Dines Bjørner's MAP-i Lecture # 8

Domain Requirements: Instantiation and Determination

Tuesday, 26 May 2015: 16:45–17:30

7.2.2. Domain Instantiation

Definition 28 . Instantiation: *By* **domain instantiation** *we mean*

- *a refinement of the partial domain requirements prescription,*
- *resulting from the projection step,*
- *in which the refinements aim at rendering the*
  - *endurants:*
    - *parts,*
    - *materials and*
    - *components,*

    as well as the
  - *perdurants:*
    - *actions,*
    - *events and*
    - *behaviours*

  of the domain requirements prescription
- *more concrete, more specific* ■

- Refinement of endurants can be expressed
  - either in the form of concrete types,
  - or of further “delineating” axioms over sorts,
  - or of a combination of concretisation and axioms.
- We shall exemplify the third possibility.
- Examples 77–78 express requirements that the road net on which the road-pricing system is to be based must satisfy.

7.2.2.1. Domain Instantiation — Narrative

Example 77 . Domain Requirements. Instantiation Road Net, Narrative:

- We now require that there is, as before, a road net, $n_{\mathcal{I}}$:$N_{\mathcal{I}}$, which can be understood as consisting of two, "connected sub-nets".
  - A toll-road net, $trn_{\mathcal{I}}$:$TRN_{\mathcal{I}}$, cf. Fig. 3 on the facing slide,
  - and an ordinary road net, n'_{Δ}.
  - The two are connected as follows:
    - The toll-road net, $trn_{\mathcal{I}}$, borders some toll-road plazas, in Fig. 3 on the next slide shown by white filled circles (i.e., hubs).
    - These toll-road plaza hubs are proper hubs of the 'ordinary' road net, n'_{Δ}.

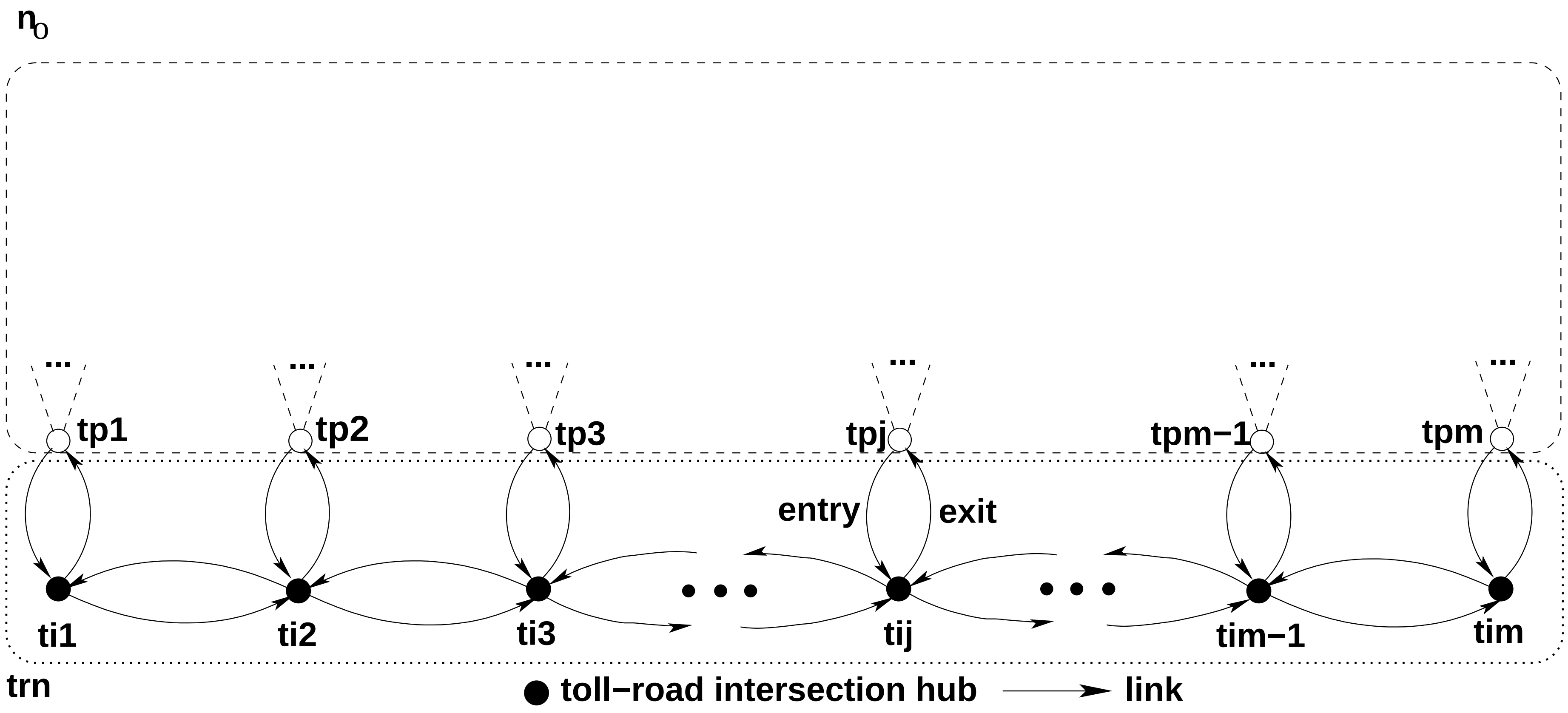

Figure 3: A simple, linear toll-road net

164 The instantiated domain, $\delta_{\mathcal{I}}$:$\Delta_{\mathcal{I}}$ has just the net, $\mathsf{n}_{\mathcal{I}}$:$\mathsf{N}_{\mathcal{I}}$ being instantiated.

165 The road net consists of two "sub-nets"

a. an "ordinary" road net, n'_{Δ}:N'_{Δ} and

b. a toll-road net proper, $\mathsf{trn}_{\mathcal{I}}$:$\mathsf{TRN}_{\mathcal{I}}$ —

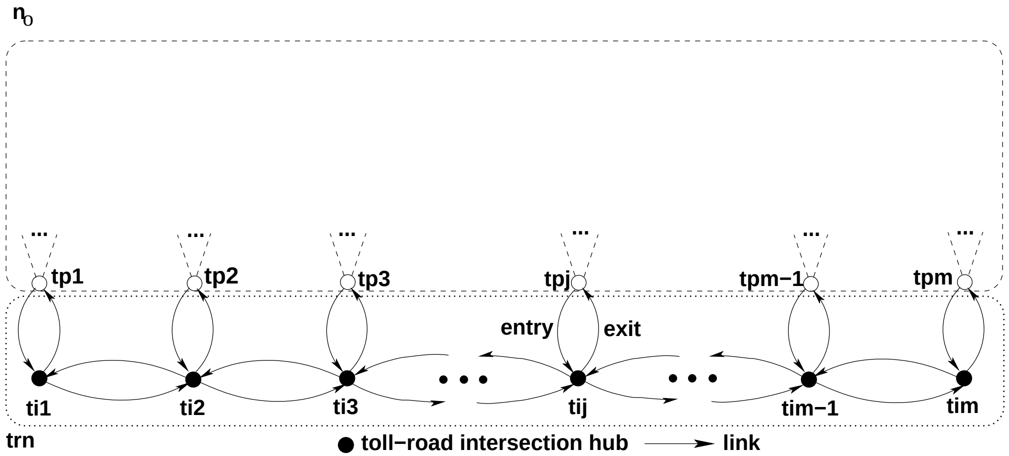

Figure 4: The Instantiated Road Net

c. "connected" by an interface hil:HIL:

i That interface consists of a number of toll-road plazas (i.e., hubs), modeled as a list of hub identifiers, hil:HI*.

ii The toll-road plaza interface to the toll-road net, trn:TRN$_{\mathcal{I}}$[26], has each plaza, hil[i], connected to a pair of toll-road links: an entry and an exit link: $(l_e{:}L, l_x{:}L)$.

iii The toll-road plaza interface to the 'ordinary' net, n$'_\Delta$:N$'_\Delta$, has each plaza, i.e., the hub designated by the hub identifier hil[i], connected to one or more ordinary net links, $\{l_{i_1}, l_{i_2}, \cdots, l_{i_\ell}\}$.

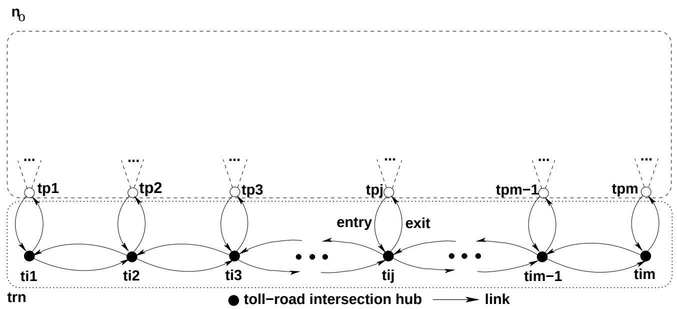

Figure 5: The Instantiated Road Net

[26]We (sometimes) omit the subscript $_{\mathcal{I}}$ when it should be clear from the context what we mean.

165b. The toll-road net, trn:TRN$_{\mathcal{I}}$, consists of three collections (modeled as lists) of links and hubs:

i a list of pairs of toll-road entry/exit links: $\langle(l_{e_1},l_{x_1}),\cdots,(l_{e_\ell},l_{x_\ell})\rangle$,

ii a list of toll-road intersection hubs: $\langle h_{i_1},h_{i_2},\cdots,h_{i_\ell}\rangle$, and

iii a list of pairs of main toll-road ("up" and "$down$") links: $\langle(ml_{i_{1u}},$-$ml_{i_{1d}}),(m_{i_{2u}},m_{i_{2d}}),\cdots,(m_{i_{\ell u}},m_{i_{\ell d}})\rangle$.

d. The three lists have commensurate lengths.

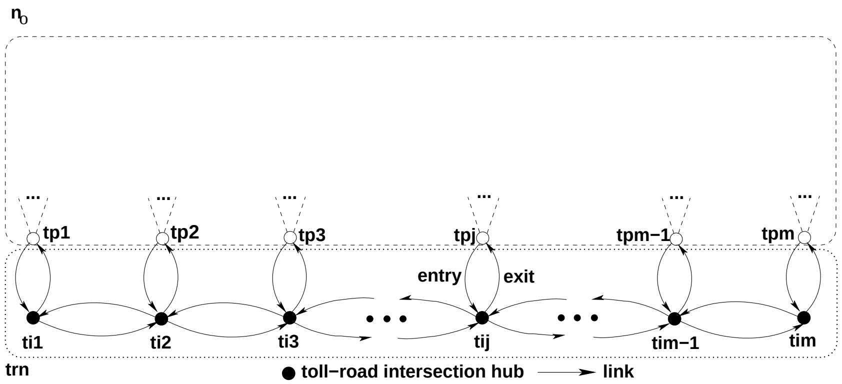

Figure 6: The Instantiated Road Net

7.2.2.2. Domain Instantiation — Formalisation

Example 78 . Domain Requirements. Instantiation Road Net, Formal Types:

type

164 $\Delta_\mathcal{I}$

165 $\mathsf{N}_\mathcal{I} = \mathsf{N}'_\Delta \times \mathsf{HIL} \times \mathsf{TRN}$

165a. N'_Δ

165b. $\mathsf{TRN}_\mathcal{I} = (\mathsf{L}\times\mathsf{L})^* \times \mathsf{H}^* \times (\mathsf{L}\times\mathsf{L})^*$

165c. $\mathsf{HIL} = \mathsf{HI}^*$

axiom

165d. $\forall\ \mathsf{n}_\mathcal{I}:\mathsf{N}_\mathcal{I}\ \bullet$

165d. $\quad$ **let** $(\mathsf{n}_\Delta,\mathsf{hil},(\mathsf{exll},\mathsf{hl},\mathsf{lll})) = \mathsf{n}_\mathcal{I}$ **in**

165d. $\quad$ **len** hil = **len** exll = **len** hl = **len** lll + 1

165d. $\quad$ **end**

[Lecturer explains N'_Δ]

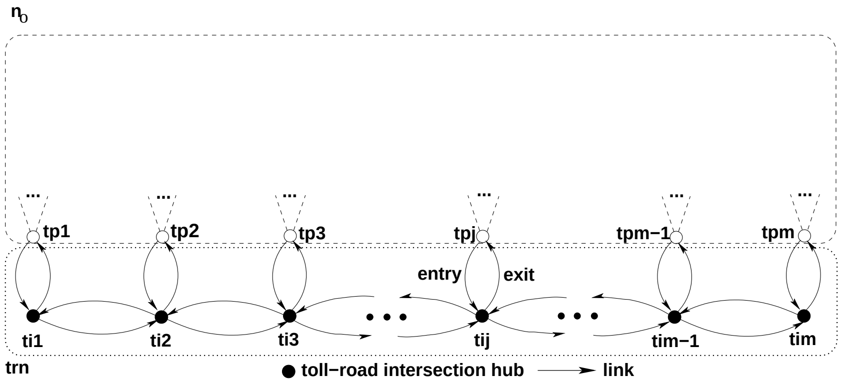

Figure 7: The Instantiated Road Net

7.2.2.3. Domain Instantiation — Formalisation: Well-formedness

Example 79 . Domain Requirements. Instantiation Road Net, Well-formedness:

- The partial concretisation of the net sorts, N, into $N_{\mathcal{R}_1}$ requires some well-formedness conditions to be satisfied.

166 The toll-road intersection hubs must all have distinct hub identifiers.

value
166. wf_dist_toll_road_isect_hub_ids: $H^* \rightarrow$ **Bool**
166. wf_dist_toll_road_isect_hub_ids(hl) $\equiv$
166. **len** hl = **card** xtr_his(hl)

167 The toll-road 'up' and 'down' links must all have distinct link identifiers.

value
167. wf_dist_toll_road_u_d_link_ids: $(L \times L)^* \rightarrow$ **Bool**
167. wf_dist_toll_road_u_d_link_ids(lll) $\equiv$
167. $2 \times$ **len** lll = **card** xtr_lis(lll)

168 The toll-road entry/exit links must all have distinct link identifiers.

value
168. wf_dist_e_x_link_ids: $(\mathsf{L}\times\mathsf{L})^*\to\mathbf{Bool}$
168. wf_dist_e_x_link_ids(exll) $\equiv$
168. $\quad 2 \times$ **len** exll $=$ **card** xtr_lis(exll)

169 Proper net links must not designate toll-road intersection hubs.

value
169. wf_isoltd_toll_road_isect_hubs: $\mathsf{HI}^*\times\mathsf{H}^*\to\mathsf{N}_{\mathcal{I}}\to\mathbf{Bool}$
169. wf_isoltd_toll_road_isect_hubs(hil,hl)($\mathsf{n}_{\mathcal{I}}$) $\equiv$
169. $\quad$ **let** ls=xtr_links($\mathsf{n}_{\mathcal{I}}$) **in**
169. $\quad$ **let** his $= \cup$ {**obs_mereo**_L(l)|l:L•l $\in$ ls} **in**
169. $\quad$ his $\cap$ xtr_his(hl) $=$ {} **end end**

170 The plaza hub identifiers must designate hubs of the 'ordinary' net.

value
170. $\mathsf{wf\_p\_hubs\_pt\_of\_ord\_net}$: $\mathsf{HI}^* \rightarrow \mathsf{N}'_\Delta \rightarrow \mathbf{Bool}$
170. $\mathsf{wf\_p\_hubs\_pt\_of\_ord\_net(hil)(n'_\Delta)} \equiv$
170. $\quad\quad \mathbf{elems}\ \mathsf{hil} \subseteq \mathsf{xtr\_his}(\mathsf{n}'_\Delta)$

171 The plaza hub mereologies must each,

a. besides identifying at least one hub of the ordinary net,
b. also identify the two entry/exit links with which they are supposed to be connected.

value
171. $\mathsf{wf\_p\_hub\_interf}$: $\mathsf{N}'_\Delta \rightarrow \mathbf{Bool}$
171. $\mathsf{wf\_p\_hub\_interf(n_o,hil,(exll,\_,\_))} \equiv$
171. $\quad \forall\ \mathsf{i}:\mathbf{Nat} \cdot \mathsf{i} \in \mathbf{inds}\ \mathsf{exll} \Rightarrow$
171. $\quad\quad \mathbf{let}\ \mathsf{h} = \mathsf{get\_H(hil(i))(n'_\Delta)}\ \mathbf{in}$
171. $\quad\quad \mathbf{let}\ \mathsf{lis} = \mathbf{obs\_mereo}\mathsf{\_H(h)}\ \mathbf{in}$
171. $\quad\quad \mathbf{let}\ \mathsf{lis'} = \mathsf{lis} \setminus \mathsf{xtr\_lis(n')}\ \mathbf{in}$
171. $\quad\quad \mathsf{lis'} = \mathsf{xtr\_lis(exll(i))}\ \mathbf{end\ end\ end}$

172 The mereology of each toll-road intersection hub must identify

a. the entry/exit links

b. and exactly the toll-road ‘up’ and ‘down’ links

c. with which they are supposed to be connected.

```
value
172.   wf_toll_road_isect_hub_iface: N_I→Bool
172.   wf_toll_road_isect_hub_iface(_,_,(exll,hl,lll)) ≡
172.      ∀ i:Nat • i ∈ inds hl ⇒
172.         obs_mereo_H(hl(i)) =
172a..        xtr_lis(exll(i)) ∪
172.         case i of
172b..            1 → xtr_lis(lll(1)),
172b..            len hl → xtr_lis(lll(len hl−1))
172b..            _ → xtr_lis(lll(i)) ∪ xtr_lis(lll(i−1))
172.         end
```

173 The mereology of the entry/exit links must identify exactly the

a. interface hubs and the

b. toll-road intersection hubs

c. with which they are supposed to be connected.

value

173. wf_exll: $(L \times L)^* \times HI^* \times H^* \rightarrow$ **Bool**
173. wf_exll(exll,hil,hl) $\equiv$
173. $\forall$ i:**Nat** $\cdot$ i $\in$ **len** exll
173. **let** (hi,(el,xl),h) = (hil(i),exll(i),hl(i)) **in**
173. **obs_mereo**_L(el) = **obs_mereo**_L(xl)
173. = {hi} $\cup$ {**uid**_H(h)} **end**
173. **pre**: **len** eell = **len** hil = **len** hl

174 The mereology of the toll-road ‘up’ and ‘down’ links must

a. identify exactly the toll-road intersection hubs

b. with which they are supposed to be connected.

value

174. wf_u_d_links: $(\mathsf{L}\times\mathsf{L})^*\times\mathsf{H}^*\to\mathbf{Bool}$

174. wf_u_d_links(lll,hl) $\equiv$

174. $\forall$ i:**Nat** $\cdot$ i $\in$ **inds** lll $\Rightarrow$

174. **let** (ul,dl) = lll(i) **in**

174. **obs_mereo**_L(ul) = **obs_mereo**_L(dl) =

174a.. **uid**_H(hl(i)) $\cup$ **uid**_H(hl(i+1)) **end**

174. **pre**: **len** lll = **len** hl+1

- We have used additional auxiliary functions:

value

$\mathsf{xtr\_his}$: $\mathsf{H}^* \rightarrow \mathsf{HI}\textbf{-set}$
$\mathsf{xtr\_his(hl)} \equiv \{\textbf{uid}\_\mathsf{HI(h)}|\mathsf{h{:}H}\cdot\mathsf{h} \in \textbf{elems}\ \mathsf{hl}\}$
$\mathsf{xtr\_lis}$: $(\mathsf{L}\times\mathsf{L})\rightarrow\mathsf{LI}\textbf{-set}$
$\mathsf{xtr\_lis(l',l'')} \equiv \{\textbf{uid}\_\mathsf{LI(l')}\}\cup\{\textbf{uid}\_\mathsf{LI(l'')}\}$
$\mathsf{xtr\_lis}$: $(\mathsf{L}\times\mathsf{L})^* - \mathsf{LI}\textbf{-set}$
$\mathsf{xtr\_lis(lll)} \equiv$
$\cup\{\mathsf{xtr\_lis(l',l'')}|\mathsf{(l',l''){:}(L\times L)}\cdot\mathsf{(l',l'')}\in \textbf{elems}\ \mathsf{lll}\}$

7.2.2.3.1 Summary Well-formedness Predicate

175 The well-formedness of instantiated nets is now the conjunction of the individual well-formedness predicates above.

value

175. wf_instantiated_net: $\mathsf{N}_{\mathcal{I}} \rightarrow \mathbf{Bool}$
175. wf_instantiated_net(n'_Δ,hil,(exll,hl,lll))
166. wf_dist_toll_road_isect_hub_ids(hl)
167. $\wedge$ wf_dist_toll_road_u_d_link_ids(lll)
168. $\wedge$ wf_dist_e_e_link_ids(exll)
169. $\wedge$ wf_isolated_toll_road_isect_hubs(hil,hl)(n′)
170. $\wedge$ wf_p_hubs_pt_of_ord_net(hil)(n′)
171. $\wedge$ wf_p_hub_interf(n'_Δ,hil,(exll,_,_))
172. $\wedge$ wf_toll_road_isect_hub_iface(_,_,(exll,hl,lll))
173. $\wedge$ wf_exll(exll,hil,hl)
174. $\wedge$ wf_u_d_links(lll,hl)

7.2.2.4. Domain Instantiation — Abstraction

Example 80 . Domain Requirements. Instantiation Road Net, Abstraction:

- Domain instantiation has refined
  - an abstract definition of net sorts, n_Δ:N_Δ,
  - into a partially concrete definition of nets, $n_\mathcal{I}$:$N_\mathcal{I}$.
- We need to show the refinement relation:
  - abstraction($n_\mathcal{I}$) $=$ n_Δ.

value

176 abstraction: $\mathsf{N}_{\mathcal{I}} \rightarrow \mathsf{N}_{\Delta}$

177 abstraction(n'_{Δ},hil,(exll,hl,lll)) $\equiv$

178 **let** n_{Δ}:N_{Δ} $\cdot$

178 **let** hs = **obs_part**_HS_{Δ}(**obs_part**_HA_{Δ}(n'_{Δ})),

178 ls = **obs_part**_LS_{Δ}(**obs_part**_LA_{Δ}(n'_{Δ})),

178 ths = **elems** hl,

178 eells = xtr_links(eell), llls = xtr_links(lll) **in**

179 hs$\cup$ths=**obs_part**_HS_{Δ}(**obs_part**_HA_{Δ}(n_{Δ}))

180 $\wedge$ ls$\cup$eells$\cup$llls=**obs_part**_LS_{Δ}(**obs_part**_LA_{Δ}(n_{Δ}))

181 n_{Δ} **end end**

176 The abstraction function takes a concrete net, $\mathsf{n}_{\mathcal{I}}$:$\mathsf{N}_{\mathcal{I}}$, and yields an abstract net, n_{Δ}:N_{Δ}.

177 The abstraction function doubly decomposes its argument into constituent lists and sub-lists.

178 There is postulated an abstract net, n_{Δ}:N_{Δ}, such that

179 the hubs of the concrete net and toll-road equals those of the abstract net, and

180 the links of the concrete net and toll-road equals those of the abstract net.

181 And that abstract net, n_{Δ}:N_{Δ}, is postulated to be an abstraction of the concrete net.

7.2.2.5. An Instantiation Operator

- Domain instantiation take a requirements prescription, $\mathcal{R}_{\mathcal{P}}$, and yields a more **concrete requirements prescription** $\mathcal{R}_{\mathcal{I}}$.
  - **type instantiation**: $\mathcal{R}_{\mathcal{P}} \to \mathcal{R}_{\mathcal{I}}$
- Semantically
  - $\mathcal{R}_{\mathcal{P}}$ denotes a possibly infinite set of meanings, say $\mathbb{R}_{\mathbb{P}}$,
  - $\mathcal{R}_{\mathcal{I}}$ denotes a possibly infinite set of meanings, say $\mathbb{R}_{\mathbb{I}}$ and
  - such that some relation $\mathbb{R}_{\mathbb{I}} \sqsubseteq \mathbb{R}_{\mathbb{P}}$ is satisfied.

7.2.3. Domain Determination

Definition 29 . Determination: *By* **domain determination** *we mean*

- *a refinement of the partial domain requirements prescription,*
- *resulting from the instantiation step,*
- *in which the refinements aim at rendering the*
  - *endurants:*
    - *parts,*
    - *materials and*
    - *components, as well as the*
  - *perdurants:*
    - *functions,*
    - *events and*
    - *behaviours*

  of the partial domain requirements prescription
- *less non-determinate, more determinate.* ■

- Determinations usually render these concepts less general.
  - That is, the value space
    - of endurants that are made more determinate
    - is "smaller", contains fewer values,
    - as compared to the endurants
      before determination has been "applied".

7.2.3.1. Domain Determination: Example

- We show an example of 'domain determination'.
  - It is expressed sôlely in terms of
  - axioms over the concrete toll-road net type.

Example 81 . Domain Requirements. Determination Toll-roads: 7.2.3

- We focus only on the toll-road net.
- We single out only two 'determinations':

182 *The entry/exit and toll-road links*

a. are always all one way links,

b. as indicated by the arrows of Fig. 2,

c. such that each pair allows traffic in opposite directions.

value

182. opposite_traffics: $(L \times L)^* \times (L \times L)^* \rightarrow$ **Bool**
182. opposite_traffics(exll,lll) $\equiv$
182. $\forall$ (lt,lf):$(L \times L)$ $\cdot$ (lt,lf) $\in$ **elems** exll$\frown$lll $\Rightarrow$
182a.. **let** (ltσ,lfσ) = (**attr**_LΣ(lt),**attr**_LΣ(lf)) **in**
182a.$'$. **attr**_LΩ(lt)=$\{$lt$\sigma\}\wedge$**attr**_LΩ(ft)=$\{$ft$\sigma\}$
182a.$''$. $\wedge$ **card** ltσ = 1 = **card** lfσ
182. $\wedge$ **let** ($\{$(hi,hi$'$)$\}$,$\{$(hi$''$,hi$'''$)$\}$) = (ltσ,lfσ) **in**
182c.. hi=hi$'''$ $\wedge$ hi$'$=hi$''$
182. **end end**

7.2.3.1.2 All Toll-road Hubs are Free-flow

183 *The hub state spaces* are singleton sets of the toll-road hub states which always allow exactly these (and only these) crossings:

a. from *e*ntry links back to the paired e*x*it *l*ink*s*,

b. from *e*ntry links to emanating *t*oll-road *l*ink*s*,

c. from incident *t*oll-road links to e*x*it *l*ink*s*, and

d. from incident *t*oll-road link to emanating *t*oll-road *l*ink*s*.

value

183. free_flow_toll_road_hubs: $(L \times L)^* \times (L \times L)^* \rightarrow$ **Bool**
183. free_flow_toll_road_hubs(exl,ll) $\equiv$
183. $\forall$ i:**Nat**•i $\in$ **inds** hl $\Rightarrow$
183. **attr**_HΣ(hl(i)) =
183a.. hσ_ex_ls(exl(i))
183b.. $\cup$ hσ_et_ls(exl(i),(i,ll))
183c.. $\cup$ hσ_tx_ls(exl(i),(i,ll))
183d.. $\cup$ hσ_tt_ls(i,ll)

183a.: from *e*ntry links back to the paired e*x*it *l*ink*s*:

value

183a.. hσ_ex_ls: (L$\times$L)$\rightarrow$LΣ

183a.. hσ_ex_ls(e,x) $\equiv$ {(**uid**_LI(e),**uid**_LI(x))}

183b.: from *e*ntry links to emanating *t*oll-road *l*ink*s*:

value

183b.. hσ_et_ls: (L×L)×(**Nat**×(em:L×in:L)*)→LΣ
183b.. hσ_et_ls((e,_),(i,ll)) ≡
183b.. **case** i **of**
183b.. 2 → {(**uid**_Ll(e),**uid**_Ll(em(ll(1))))},
183b.. **len** ll+1 → {(**uid**_Ll(e),**uid**_Ll(em(ll(**len** ll))))},
183b.. _ → {(**uid**_Ll(e),**uid**_Ll(em(ll(i−1)))),
183b.. (**uid**_Ll(e),**uid**_Ll(em(ll(i))))}
183b.. **end**

- The *em* and *in* in the toll-road link list (em:L×in:L)* designate selectors for *em*anating, respectively *in*cident links.

183c.: from incident *t*oll-road links to e*x*it *l*ink*s*:

value

183c.. $\mathsf{h}\sigma$_tx_ls: (L×L)×(**Nat**×(em:L×in:L)*)→LΣ
183c.. $\mathsf{h}\sigma$_tx_ls((_,x),(i,ll)) ≡
183c.. **case** i **of**
183c.. 2 → {(**uid**_Ll(in(ll(1))),**uid**_Ll(x))},
183c.. **len** ll+1 → {(**uid**_Ll(in(ll(**len** ll))),**uid**_Ll(x))},
183c.. _ → {(**uid**_Ll(in(ll(i−1))),**uid**_Ll(x)),
183c.. (**uid**_Ll(in(ll(i))),**uid**_Ll(x))}
183c.. **end**

183d.: from incident $t$oll-road link to emanating $t$oll-road $l$ink$s$:

value

183d.. hσ_tt_ls: **Nat**$\times$(em:L$\times$in:L)*$\rightarrow$LΣ
183d.. hσ_tt_ls(i,ll) $\equiv$
183d.. **case** i **of**
183d.. 2 $\rightarrow$ {(**uid**_LI(in(ll(1))),**uid**_LI(em(ll(1))))},
183d.. **len** ll+1 $\rightarrow$ {(**uid**_LI(in(ll(**len** ll))),**uid**_LI(em(ll(**len** ll))))},
183d.. _ $\rightarrow$ {(**uid**_LI(in(ll(i$-$1))),**uid**_LI(em(ll(i$-$1)))),
183d.. (**uid**_LI(in(ll(i))),**uid**_LI(em(ll(i))))}
183d.. **end**

7.2.3.2. A Domain Determination Operator

- Domain determination take a requirements description, $\mathcal{R}_\mathcal{I}$, and yields a more **deterministic requirements prescription**, $\mathcal{R}_\mathcal{D}$.
  - **type instantiation**: $\mathcal{R}_\mathcal{I} \rightarrow \mathcal{R}_\mathcal{D}$
- Semantically
  - $\mathcal{R}_\mathcal{I}$ denotes a possibly infinite set of meanings, say $\mathbb{R}_\mathbb{I}$,
  - $\mathcal{R}_\mathcal{D}$ denotes a possibly infinite set of meanings, say $\mathbb{R}_\mathbb{D}$ and
  - such that some relation $\mathbb{R}_\mathbb{I} \sqsubseteq \mathbb{R}_\mathbb{D}$ is satisfied.

Dines Bjørner's MAP-i Lecture # 8

End of MAP-i Lecture # 8: Domain Requirements: Instantiation and Determination

Tuesday, 26 May 2015: 16:45–17:30