Cabin - Ping DFD

Leon Starr
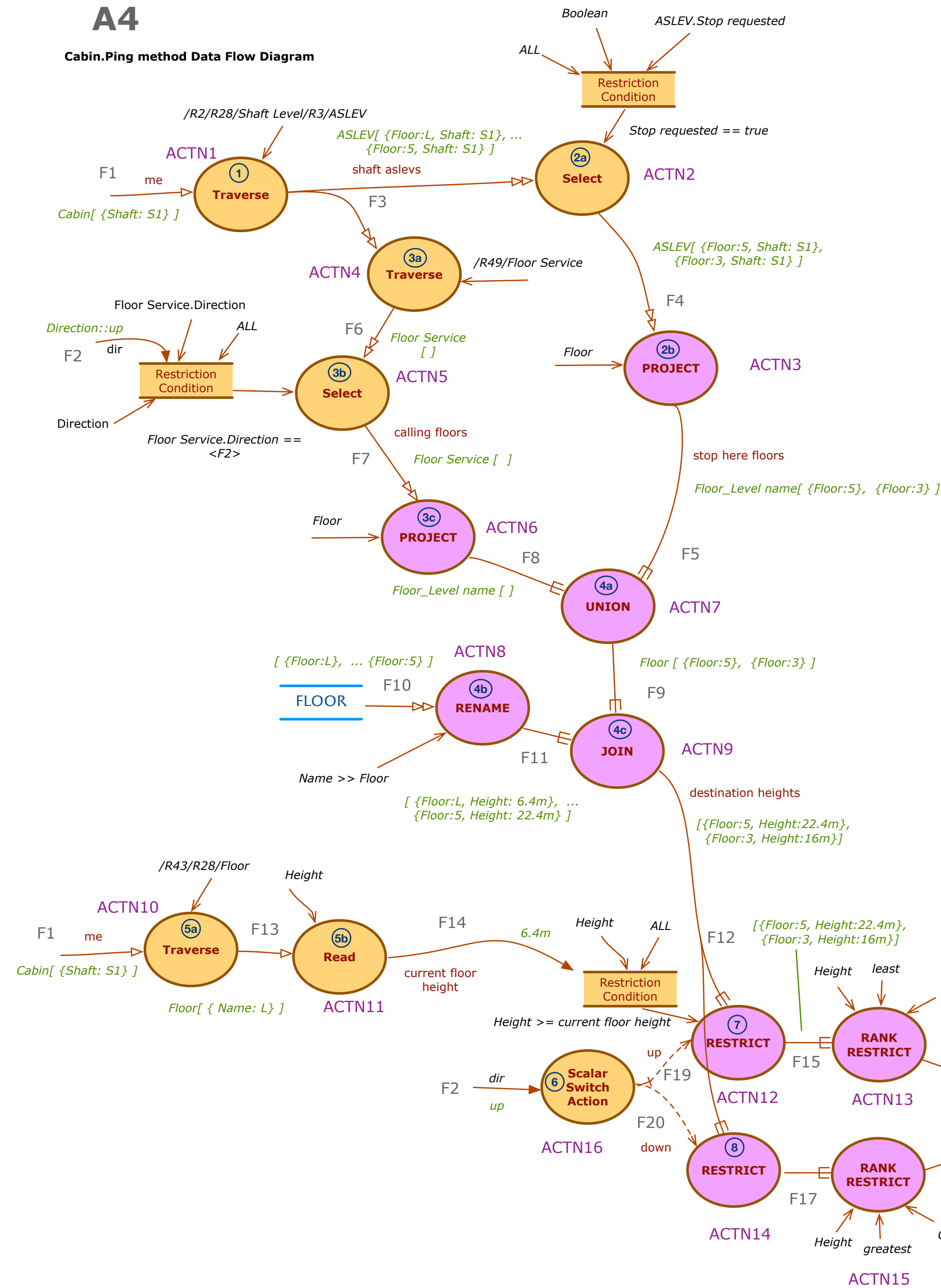2025-10-29 / v0.6

```
--
Cabin.Ping( dir: Direction ) : Accessible Shaft Level
--

//
// Search ahead of the Cabin in the indicated direction
// return the nearest ASL with a stop request and/or a Floor Service calling
// in that direction. If none is found, return the empty set.

// Everywhere I could stop
shaft aslevs ..= /R2/R28/Shaft Level/R3/Accessible Shaft Level
// Pending stop and call requests
stop here floors #= shaft aslevs( Stop requested ).Floor
calling floors #= shaft aslevs/R49/Floor Service(Direction : ^dir ).Floor

// Take the union of all requests and join with Floor to get a table of heights for each
destination heights #= ((stop here floors + calling floors) ## Floor[Name >> Floor])

// Find all Accessible Shaft Levels ahead of the cabin
current floor height = /R43/R28/Floor.Height
^dir? {
    _up:
        // Get the lowest floor at or above the cabin
        nearest dest ||= destination heights(Height >= current floor height)(1, ^-Height)
    _down:
        // Get the highest floor below the cabin
        nearest dest ||= destination heights(Height < current floor height)(1, ^+Height)
}
// Since you can't have two floors at the same height for the same shaft, we have
// at most one row in our table, nonetheless we set cardinality to 1
// to clarify that we have a single tuple result

// Convert to either zero or one instance of ASLEV class and provide as output
=>> Accessible Shaft Level( Floor: nearest dest.Floor; Shaft )
```
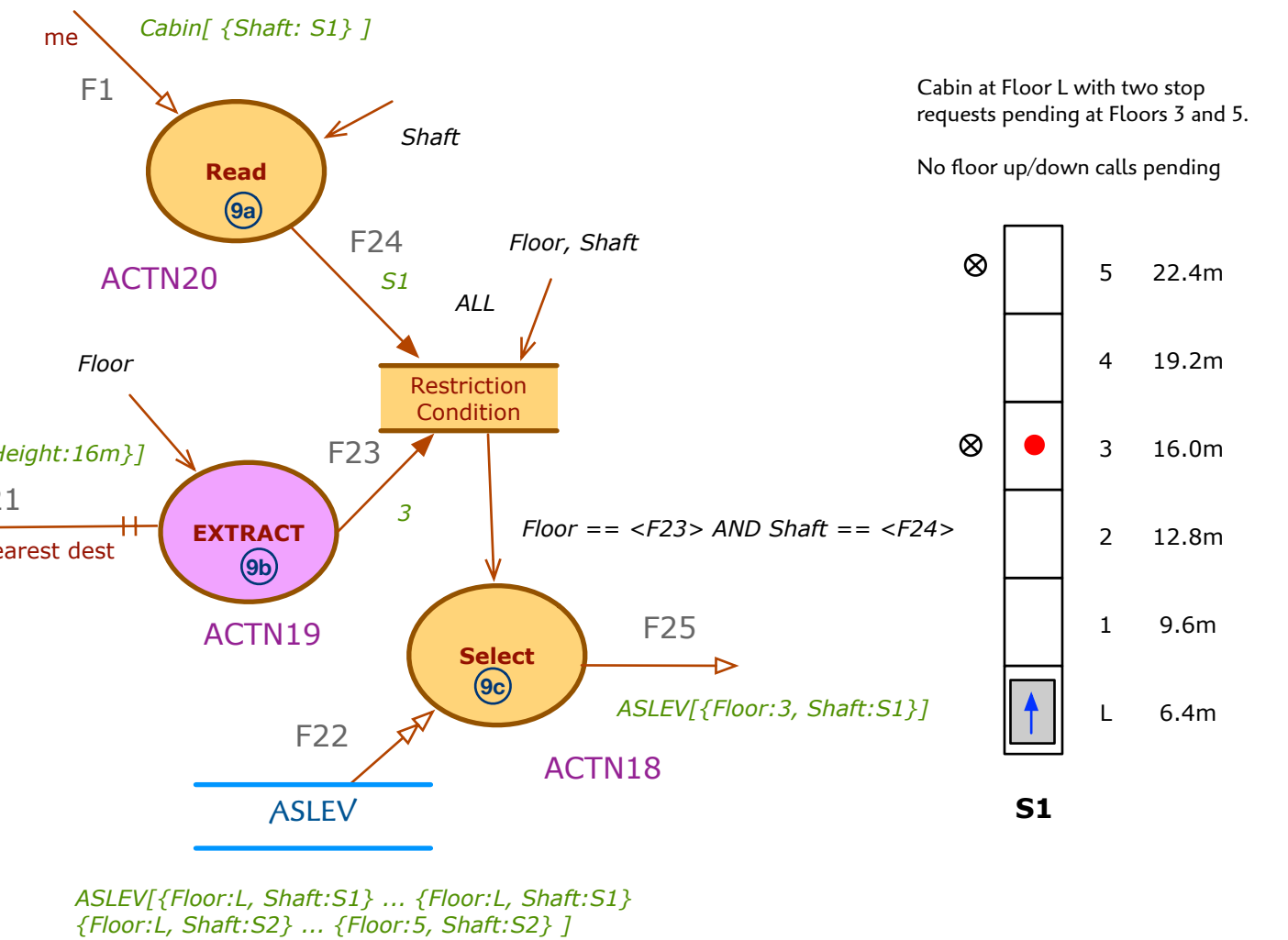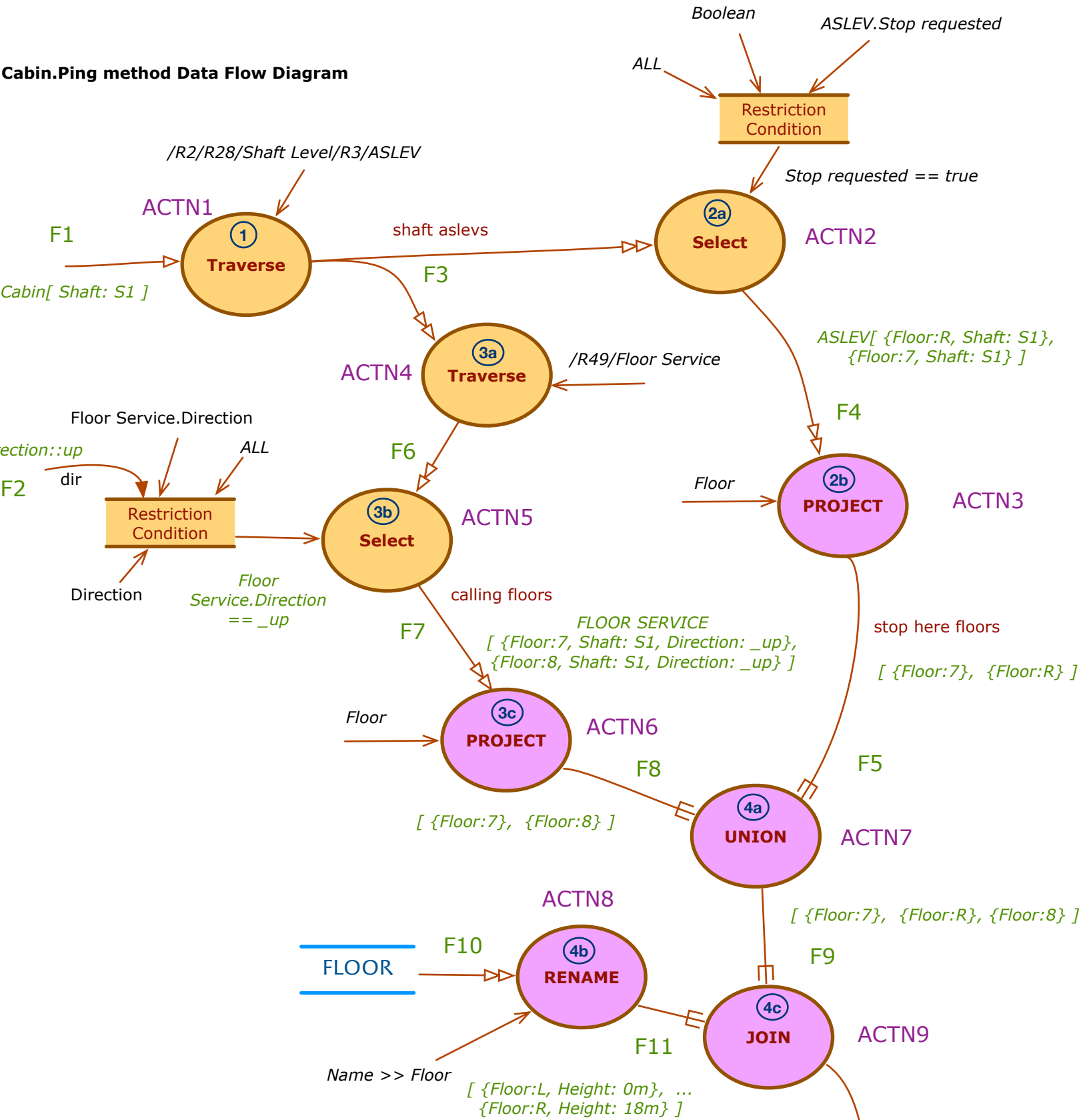
① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨

ACTN1 ① **Traverse** — F1 me — /R2/R28/Shaft Level/R3/ASLEV
Cabin[{Shaft: S1}]
ASLEV[{Floor:L, Shaft: S1}, ... {Floor:5, Shaft: S1}]
shaft aslevs — F3

Restriction Condition — Boolean — ALL — ASLEV.Stop requested
Stop requested == true

ACTN2 ②a **Select**
ASLEV[{Floor:5, Shaft: S1}, {Floor:3, Shaft: S1}] — F4

ACTN4 ③a **Traverse** — /R49/Floor Service — F6

ACTN3 ②b **PROJECT** — Floor
stop here floors
Floor_Level name[{Floor:5}, {Floor:3}]

Restriction Condition — Floor Service.Direction — ALL — Direction
Direction::up — F2 dir
Floor Service.Direction == <F2>

ACTN5 ③b **Select**
Floor Service [ ]
calling floors — F7
Floor Service [ ]

ACTN6 ③c **PROJECT** — Floor
Floor_Level name [ ] — F8

ACTN7 ④a **UNION** — F5
Floor [{Floor:5}, {Floor:3}] — F9

ACTN8 ④b **RENAME** — FLOOR — F10
[{Floor:L}, ... {Floor:5}]
Name >> Floor — F11

ACTN9 ④c **JOIN**
[{Floor:L, Height: 6.4m}, ... {Floor:5, Height: 22.4m}]
destination heights
[{Floor:5, Height:22.4m}, {Floor:3, Height:16m}]

ACTN10 ⑤a **Traverse** — /R43/R28/Floor — F1 me — F13
Cabin[{Shaft: S1}]

ACTN11 ⑤b **Read** — Height — F14
Floor[{ Name: L}]
current floor height
6.4m

Restriction Condition — Height — ALL
Height >= current floor height

ACTN16 ⑥ **Scalar Switch Action** — F2 dir
up — up F19 — down F20

ACTN12 ⑦ **RESTRICT** — F12 — F15
[{Floor:5, Height:22.4m}, {Floor:3, Height:16m}]

ACTN13 ⑦ **RANK RESTRICT** — Height — least — ONE
up nearest dest
[{Floor:3, Height:16m}] — F16

ACTN14 ⑧ **RESTRICT** — F17

ACTN15 ⑧ **RANK RESTRICT** — Height — greatest — ONE
down nearest dest — F18

ACTN17 — F21 nearest dest
[{Floor:3, Height:16m}]

ACTN19 ⑨b **EXTRACT** — Floor — F23 — 3
3

ACTN20 ⑨a **Read** — me — F1 — Shaft — F24
Cabin[{Shaft: S1}]
S1

Restriction Condition — Floor, Shaft — ALL
Floor == <F23> AND Shaft == <F24>

ACTN18 ⑨c **Select** — F25
ASLEV[{Floor:3, Shaft:S1}]
ASLEV — F22
*ASLEV[{Floor:L, Shaft: S1} ... {Floor:L, Shaft:S1}
{Floor:L, Shaft: S2} ... {Floor:5, Shaft:S2}]*

Cabin at Floor L with two stop requests pending at Floors 3 and 5.

No floor up/down calls pending

⊗		5	22.4m
		4	19.2m
⊗	●	3	16.0m
		2	12.8m
		1	9.6m
	↑	L	6.4m

S1

Cabin.Ping method Data Flow Diagram

The code panel (right side):

```
--
Cabin.Ping( dir: Direction ) : Accessible Shaft Level
--

//
// Search ahead of the Cabin in the indicated direction
// return the nearest ASL with a stop request and/or a Floor Service calling
// in that direction. If none is found, return the empty set.

// Everywhere I could stop
shaft aslevs ..= /R2/R28/Shaft Level/R3/Accessible Shaft Level
// Pending stop and call requests
stop here floors #= shaft aslevs( Stop requested ).Floor
calling floors #= shaft aslevs/R49/Floor Service(Direction : ^dir ).Floor

// Take the union of all requests and join with Floor to get a table of heights for each
destination heights #= ((stop here floors + calling floors) ## Floor[Name >> Floor])

// Find all Accessible Shaft Levels ahead of the cabin
current floor height = /R43/R28/Floor.Height
^dir? {
    _up:
        // Get the lowest floor at or above the cabin
        nearest dest ||= destination heights(Height >= current floor height)(1, ^-Height)
    _down:
        // Get the highest floor below the cabin
        nearest dest ||= destination heights(Height < current floor height)(1, ^+Height)
}
// Since you can't have two floors at the same height for the same shaft, we have
// at most one row in our table, nonetheless we set cardinality to 1
// to clarify that we have a single tuple result

// Convert to either zero or one instance of ASLEV class and provide as output
=>> Accessible Shaft Level( Floor: nearest dest.Floor; Shaft )
```
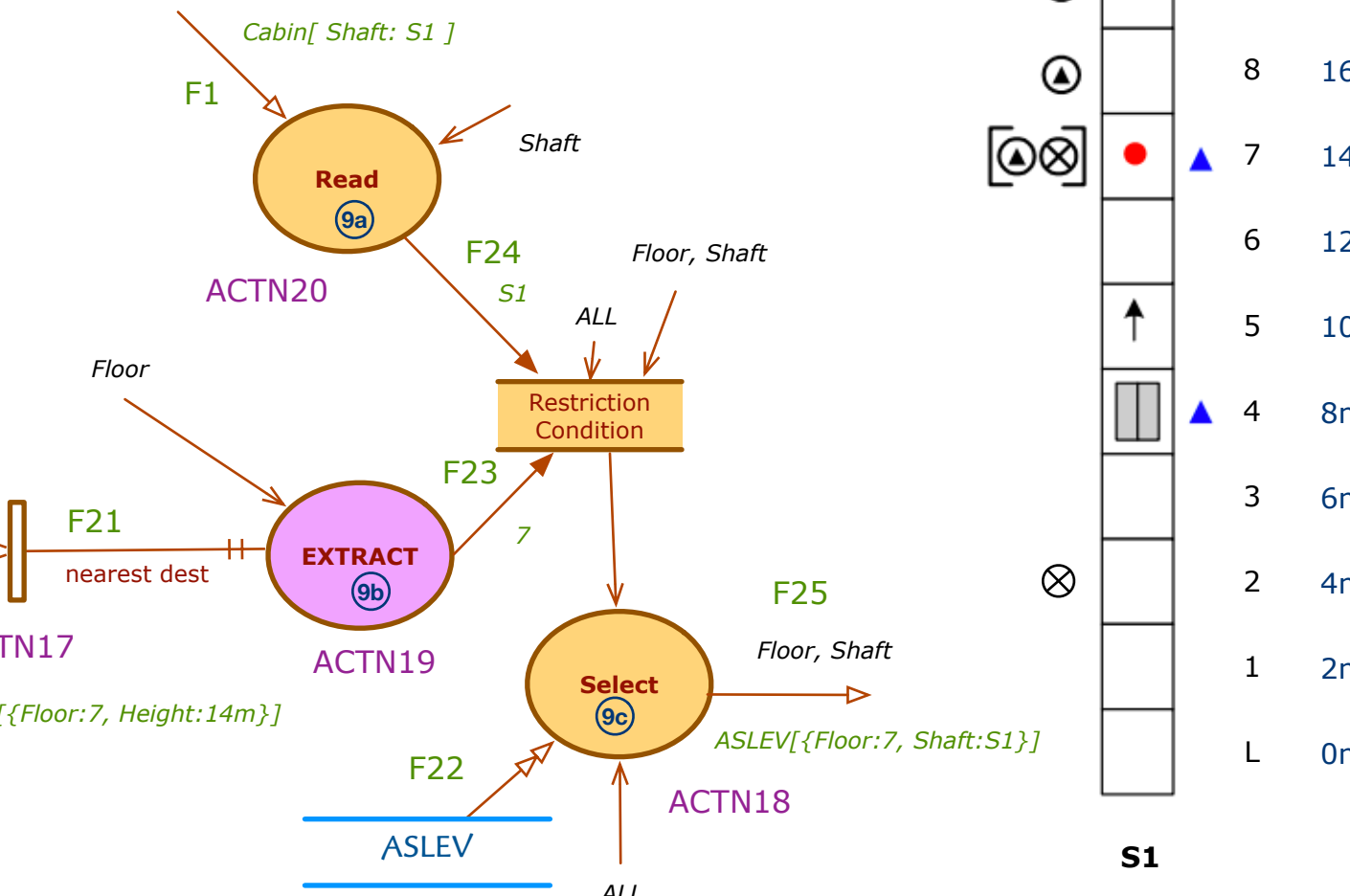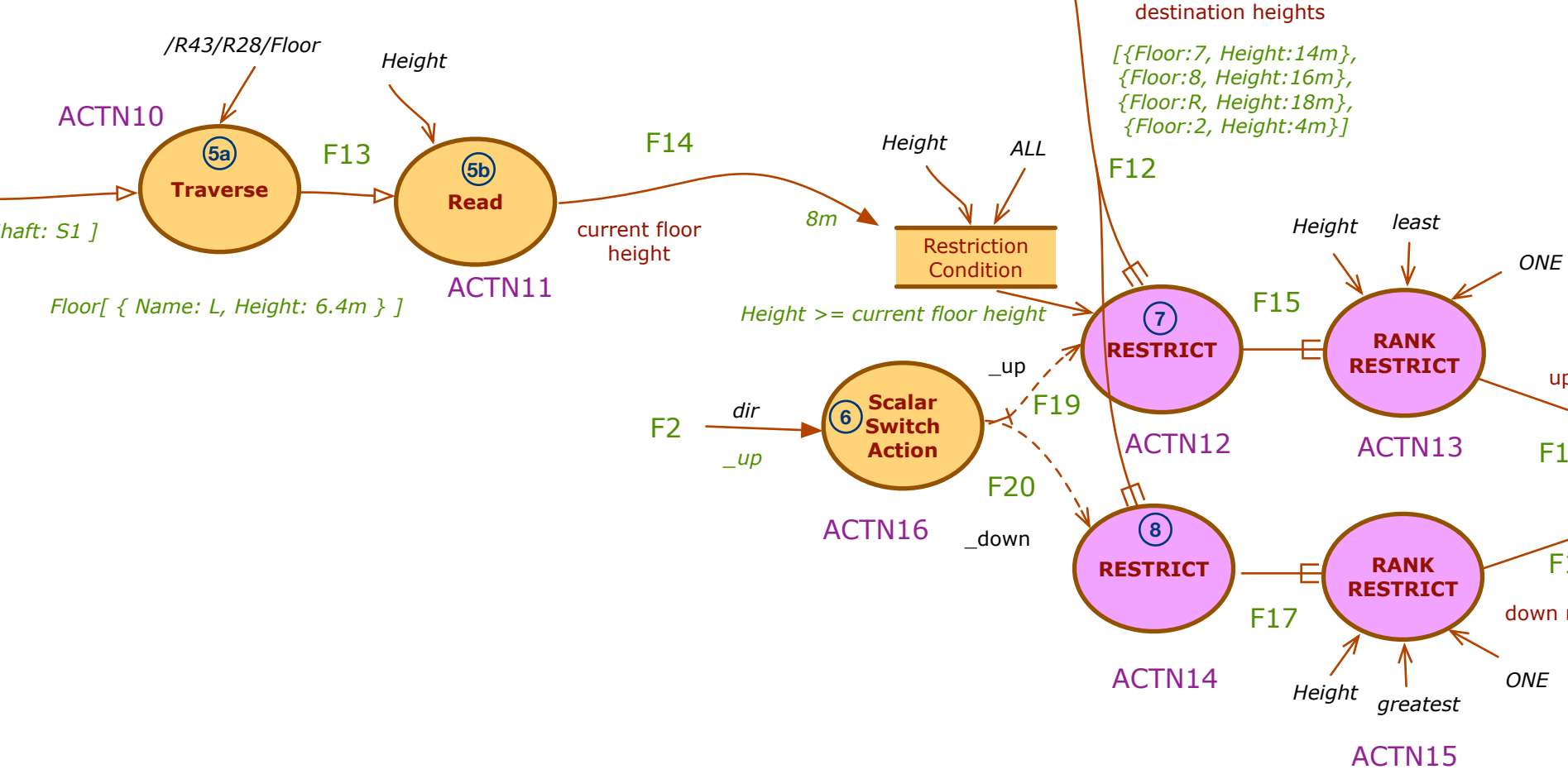
Code line markers: ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨

Diagram labels:

ACTN1 ① Traverse — /R2/R28/Shaft Level/R3/ASLEV — F1 — Cabin[Shaft: S1] — shaft aslevs F3

ACTN2 ②a Select — ASLEV[{Floor:R, Shaft: S1}, {Floor:7, Shaft: S1}]

Restriction Condition — Boolean / ALL / ASLEV.Stop requested — Stop requested == true

ACTN4 ③a Traverse — /R49/Floor Service — F6

ACTN3 ②b PROJECT — Floor — F4 — stop here floors [{Floor:7}, {Floor:R}] — F5

ACTN5 ③b Select — Restriction Condition — Direction::up / dir / ALL — F2 — Floor Service.Direction — Direction — Floor Service.Direction == _up — calling floors F7 — FLOOR SERVICE [{Floor:7, Shaft: S1, Direction: _up}, {Floor:8, Shaft: S1, Direction: _up}]

ACTN6 ③c PROJECT — Floor — [{Floor:7}, {Floor:8}] — F8

ACTN7 ④a UNION — [{Floor:7}, {Floor:R}, {Floor:8}] — F9

ACTN8 ④b RENAME — FLOOR — F10 — Name >> Floor — F11 — [{Floor:L, Height: 0m}, ... {Floor:R, Height: 18m}]

ACTN9 ④c JOIN — destination heights [{Floor:7, Height:14m}, {Floor:8, Height:16m}, {Floor:R, Height:18m}, {Floor:2, Height:4m}] — F12

ACTN10 ⑤a Traverse — /R43/R28/Floor — F1 — Cabin[Shaft: S1] — F13

ACTN11 ⑤b Read — Height — Floor[{ Name: L, Height: 6.4m }] — current floor height F14 — 8m

Restriction Condition — Height / ALL — Height >= current floor height

ACTN16 ⑥ Scalar Switch Action — F2 — dir / _up — _up F19 / _down F20

ACTN12 ⑦ RESTRICT — F15

ACTN13 RANK RESTRICT — Height / least / ONE — up nearest dest — F16

ACTN14 ⑧ RESTRICT — F17

ACTN15 RANK RESTRICT — Height / greatest / ONE — down nearest dest — F18

ACTN17 EXTRACT — nearest dest — [{Floor:7, Height:14m}] — F21

ACTN19 ⑨b EXTRACT — Floor — F23

ACTN20 ⑨a Read — Cabin[Shaft: S1] — Shaft — F1 — F24 — S1

Restriction Condition — Floor, Shaft / ALL — 7

ACTN18 ⑨c Select — ASLEV — F22 — ALL — Floor, Shaft — ASLEV[{Floor:7, Shaft:S1}] — F25

Shaft S1 floor indicator:
R	18m
8	16m
7	14m
6	12m
5	10m
4	8m
3	6m
2	4m
1	2m
L	0m