> = <i>noOperation</i> <i>act'</i> = <i>act</i> <i>cf'</i> = <i>cf</i> - 1

8.2.7 下降到着指令を出す

<i>LiftCommandArrivingDown</i> $\Delta Lift$ <i>cid!</i> : <i>COMKIND</i>
$act = movingDown$ <i>cid!</i> = <i>arriveDown</i> $act = waitingDown$ <i>cf'</i> = <i>cf</i>

8.2.8 待機状態になる

<i>LiftWait</i> $\Delta Lift$ <i>cid!</i> : <i>COMKIND</i>
$act \in \{waitingUp, waitingDown\}$ <i>cid!</i> = <i>noOperation</i> $act = waiting$ <i>cf'</i> = <i>cf</i>

8.3 リフトの状況判断

8.3.1 リフトは待機中である

<i>LiftIsWaiting</i> <i>Lift</i>
$act = waiting$

8.3.2 リフトは上昇待機中である

<i>LiftIsWaitingUp</i>
<i>Lift</i>
<i>act = waitingUp</i>

8.3.3 リフトは下降待機中である

<i>LiftIsWaitingDown</i>
<i>Lift</i>
<i>act = waitingDown</i>

8.3.4 リフトは上昇中である

<i>LiftIsMovinUp</i>
<i>Lift</i>
<i>act = movingUp</i>

8.3.5 リフトは下降中である

<i>LiftIsMovinUp</i>
<i>Lift</i>
<i>act = movingDown</i>

9. リフトと他の対象との協調

9.1 リフトの協調のための環境

リフトはリフトボタンの配列盤やリフトの行先要求と一体となっている。これをリフトパッケージ (*LiftPackage*) と呼ぶ。パッケージ内の対象は相互に協調し合う。

<i>LiftPkg</i>
<i>LiftButtonPanel</i>
<i>LiftRequest</i>
<i>Lift</i>

リフトはフロア要求 (上昇要求, 下降要求) やフロアボタン (上昇ボタン, 下降ボタン) とも協調する。リフトの協調のための環境は次のようになる。

<i>LiftCoopEnv</i>
$\Delta UpButtonPanel$; $\Delta UpRequest$
$\Delta DownButtonPanel$; $\Delta DownRequest$
$\Delta LiftPkg$
<i>cid!</i> : <i>COMKIND</i>

9.2 リフトとフロア要求との協調

- 上昇するとき : $LiftLeavesUp$
上昇指令を出すときは、フロアの上昇要求を削除し、フロアの上昇ボタンを消燈する。
- 下降するとき : $LiftLeavesDown$
下降指令を出すときは、フロアの下降要求を削除し、フロアの下降ボタンを消燈する。

$$LiftLeaveUp \hat{=} [LiftCommandLeavingUp; UpRequestRemove; ItsUpButtonLightOff \mid f? = cf] \setminus (f?) \\ \wedge [LiftCoopEnv \mid \theta DownButtonPanel' = \theta DownButtonPanel; \theta DownRequest' = \theta DownRequest; \\ \theta LiftButtonPanel' = \theta LiftButtonPanel; \theta LiftRequest' = \theta LiftRequest]$$

$$LiftLeaveDown \hat{=} \\ [LiftCommandLeavingDown; DownRequestRemove; ItsDownButtonLightOff \mid f? = cf] \setminus (f?) \\ \wedge [LiftCoopEnv \mid \theta UpButtonPanel' = \theta UpButtonPanel; \theta UpRequest' = \theta UpRequest; \\ \theta LiftButtonPanel' = \theta LiftButtonPanel; \theta LiftRequest' = \theta LiftRequest]$$

9.3 リフトとリフトの行先要求との協調

- 上昇到着する : $LiftArrivesUp$
到着指令を出すときは、そのフロアへの到着要求を削除し、リフトボタンを消燈する。
- 下降到着する : $LiftArrivesDown$
到着指令を出すときは、そのフロアへの到着要求を削除し、リフトボタンを消燈する。

$$LiftArriveUp \hat{=} [LiftCommandArrivingUp; LiftRequestRemove; ItsLiftButtonLightOff \mid f? = cf] \setminus (f?) \\ \wedge [LiftCoopEnv \mid \theta UpButtonPanel' = \theta UpButtonPanel; \theta UpRequest' = \theta UpRequest; \\ \theta DownButtonPanel' = \theta DownButtonPanel; \theta DownRequest' = \theta DownRequest]$$

$$LiftArriveDown \hat{=} \\ [LiftCommandArrivingDown; LiftRequestRemove; ItsLiftButtonLightOff \mid f? = cf] \setminus (f?) \\ \wedge [LiftCoopEnv \mid \theta UpButtonPanel' = \theta UpButtonPanel; \theta UpRequest' = \theta UpRequest; \\ \theta DownButtonPanel' = \theta DownButtonPanel; \theta DownRequest' = \theta DownRequest]$$

9.4 協調しないとき

- 上昇を継続するとき : $LiftContinueUp$
上昇を継続するときは、周囲には影響をおよぼさない。
- 下降を継続するとき : $LiftContinueDown$
下降を継続するときは、周囲には影響をおよぼさない。
- 待機状態に戻るとき : $LiftContinueDown$
待機状態に戻るときは、周囲には影響をおよぼさない。

$$LiftContinueUpInEnv \hat{=} LiftContinueLeavingUp \\ \wedge [LiftCoopEnv \mid \theta UpButtonPanel' = \theta UpButtonPanel; \theta UpRequest' = \theta UpRequest; \\ \theta DownButtonPanel' = \theta DownButtonPanel; \theta DownRequest' = \theta DownRequest \\ \theta LiftButtonPanel' = \theta LiftButtonPanel; \theta LiftRequest' = \theta LiftRequest]$$

$$LiftContinueDownInEnv \hat{=} LiftContinueLeavingDown \\ \wedge [LiftCoopEnv \mid \theta UpButtonPanel' = \theta UpButtonPanel; \theta UpRequest' = \theta UpRequest; \\ \theta DownButtonPanel' = \theta DownButtonPanel; \theta DownRequest' = \theta DownRequest \\ \theta LiftButtonPanel' = \theta LiftButtonPanel; \theta LiftRequest' = \theta LiftRequest]$$

$$LiftWaitInEnv \hat{=} LiftWait \\ \wedge [LiftCoopEnv \mid \theta UpButtonPanel' = \theta UpButtonPanel; \theta UpRequest' = \theta UpRequest; \\ \theta DownButtonPanel' = \theta DownButtonPanel; \theta DownRequest' = \theta DownRequest \\ \theta LiftButtonPanel' = \theta LiftButtonPanel; \theta LiftRequest' = \theta LiftRequest]$$

10. リフトに対するフロア要求と行先要求に関する判断

$$\begin{aligned}
\text{FloorRequestFromAboveFlr} &\hat{=} \text{UpRequestFromAboveFlr} \vee \text{DownRequestFromAboveFlr} \\
\text{FloorRequestFromBelowFlr} &\hat{=} \text{UpRequestFromBelowFlr} \vee \text{DownRequestFromBelowFlr} \\
\text{ToGoUp} &\hat{=} \text{LiftRequestToAboveFlr} \vee \text{FloorRequestFromAboveFlr} \\
\text{ToArriveUp} &\hat{=} \text{LiftRequestToCurrentFlr} \vee \text{UpRequestFromCurrentFlr} \vee \neg \text{ToGoUp} \\
\text{ToGoDown} &\hat{=} \text{LiftRequestToBelowFlr} \vee \text{FloorRequestFromBelowFlr} \\
\text{ToArriveDown} &\hat{=} \text{LiftRequestToCurrentFlr} \vee \text{DownRequestFromCurrentFlr} \vee \neg \text{ToGoDown}
\end{aligned}$$

11. リフトの反応

シグナル *canLeaveSig* や *arrivingSig* を受信したときのリフトの反応 .

$ \begin{aligned} &\text{LiftReactEnv} \\ &\Delta \text{UpButtonPanel}; \Delta \text{UpRequest} \\ &\Delta \text{DownButtonPanel}; \Delta \text{DownRequest} \\ &\Delta \text{LiftPkg} \\ &\text{cid!} : \text{COMMANDID} \end{aligned} $

$$\begin{aligned}
\text{LiftReactForCanLeaveSigWhenWaiting} &\hat{=} \\
&\text{LiftIsWaiting} \\
&\wedge ([\text{ToGoUp} \wedge \text{ToGoDown} \mid f? = cf] \Rightarrow \text{LiftLeaveUp} \vee \text{LiftLeaveDown}) \\
&\wedge ([\text{ToGoUp} \wedge \neg \text{ToGoDown} \mid f? = cf] \Rightarrow \text{LiftLeaveUp}) \\
&\wedge ([\neg \text{ToGoUp} \wedge \text{ToGoDown} \mid f? = cf] \Rightarrow \text{LiftLeaveDown}) \\
&\wedge ([\neg \text{ToGoUp} \wedge \neg \text{ToGoDown} \mid f? = cf] \Rightarrow \text{LiftWaitInEnv})
\end{aligned}$$

$$\begin{aligned}
\text{LiftReactForArrivingWhenMovingUp} &\hat{=} \\
&\text{LiftIsMovingUp} \\
&\wedge ([\text{ToArriveUp} \mid f? = cf] \Rightarrow \text{LiftArriveUp}) \\
&\wedge ([\neg \text{ToArriveUp} \mid f? = cf] \Rightarrow \text{LiftContinueUpInEnv})
\end{aligned}$$

$$\begin{aligned}
\text{LiftReactForCanLeaveSigWhenWaitingUp} &\hat{=} \\
&\text{LiftIsWaitingUp} \\
&\wedge ([\text{ToGoUp} \mid f? = cf] \Rightarrow \text{LiftLeaveUp}) \\
&\wedge ([\neg \text{ToGoUp} \wedge \text{ToGoDown} \mid f? = cf] \Rightarrow \text{LiftLeaveDown}) \\
&\wedge ([\neg \text{ToGoUp} \wedge \neg \text{ToGoDown} \mid f? = cf] \Rightarrow \text{LiftWaitInEnv})
\end{aligned}$$

$$\begin{aligned}
\text{LiftReactForArrivingSigWhenMovingDown} &\hat{=} \\
&\text{LiftIsMovingDown} \\
&\wedge ([\text{ToArriveDown} \mid f? = cf] \Rightarrow \text{LiftArriveDown}) \\
&\wedge ([\neg \text{ToArriveDown} \mid f? = cf] \Rightarrow \text{LiftContinueDownInEnv})
\end{aligned}$$

$$\begin{aligned}
\text{LiftReactForCanLeaveSigWhenWaitingDown} &\hat{=} \\
&\text{LiftIsWaitingDown} \\
&\wedge ([\text{ToGoDown} \mid f? = cf] \Rightarrow \text{LiftLeaveDown}) \\
&\wedge ([\neg \text{ToGoDown} \wedge \text{ToGoUp} \mid f? = cf] \Rightarrow \text{LiftLeaveUp}) \\
&\wedge ([\neg \text{ToGoDown} \wedge \neg \text{ToGoUp} \mid f? = cf] \Rightarrow \text{LiftWaitInEnv})
\end{aligned}$$

$$\begin{aligned}
\text{LiftReactForCanLeaveSig} &\hat{=} \text{LiftReactForCanLeaveSigWhenWaiting} \\
&\vee \text{LiftReactForCanLeaveSigWhenWaitingUp} \\
&\vee \text{LiftReactForCanLeaveSigWhenWaitingDown}
\end{aligned}$$

$$\begin{aligned} \text{LiftReactForArrivingSig} &\hat{=} \text{LiftReactForArrivingWhenMovingUp} \\ &\vee \text{LiftReactForArrivingWhenMovingDown} \end{aligned}$$

$$\text{LiftReactForSignal} \hat{=} \text{LiftReactForCanLeaveSig} \vee \text{LiftReactForArrivingSig}$$

12. 指定されたリフトパッケージの挙動

リフトは複数台ある．リフトボタンパネルやリフト行先集合もリフトと同じ数だけある．

12.1 リフトパッケージの配列

$$\begin{array}{l} \text{LiftPkgArray} \\ \text{liftpkgarray} : \text{LIFTID} \rightarrow \text{LiftPkg} \end{array}$$

12.2 リフトパッケージ配列の初期化

$$\begin{array}{l} \text{LiftPkgArrayInit} \\ \text{LiftPkgArray}' \\ \text{farray?} : \text{LIFTID} \rightarrow \text{FLOORNO} \\ \forall l : \text{LIFTID} \bullet \\ \quad [\text{LiftButtonPanelInit}; \text{LiftRequestInit}; \text{LiftInit} \mid \\ \quad f? = \text{farray}(l); \text{liftpkgarray}'(l) = \theta \text{LiftPkg}'] \setminus (\text{LiftPkg}', f?) \end{array}$$

12.3 指定されたリフトの挙動

リフトパッケージの配列の中の，ある一つのリフト $l?$ が，シグナル canLeaveSig や arrivingSig を受信したときのリフトの挙動．フロアボタン，フロア要求の集合，当リフト $l?$ の行先ボタンパネル，行先要求と協調する．その環境は次のとおり．

$$\begin{array}{l} \text{ItsLiftReactEnv} \\ \Delta \text{UpButtonPanel}; \Delta \text{UpRequest} \\ \Delta \text{DownButtonPanel}; \Delta \text{DownRequest} \\ \Delta \text{LiftPkgArray} \\ \Delta \text{LiftPkg} \\ \text{cid!} : \text{COMKIND} \\ \text{l?} : \text{LIFTID} \\ \text{c!} : \text{COMMAND} \\ \theta \text{LiftPkg} = \text{liftpkgarray}(l?) \\ \text{liftpkgarray}' = \text{liftpkgarray} \oplus \{l? \mapsto \theta \text{LiftPkg}'\} \\ \text{cid!} = \text{noOperation} \Rightarrow \text{c!} = \text{noCommand} \\ \text{cid!} \neq \text{noOperation} \Rightarrow \text{c!} = \text{liftCommand}(\text{cid!}, l?) \end{array}$$

$$\text{ItsLiftReactForSignal} \hat{=} (\text{LiftReactForSignal} \wedge \text{ItsLiftReactEnv}) \setminus (\Delta \text{LiftPkg}, \text{cid!})$$

12.4 待機中のリフトがあればそれを起動する

$LiftPkgState$
$LiftPkgArray$
$LiftPkg$
$l? : LIFTID$
$\theta LiftPkg = liftpkgarray(l?)$

$$ActivateWaitingLiftIfAny \hat{=} (\exists l? : LIFTID \bullet (LiftPkgState \wedge LiftIsWaiting)) \Rightarrow (ItsLiftReactEnv \wedge LiftReactForCanLeaveSigWhenWaiting \setminus (l?, \Delta LiftPkg, cid!))$$

12.5 指定されたリフトボタンの挙動

リフトパッケージの配列の中の、ある一つのリフト $l?$ の中の行先ボタン $f?$ が押下された．その環境は次のとおり．

$ItsLiftButtonOp$
$\Delta LiftPkgArray$
$\Delta LiftPkg$
$l? : LIFTID$
$f? : FLOORNO$
$\theta LiftPkg = liftpkgarray(l?)$
$\theta Lift' = \theta Lift$
$liftpkgarray' = liftpkgarray \oplus \{l? \mapsto \theta LiftPkg'\}$

$$ItsLiftButtonPush \hat{=} (LiftButtonPush \wedge ItsLiftButtonOp) \setminus (\Delta LiftPkg)$$

13. リフトシステムの作動環境下での挙動

13.1 リフトシステムの作動環境

$LiftSysEnv$
$UpButtonPanel; UpRequest$
$DownButtonPanel; DownRequest$
$LiftPkgArray$

$LiftSysInOutEnv$
$\Delta LiftSysEnv$
$s? : SIGNAL$
$c! : COMMAND$

13.2 システム環境下での上昇ボタンの押下 *PushUpButtonInSysEnv*

上昇ボタンが押下されたら

- そのボタンを点燈し、
- その上昇要求を登録し、

そのあと

- 待機状態のリフトが、もしあれば、それを起動する。

詳しくは以下のとおり。

$\begin{array}{l} \text{UpButtonOpInSysEnv} \\ \text{LiftSysInOutEnv} \\ f? : \text{FLOORNO} \\ \hline s? = \text{pushUpButton}(f?) \\ \theta \text{DownButtonPanel}' = \theta \text{DownButtonPanel}; \theta \text{DownRequest}' = \theta \text{DownRequest} \\ \theta \text{LiftPkgArray}' = \theta \text{LiftPkgArray} \\ c! = \text{noCommand} \end{array}$

$$\begin{array}{l} \text{PushUpButtonInSysEnv} \hat{=} (\text{ItsUpButtonPush} \wedge \text{UpButtonOpInSysEnv}) \setminus (f?) \\ \quad \S \text{ActivateWaitingLiftIfAny} \end{array}$$

ここで \S については後出。

13.3 システム環境下での下降ボタンの押下 *PushDownButtonInSysEnv*

$\begin{array}{l} \text{DownButtonOpInSysEnv} \\ \text{LiftSysInOutEnv} \\ f? : \text{FLOORNO} \\ \hline s? = \text{pushDownButton}(f?) \\ \theta \text{UpButtonPanel}' = \theta \text{UpButtonPanel}; \theta \text{UpRequest}' = \theta \text{UpRequest} \\ \theta \text{LiftPkgArray}' = \theta \text{LiftPkgArray} \\ c! = \text{noCommand} \end{array}$

下降ボタンが押下されたら

- そのボタンを点燈し、
- その下降要求を登録し、

そのあと

- 待機状態のリフトが、もしあれば、それを起動する。

詳しくは以下のとおり。

$$\begin{array}{l} \text{PushDownButtonInSysEnv} \hat{=} (\text{ItsDownButtonPush} \wedge \text{DownButtonOpInSysEnv}) \setminus (f?) \\ \quad \S \text{ActivateWaitingLiftIfAny} \end{array}$$

ここで \S については後出。

13.4 システム環境下でのリフトボタンの押下 *PushLiftButtonInSysEnv*

$\begin{array}{l} \text{LiftButtonOpInSysEnv} \\ \text{LiftSysInOutEnv} \\ l? : \text{LIFTID} \\ f? : \text{FLOORNO} \\ \hline s? = \text{pushLiftButton}(l?, f?) \\ \theta \text{UpButtonPanel}' = \theta \text{UpButtonPanel} \wedge \theta \text{UpRequest}' = \theta \text{UpRequest} \\ \theta \text{DownButtonPanel}' = \theta \text{DownButtonPanel} \wedge \theta \text{DownRequest}' = \theta \text{DownRequest} \\ c! = \text{noCommand} \end{array}$

$$\text{PushLiftButtonInSysEnv} \hat{=} (\text{ItsLiftButtonPush} \wedge \text{LiftButtonOpInSysEnv}) \setminus (l?, f?)$$
13.5 システム環境下でのリフトの挙動 *PushLiftButtonInSysEnv*

$\begin{array}{l} \text{LiftActInSysEnv} \\ \text{LiftSysInOutEnv} \\ l? : \text{LIFTID} \\ \hline s? = \text{canLeaveSignal}(l?) \vee s? = \text{arrivingSignal}(l?) \end{array}$
--

$$\text{LiftRunInSysEnv} \hat{=} (\text{ItsLiftReactForSignal} \wedge \text{LiftActInSysEnv}) \setminus (l?)$$

14. リフトシステムの仕様

$$\begin{aligned} \text{LiftSystem} &= \text{LiftSysInit} ; \text{LiftSysRun} \\ \text{LiftSysInit} &= \text{UpButtonPanelInit} \wedge \text{UpRequestInit} \\ &\quad \wedge \text{DownButtonPanelInit} \wedge \text{DownRequestInit} \\ &\quad \wedge \text{LiftPkgArrayInit} \\ \text{LiftSysRun} &= \\ &\quad ((\text{PushUpButtonInSysEnv} \\ &\quad \quad | \text{PushDownButtonInSysEnv} \\ &\quad \quad | \text{PushLiftButtonInSysEnv} \\ &\quad \quad | \text{LiftRunInSysEnv}) ; \text{LiftSysRun}) \\ &\quad | \text{LiftSysFin} \end{aligned}$$

注意 1 : ここで基本スキーマ S やその結合 $S ; T$ は、次のようなトレースの集合を表すものとする。

- 基本スキーマ S は、空列と長さ 1 の束縛列の集合を表す。

$$\text{tr } S \hat{=} \{\langle \rangle\} \cup \{S \bullet \langle \theta S \rangle\}$$

- 勝手なスキーマ S, T について

$$\text{tr}(S | T) \hat{=} \text{tr } S \cup \text{tr } T$$

とする。

- $S = [\Delta \text{State}; \text{In}S?; \text{Out}S!], \dots, T = [\Delta \text{State}; \text{In}T?; \text{Out}T!], U = [\Delta \text{State}; \text{In}U?; \text{Out}U!]$ であるようなスキーマ S, \dots, T, U について

$$\begin{aligned} \text{tr}(S \circ \dots \circ T \circ U) &\hat{=} \{\langle \rangle\} \\ &\cup \{ T_1; U_2 \mid t \in \text{tr}(S \circ \dots \circ T) \wedge \text{last } t = \theta T_1 \wedge \theta \text{State}'_1 = \theta \text{State}'_2 \bullet t \wedge \langle \theta U_2 \rangle\} \\ \text{tr}((S \mid T) \circ U) &\hat{=} \text{tr}(S \circ U) \cup \text{tr}(T \circ U) \end{aligned}$$

とする。

4. スキーマ S について、つぎの再帰式

$$T = S \circ T$$

は、特に断らない限り、つぎの最小解 T を定義するものとする。

$$T \hat{=} \{\langle \rangle\} \cup \text{tr } S \cup \text{tr}(S \circ S) \cup \dots \cup \text{tr}(S \circ S \circ \dots \circ S) \cup \dots$$

注意 2 : Z 言語の標準的な解釈では S は

$$S \hat{=} \{ S \bullet \theta S \}$$

を表し、 $S \circ T$ はつぎのスキーマ

$$\begin{aligned} S \circ T &\hat{=} \exists \text{State}'' \bullet [S; \text{State}'' \mid \theta \text{State}' = \theta \text{State}''] \\ &\quad \wedge [T; \text{State}'' \mid \theta \text{State} = \theta \text{State}''] \end{aligned}$$

を表す。これは通常の \circ の解釈とは異なる。上記解釈はこれとは違い、通常の \circ の解釈に従う。

執筆者紹介 染谷 誠 (Makoto Someya)

1942 年生 . 1966 年立教大学理学部数学科卒業 . 1969 年同大学院修士課程中退 . (株)日本ユニパック総合研究所入社 . その後日本ユニパック(株)現日本ユニシス(株)に移籍 . 現在 , 情報技術部技術研究開発室 .