

---

# Support: Domain Engineering: Rules and Regulations

---

## Rules and Regulations

For the motivation and the principles and techniques for carrying out this stage of development of domain rules and regulations description we refer to material starting on Slide 355.

### Two Informal Examples

#### Example. 1 – Trains at Stations: The “Available Station” Rule and Regulation:

- Rule:

- ★ *In China the arrival and departure of trains at, respectively from, railway stations is subject to the following rule:*
- ★ *In any three-minute interval at most one train may either arrive to or depart from a railway station.*

- Regulation:

- ★ *If it is discovered that the above rule is not obeyed, then there is some regulation which prescribes administrative or legal management and/or staff action, as well as some correction to the railway traffic.*

## [ Two Informal Examples ]

## Example. 2 – Trains Along Lines: The “Free Sector” Rule and Regulation:

- Rule:

- ★ *In many countries railway lines (between stations) are segmented into blocks or sectors. The purpose is to stipulate that if two or more trains are moving along the line, then:*
- ★ *There must be at least one free sector (i.e., without a train) between any two trains along a line.*

- Regulation:

- ★ *If it is discovered that the above rule is not obeyed, then there is some regulation which prescribes administrative or legal management and/or staff action, as well as some correction to the railway traffic.*

The above incomplete regulation will be completed later.

## Two Formal Examples

- We shall develop Example 2 on the preceding page into a partial, formal specification.
- That is, not complete, but “complete enough” for the reader to see what goes on.

[ Two Formal Examples ]

## The “Free Sector” Rule Analysis of Informal “Free Sector” Rule Text

- We start by analysing the text of the rule and regulation.
  - ★ The rule text: *There must be at least one free sector (i.e., without a train) between any two trains along a line.* contains the following terms:
    - ◇ free (a predicate),
    - ◇ sector (an entity),
    - ◇ train (an entity) and
    - ◇ line (an entity).
- We shall therefore augment our formal model to reflect these terms.
- We start by modelling
  - ★ sectors and sector descriptors,
  - ★ lines and train position descriptors,
  - ★ trains, and
  - ★ the predicate free.

[ Two Formal Examples, The “Free Sector” Rule ]

## Formalised Concepts of Sectors, Lines, and Free Sectors

**type**

$$\text{Sect}' = H \times L \times H,$$

$$\text{SectDescr} = HI \times LI \times HI$$

$$\text{Sect} = \{ |(h,l,h'):\text{Sect}' \cdot \text{obs\_HIs}(l) = \{\text{obs\_HI}(h), \text{obs\_HI}(h')\} | \}$$

$$\text{SectDescr} = \{ |(hi,li,hi'):\text{SectDescr}' \cdot \\ \exists (h,l,j'):\text{Sect} \cdot \text{obs\_HIs}(l) = \{\text{obs\_HI}(h), \text{obs\_HI}(h')\} | \}$$

$$\text{Line}' = \text{Sect}'^*,$$

$$\text{Line} = \{ | \text{line}:\text{Line}' \cdot \text{wf\_Line}(\text{line}) | \}$$

$$\text{TrnPos}' = \text{SectDescr}'^*$$

$$\text{TrnPos} = \{ | \text{trnpos}':\text{TrnPos}' \cdot \exists \text{line}:\text{Line} \cdot \text{conv\_Line\_to\_TrnPos}(\text{line}) = \text{trnpos}' | \}$$

**value**

$$\text{wf\_Line}: \text{Line}' \rightarrow \mathbf{Bool}$$

$$\text{wf\_Line}(\text{line}) \equiv$$

$$\forall i:\mathbf{Nat} \cdot \{i, i+1\} \subseteq \mathbf{inds}(\text{line}) \Rightarrow$$

$$\mathbf{let} (\_, l, h) = \text{line}(i), (h', l', \_) = \text{line}(i+1) \mathbf{in} h = h' \mathbf{end}$$

$$\text{conv\_Line\_to\_TrnPos}: \text{Line} \rightarrow \text{TrnPos}$$

$$\text{conv\_Line\_to\_TrnPos}(\text{line}) \equiv$$

$$\langle (\text{obs\_HI}(h), \text{obs\_LI}(l), \text{obs\_HI}(h')) \mid 1 \leq i \leq \mathbf{len} \text{ line} \wedge \text{line}(i) = (h, l, h') \rangle$$

[ **Two Formal Examples**, **The “Free Sector” Rule**, **Formalised Concepts of Sectors, Lines, and Free Sectors** ]

**value**

lines:  $N \rightarrow \text{Line-set}$

lines(hs,ls)  $\equiv$

**let** lns =  $\{ \langle (h,l,h') \rangle \mid h,h':H, l:L \cdot \text{proper\_line}((h,l,h'),(hs,ls)) \}$   
 $\cup \{ \text{ln} \sim \text{ln}' \mid \text{ln}, \text{ln}': \text{Line} \cdot \{ \text{ln}, \text{ln}' \} \subseteq \text{lns} \wedge \text{adjacent}(\text{ln}, \text{ln}') \}$  **in**

lns **end**

adjacent:  $\text{Line} \times \text{Line} \rightarrow \text{Bool}$

adjacent( $(\_,l,h)$ , $(h',l',\_)$ )  $\equiv h=h'$

**pre**  $\{ \text{obs\_LI}(l), \text{obs\_LI}(l') \} \subseteq \text{obs\_LIs}(h)$

[ Two Formal Examples, The “Free Sector” Rule, Formalised Concepts of Sectors, Lines, and Free Sectors ]

**type**

$$\text{TF} = \text{T} \xrightarrow{m} (\text{N} \times (\text{TN} \xrightarrow{m} \text{TrnPos}))$$

**value**

$$\text{wf\_TF}: \text{TF} \rightarrow \mathbf{Bool}$$

$$\text{wf\_TF}(\text{tf}) \equiv$$

$$\forall t:\text{T} \cdot t \in \mathbf{dom} \text{tf} \Rightarrow$$

$$\mathbf{let} ((\text{hs}, \text{ls}), \text{trnposs}) = \text{tf}(t) \mathbf{in}$$

$$\forall \text{trn}:\text{TN} \cdot \text{trn} \in \mathbf{dom} \text{trnposs} \Rightarrow$$

$$\exists \text{line}:\text{Line} \cdot \text{line} \in \text{lines}(\text{hs}, \text{ls}) \wedge$$

$$\text{trnposs}(\text{trn}) = \text{conv\_Line\_to\_TrnPos}(\text{line}) \mathbf{end}$$

- Nothing prevents two or more trains from occupying overlapping train positions.
- They have “merely” – and regrettably – crashed. But such is the domain.
- So  $\text{wf\_TF}(\text{tf})$  is not part of an axiom of traffic, merely a desirable property.

## [ Two Formal Examples, The “Free Sector” Rule, Formalised Concepts of Sectors, Lines, and Free Sectors ]

value

has\_free\_Sector:  $TN \times T \rightarrow TF \rightarrow \mathbf{Bool}$ has\_free\_Sector(trn,(hs,ls),t)(tf)  $\equiv$ 

let ((hs,ls),trnpos) = tf(t) in

(trn  $\notin$  dom trnpos  $\vee$  (tn  $\in$  dom trnpos(t)  $\wedge$  $\exists$  ln:Line  $\cdot$  ln  $\in$  lines(hs,ls)  $\wedge$ is\_prefix(trnpos(trn),ln))(hs,ls))  $\wedge$  $\sim \exists$  trn':TN  $\cdot$  trn'  $\in$  dom trnpos  $\wedge$  trn'  $\neq$  trn  $\wedge$ trnpos(trn') = conv\_Line\_to\_TrnPos( $\langle$ follow\_Sect(ln)(hs,ls) $\rangle$ )

end

pre exists\_follow\_Sect(ln)(hs,ls)

is\_prefix: Line  $\times$  Line  $\rightarrow N \rightarrow \mathbf{Bool}$ is\_prefix(ln,ln')(hs,ls)  $\equiv \exists$  ln'':Line  $\cdot$  ln''  $\in$  lines(hs,ls)  $\wedge$  ln  $\wedge$  ln'' = ln'exists\_follow\_Sect: Line  $\rightarrow$  Net  $\rightarrow \mathbf{Bool}$ exists\_follow\_Sect(ln)(hs,ls)  $\equiv$  $\exists$  ln':Line  $\cdot$  ln'  $\in$  lines(hs,ls)  $\wedge$  ln  $\wedge$  ln'  $\in$  lines(hs,ls)pre ln  $\in$  lines(hs,ls)follow\_Sect: Line  $\rightarrow$  Net  $\xrightarrow{\sim}$  Sectfollow\_Sect(ln)(hs,ls)  $\equiv$ let ln':Line  $\cdot$  ln'  $\in$  lines(hs,ls)  $\wedge$  ln  $\wedge$  ln'  $\in$  lines(hs,ls) in hd ln' endpre line  $\in$  lines(hs,ls)  $\wedge$  exists\_follow\_Sect(ln)(hs,ls)

[ Two Formal Examples, The “Free Sector” Rule ]

## Formalisation of the “Free Sector” Rule

- We doubly recursively define a function  $\text{free\_sector\_rule}(\text{tf})(\text{r})$ .
- $\text{tf}$  is that part of the traffic which has yet to be “searched” for non-free sectors.
  - ★ Thus  $\text{tf}$  is “counted” up from a first time  $\text{t}$  till the traffic  $\text{tf}$  is empty.
  - ★ That is, we assume a finite definition set  $\text{tf}$  .
- $\text{r}$  is like a traffic but without the net.
  - ★ Initially  $\text{r}$  is the empty traffic.
  - ★  $\text{r}$  is “counted” up from “earliest” cases of trains with no free sector ahead of them.
- The recursion stops, for a given time when
  - ★ there are no more train positions to be “searched” for that time;
  - ★ and when the “to-be-searched” traffic is empty.

[ **Two Formal Examples**, **The “Free Sector” Rule**, **Formalisation of the “Free Sector” Rule** ]

**type**

TNPoss = T  $\xrightarrow{m}$  (TN  $\rightarrow$  TrnPos)

**value**

free\_sector\_rule: TF  $\times$  TF  $\rightarrow$  TNPoss

free\_sector\_rule(tf)(r)  $\equiv$

**if** tf=[ ] **then** r **else**

**let** t:T·t  $\in$  **dom** tf  $\wedge$  smallest(t)(tf) **in**

**let** ((hs,ls),trnpos)=tf(t) **in**

**if** trnpos=[ ] **then** free\_sector\_rule(tf\{t})(r) **else**

**let** tn:TN·tn  $\in$  **dom** trnpos **in**

**if** exists\_follow\_Sect(trnpos(tn))(hs,ls)  $\wedge$   $\sim$ has\_free\_Sector(tn,(hs,ls),t)(tf)

**then**

**let** r' = **if** t  $\in$  **dom** r **then** r **else** r  $\cup$  [ t  $\mapsto$  [ ] ] **end in**

free\_sector\_rule(tf  $\dagger$  [ t  $\mapsto$  ((hs,ls),trnpos\{tn} ) ] )(r  $\dagger$  [ t  $\mapsto$  r(t)  $\cup$  [ tn  $\mapsto$  trnpos(tn) ] ] ) **end**

**else**

free\_sector\_rule(tf  $\dagger$  [ t  $\mapsto$  ((hs,ls),trnpos\{trn} ) ] )(r)

**end end end end end end**

smallest(t)(tf)  $\equiv$   $\sim \exists$  t':T· t' is in **dom** tf  $\wedge$  t' < t **pre** t  $\in$  **dom** tf

[ Two Formal Examples, The “Free Sector” Rule, Formalisation of the “Free Sector” Rule ]

- The rule is obeyed for a traffic  $tf$  if  $\text{free\_sector\_rule}(tf)([]) = []$ .
- Please observe that the rule is obeyed if two or more trains occupy the same sectors!
  - ★ If you do not like that, then you must state so.
  - ★ This is left as an exercise!

[ Two Formal Examples ]

## The “Free Sector” Regulation

### Completion of the “Free Sector” Regulation

- The “free sector” regulation read:
  - ★ *If it is discovered that the above rule is not obeyed,*
  - ★ then there is some regulation which prescribes administrative or legal management and/or staff action, as well as some correction to the railway traffic.
- That regulation text must be made more precise.
- Some precisions could be:
  - ★ (i) **Administrative action:** *The railway regulatory agency must establish an investigation seeking to uncover the reasons for the breach of the “free sector” rule.*

[ Two Formal Examples, The “Free Sector” Regulation, Completion of the “Free Sector” Regulation ]

- ★ (ii) Legal action: *If the railway regulatory agency finds that a potentially punishable staff conduct has occurred then the public prosecutor must be notified and given all investigation material.*
- ★ (iii) Current traffic correction: *As soon as it has been established that a train has progressed into a non-free sector that train must be stopped and the train ahead of it must be informed of the situation.*
- ★ (iv) Future traffic corrections: *If the railway regulatory agency finds that the ‘business processes’ could be improved then the rail and train operators are asked to improve their train traffic handling procedures.*
- Usually all of these parts are present in the regulation.

[ Two Formal Examples, The “Free Sector” Regulation ]

## Analysis of the Completed “Free Sector” Regulation

- The nature of part regulations (i, ii, iv) is such that they cannot be formalised.
- Part regulation (iii) can be formalised:
  - ★ (iii.A) the offending, the “rear”, train must be stopped,
  - ★ (iii.B) possible “follower” trains must presumably be informed or stopped, and
  - ★ (iii.C) the “ahead” train must be informed.
- We omit any formalisation.
- The regulation statements (iii.A–.C) amount to management actions.

# Review

Dines Bjørner: 8th DRAFT: October 14, 2008

---

**End of Support: Domain Engineering: Rules and Regulations**

---