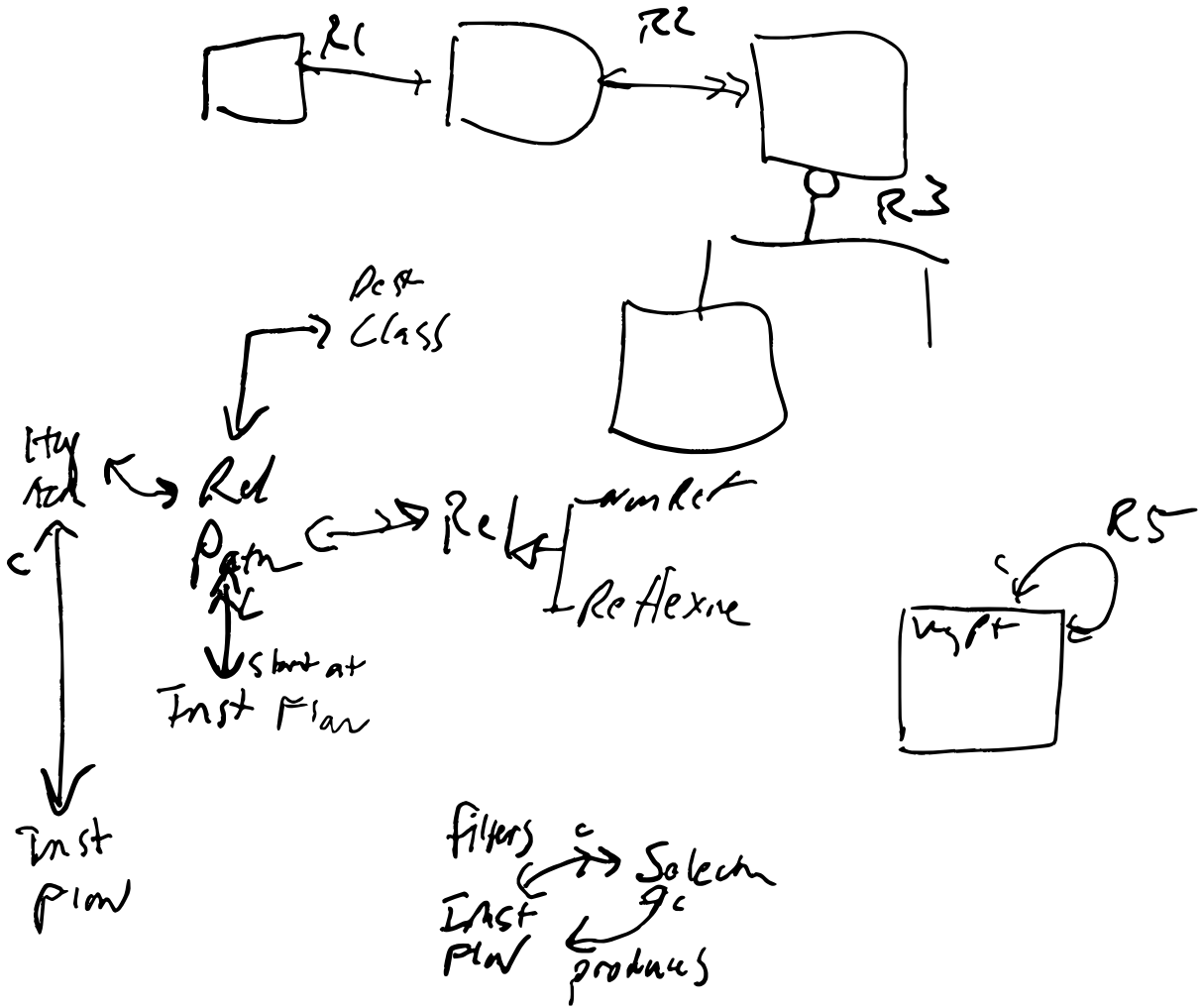
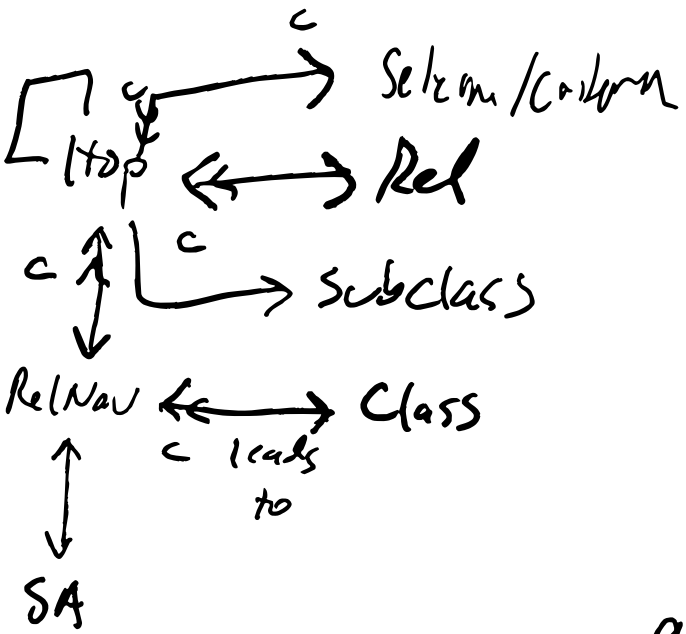
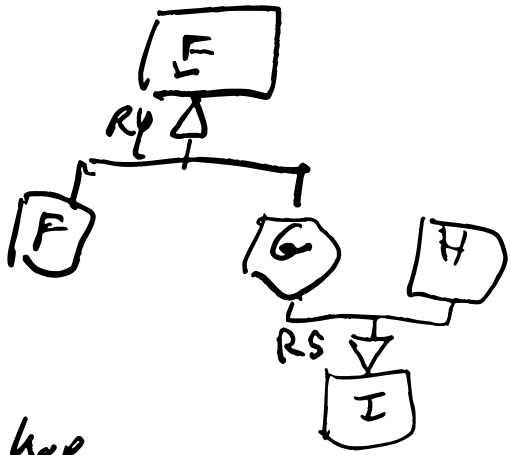
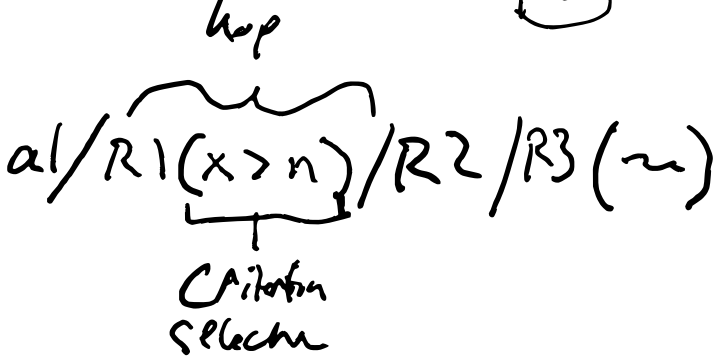
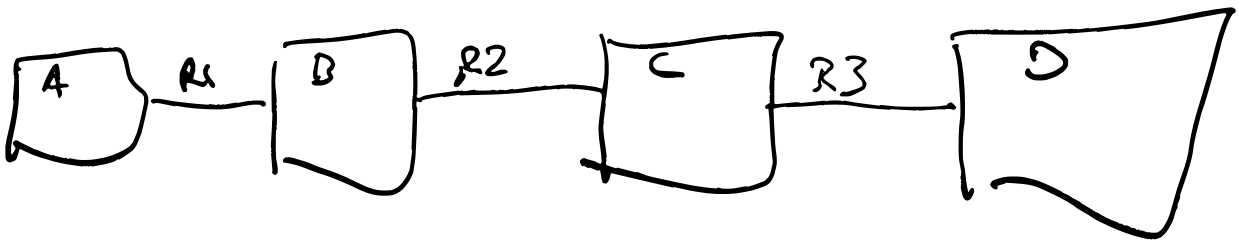
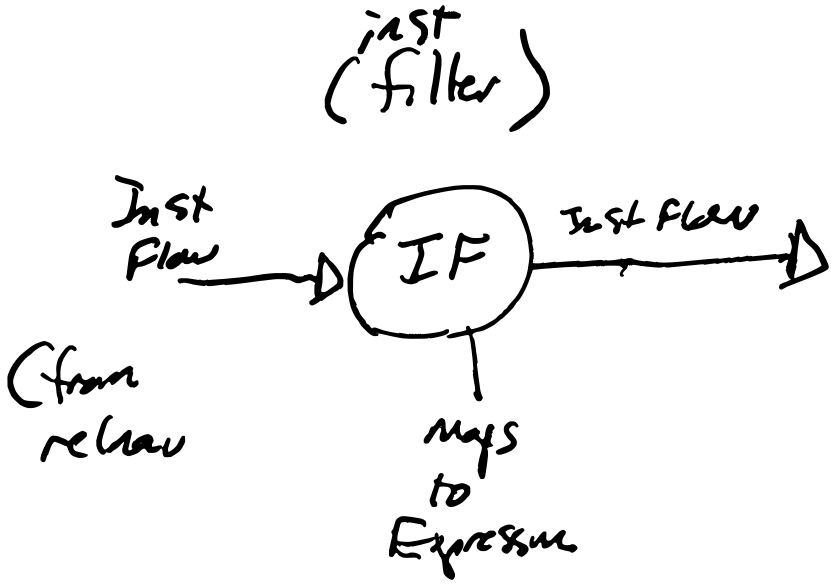

R1
R2
R3
Desc
Class
Rel
Path
Rel
number
Relexive
start at
Inst Plan
Inst
Plan
R5
filters
Selects
Inst
Plan
produces

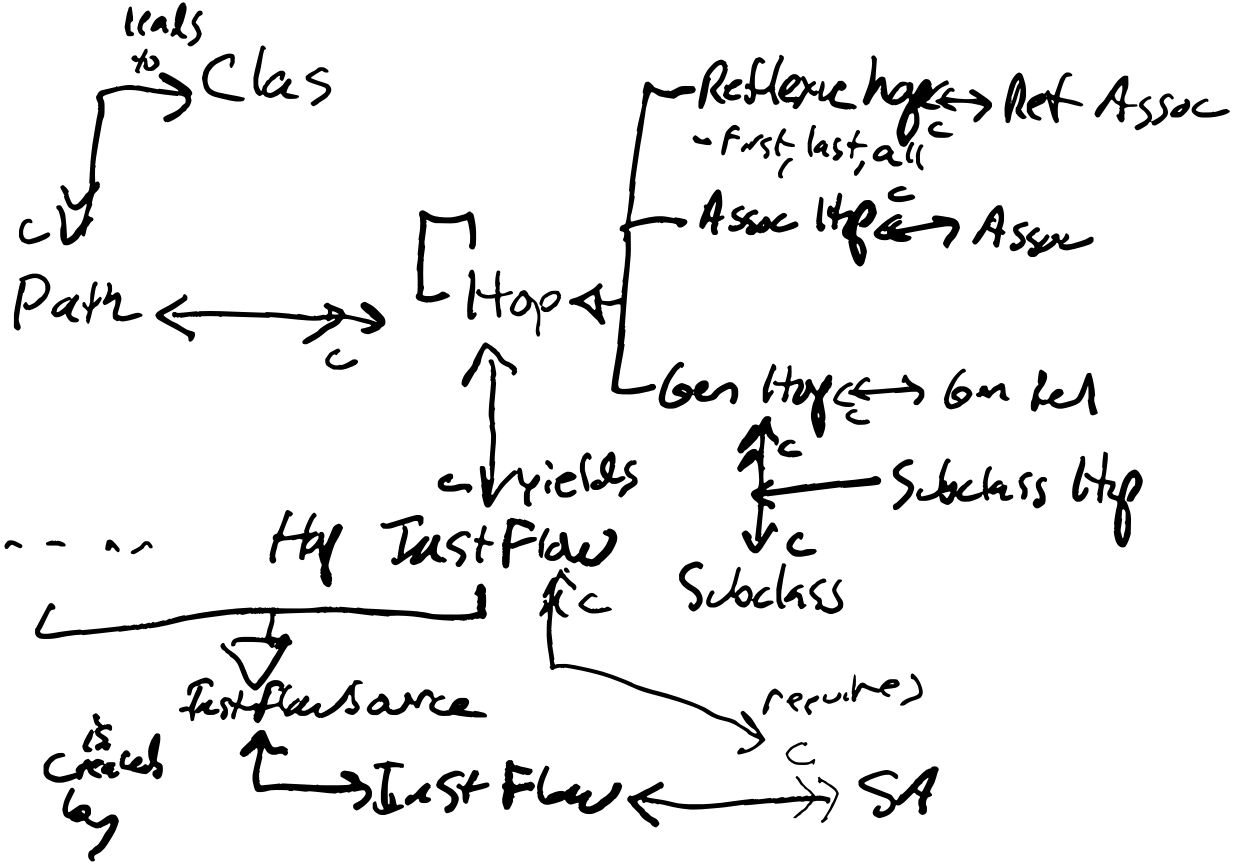

c
c
Selem/Column
Hop
Rel
c
c
Subclass
RelNav
Class
c leads to
SA

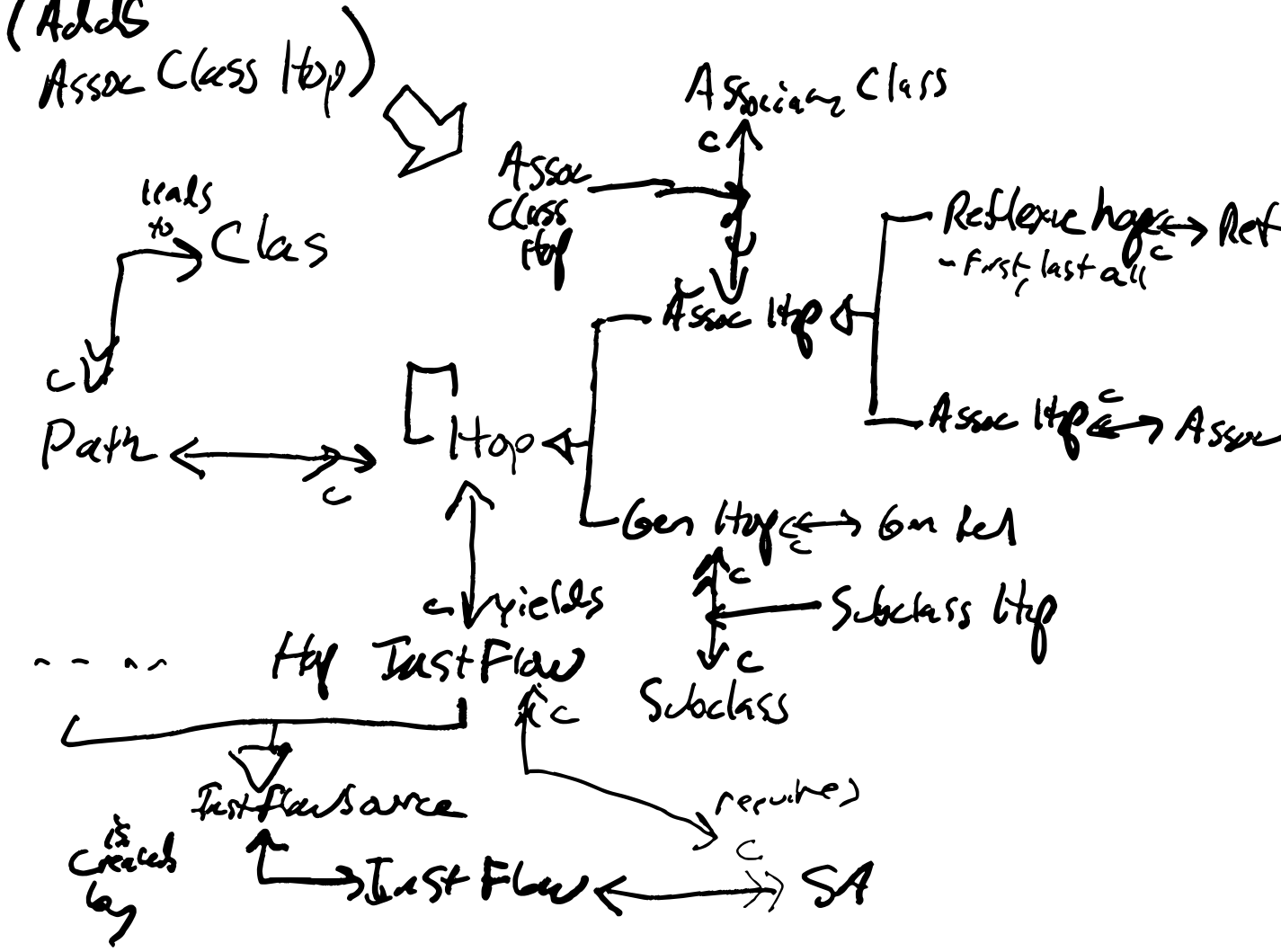

E
Ry
F
G
H
RS
I


Hop
a1/R1(x>n)/R2/R3(~)
Criterion
Selection

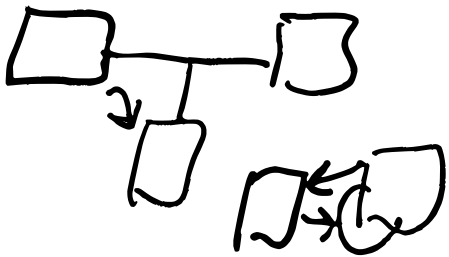

A
R1
B
R2
C
R3
D


inst
(filter)
Inst
Flow
IF
Inst Flow
(from
relay
maps
to
Expression

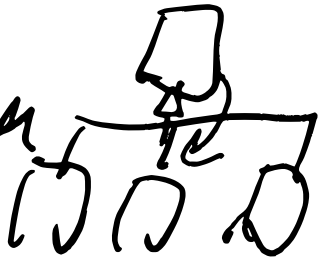

leals to
Clas
c
Path
c
Htop
Reflexive Hop
Ref Assoc
- first, last, all
Assoc Htop
Assoc
Gen Htop
Gen Ref
c
yields
Subclass Htop
Hq TestFlow
Subclass
TestFlowSource
is created by
TestFlow
SA
requires

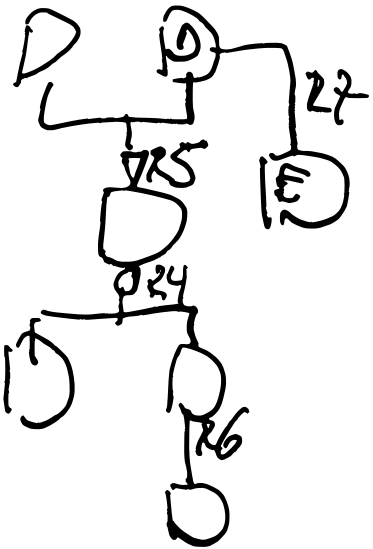

(Adds
Assoc Class Htop)
Associate Class
Assoc Class Htop
leads to
Class
Reflexive hops
- first, last all
Assoc Htop
Path
Htop
Assoc Htop
Gen Htop
Subclass Htop
yields
Htop InstFlow
Subclass
InstFlowSource
requires
is created by
InstFlow
SA

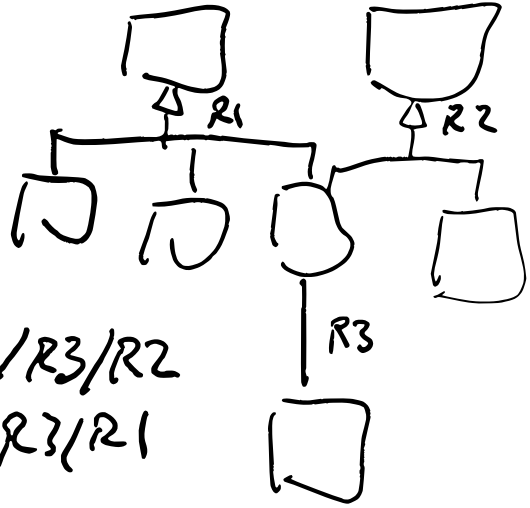
class
Assoc Branch



Subclass Branch







Dest

/R3/R2

/R3/R1

/R6/R4/R5/D/R7/E

← Sub branch

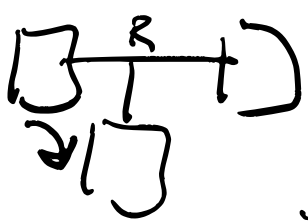
A Path starts with an input instance flow → then an optional sequence of Hops, leading to a required Destination Class.

There are different kinds of Hops:

(1) Straight Hop

Across an association to a participating class (not the assoc class formalizing the hopped Association)

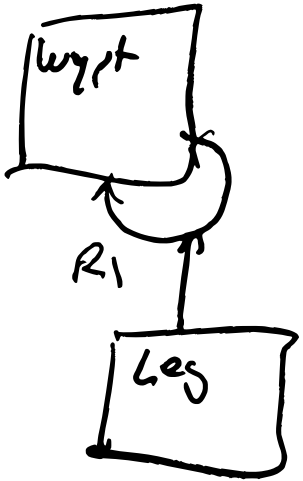
(2) AssocClass Hop



Across an association to the assoc class formalizing that association

(3) Subclass Hop

Across a gen relationship to a subclass of that same relationship



Get all waypoints from wl where (~)

wpts.. = wl/R1/~*/(:ts.Alt>h)

(all)

Get all legs where ...

Rule: Destalog determines type of assignment

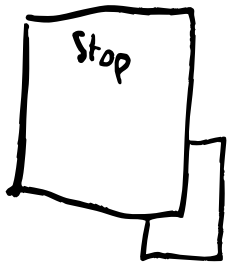
legs.. = wl/R1/~*/(~)/Leg — Destalog

SkipSpec | Assoc Branch | Path

getallLegswhere

/R1/~/(:ts.Alt>h)/Leg

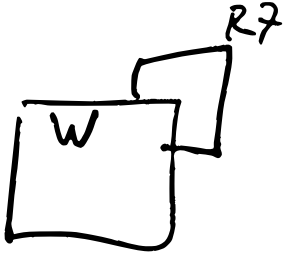
Asso Branch



ordinal

(no assoc class possible)

Reflexive Hops



keep hopping until criteria is satisfied

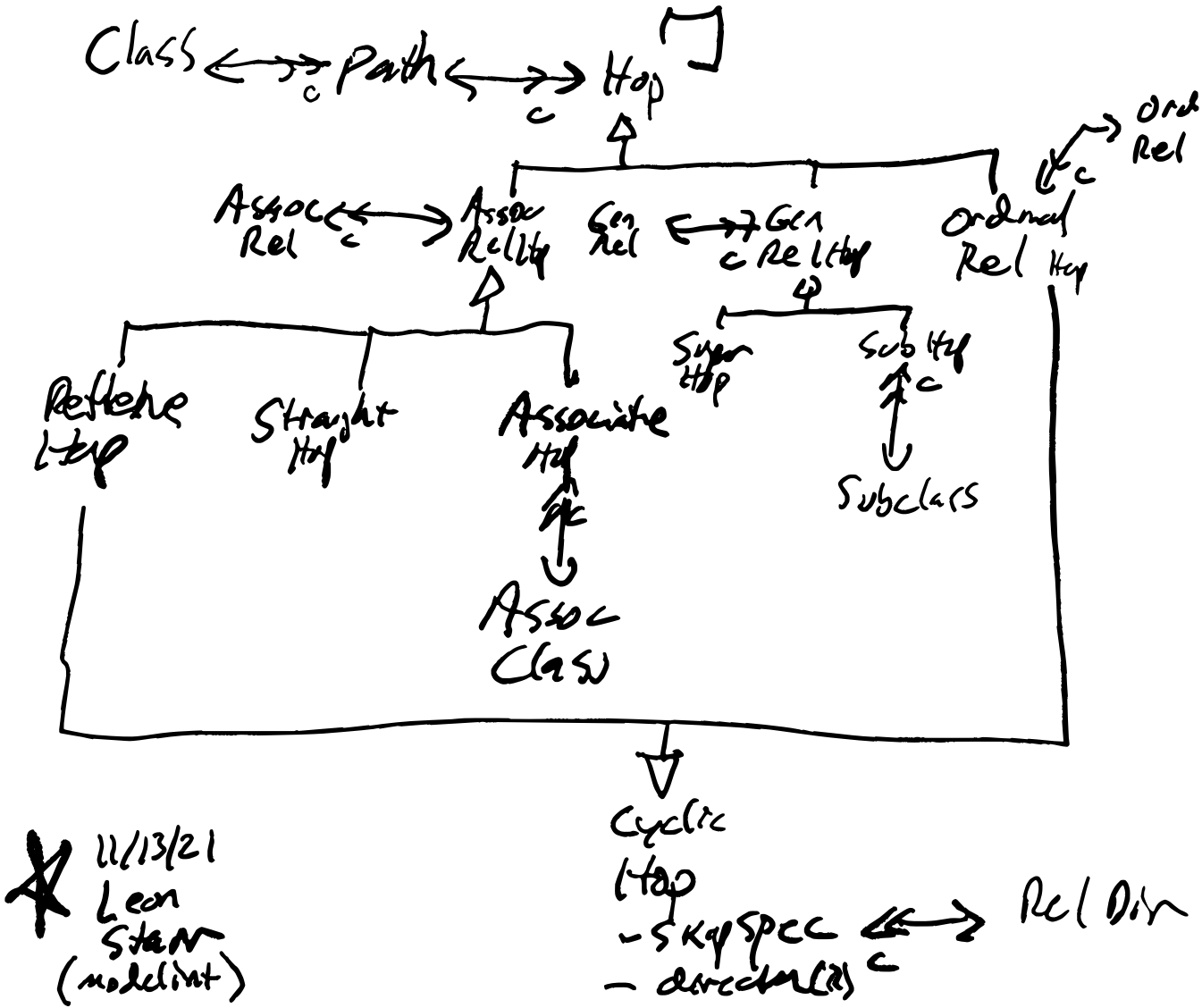
/R7/~/(~~~)

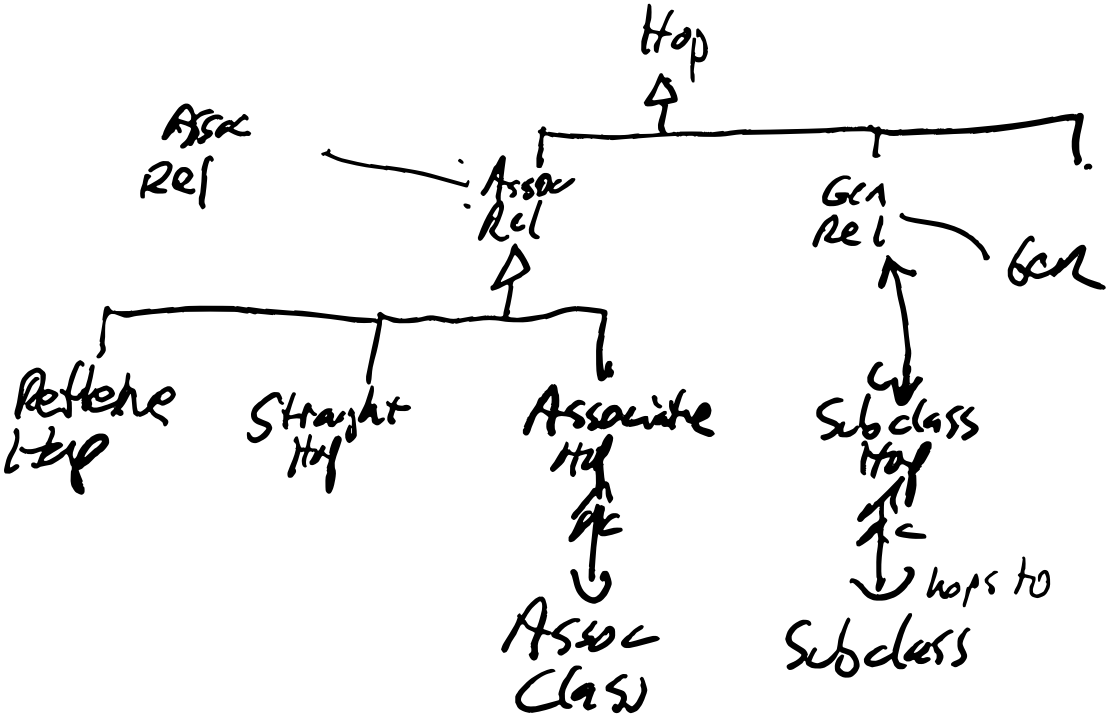
Reflexive hop

first satisfied,
last satisfied
all satisfied

-~
+~
*~

Note: Number of hops is unknown until run-time


Class
Path
Hop
Ord Rel
Assoc Rel
Assoc Rel Hop
Gen Rel
Gen Rel Hop
Ordinal Rel Hop
Reflexive Hop
Straight Hop
Associative Hop
Super Hop
Sub Hop
Subclass
Assoc Class
Cyclic Hop
- skipSpec
- direction
Rel Dir
11/13/21
Leon
Stan
(modelint)


Hop
Assoc Rel
Assoc Rel
Gen Rel
Gen
Reflexive Hop
Straight Hop
Associate Hop
Subclass Hop
hops to
Assoc Class
Subclass

a1

$\rightarrow f_{R1}(a1)$

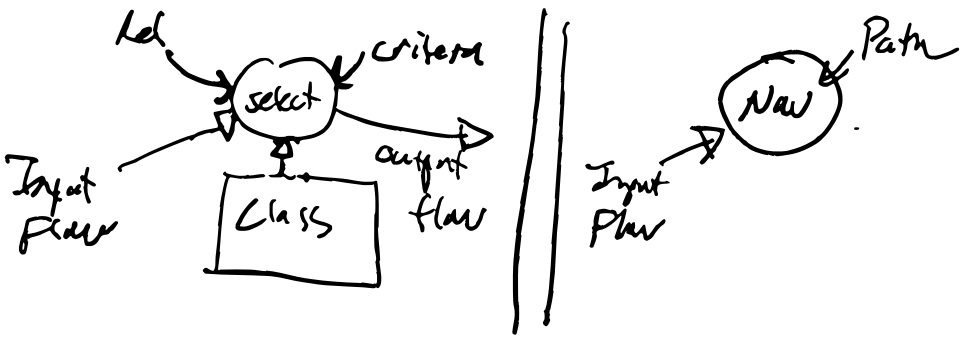
Each hop may apply Criteria

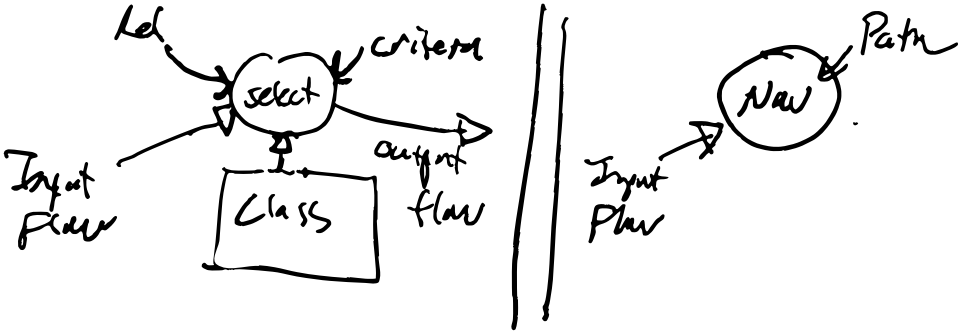
A select is performed after each hop

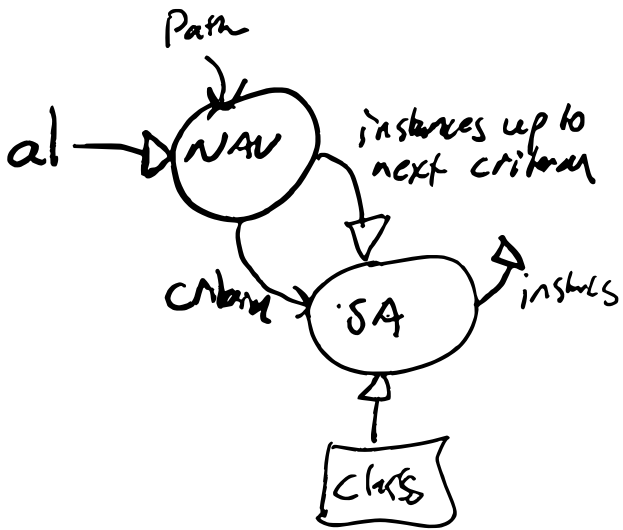
$C_{..} = a1/R1/R2 = f_{R2}(f_{R1}(a1))$

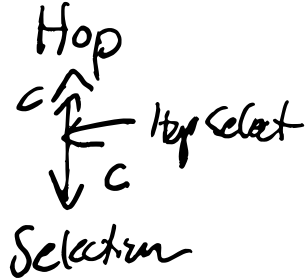
$C2_{..} = a1/R1(\dots)/C(\dots)$

↑ selection criteria









al ...:= Aircraft(...)

Simple case of a path — one class + criteria

Nav action begins with an InputFlow and a Path

Path begins w/ Hop

A(m)

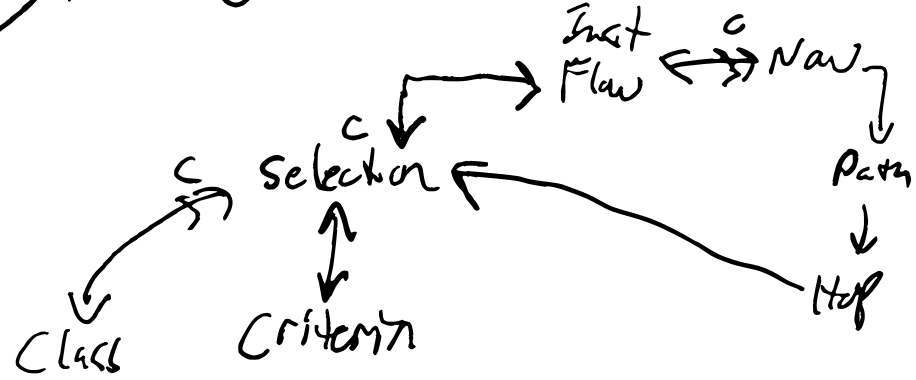
A(m) B

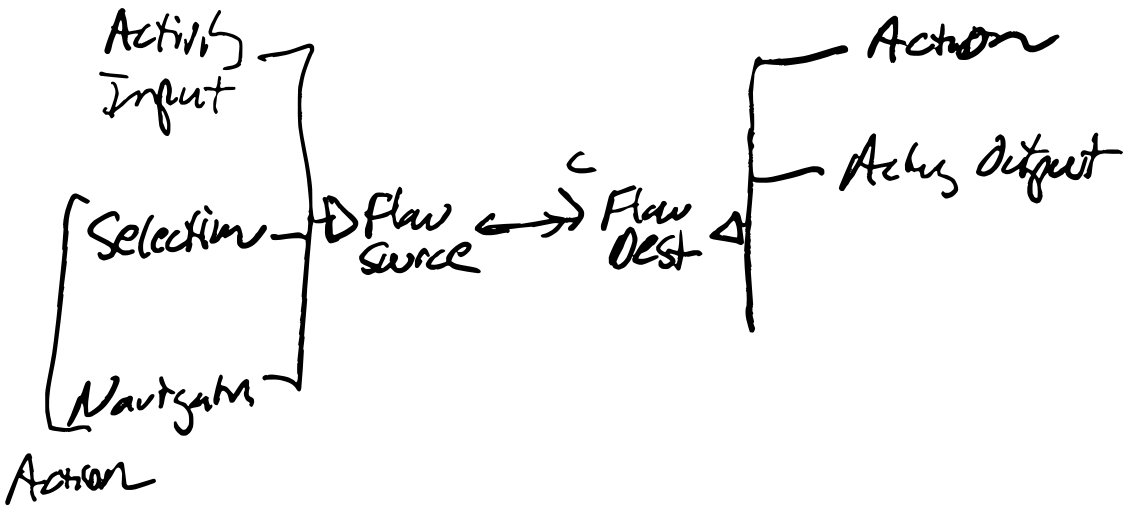
a1 / R1 / R2 / R3

a1 / R1(m) / R2 / R3

Paths

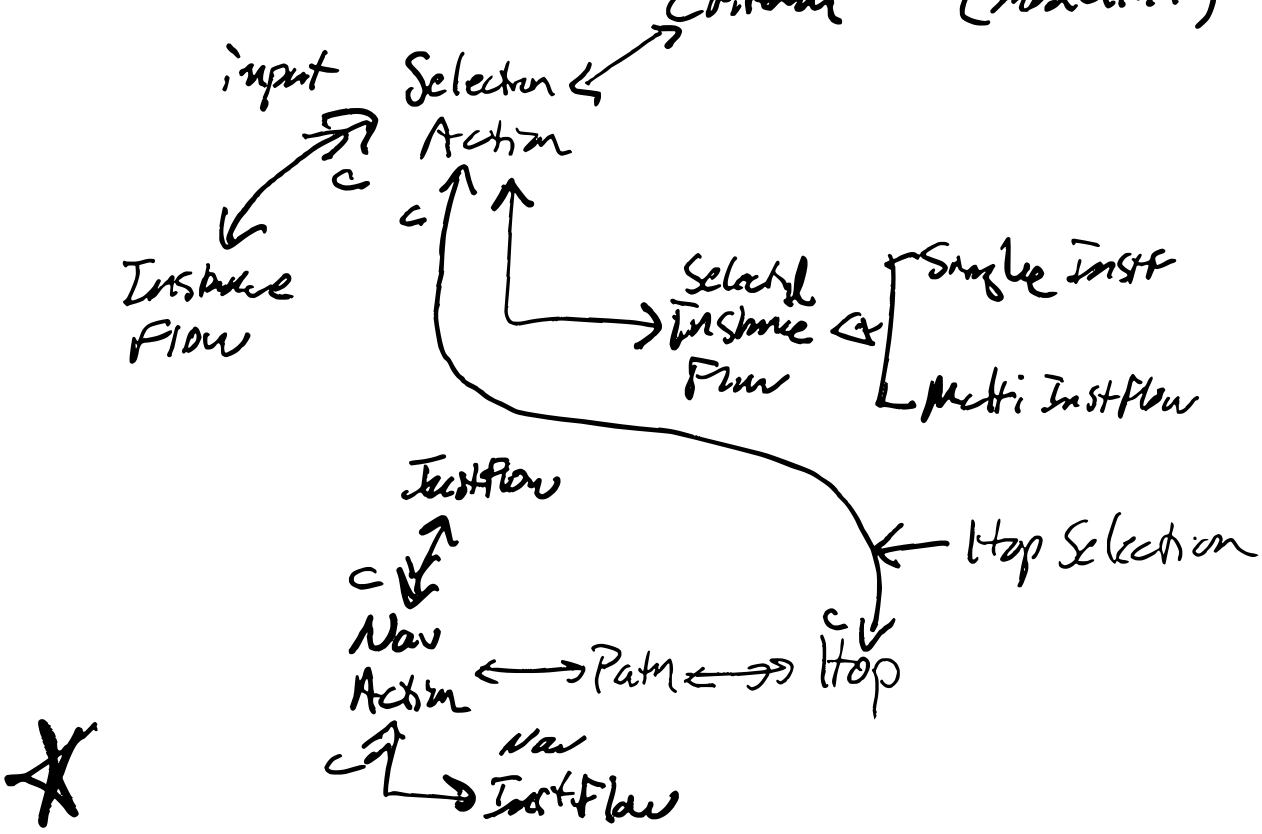
Hops

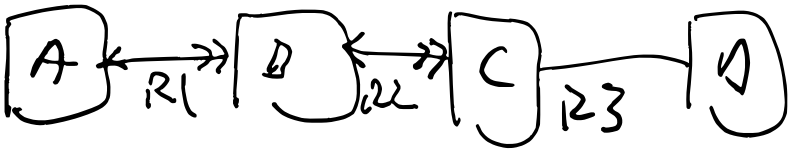



Activity Input
Selection
Navigation
Action
Flow Source
Flow Dest
Action
Activity Output

Selection & Navigation Actions

11/14/21
Lean Start
(modelint)



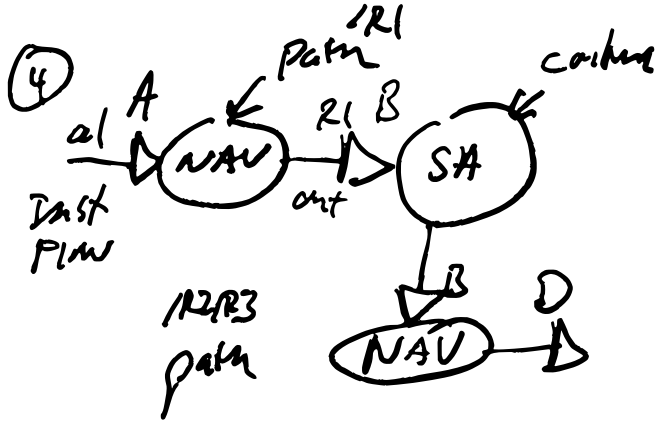


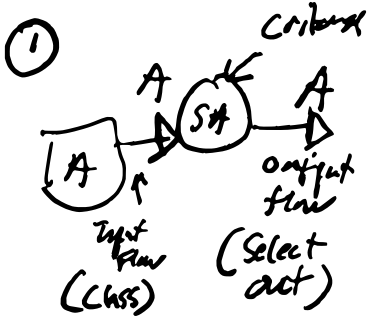
① A(m)

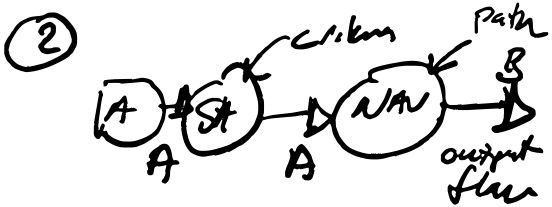
② A(m)/B

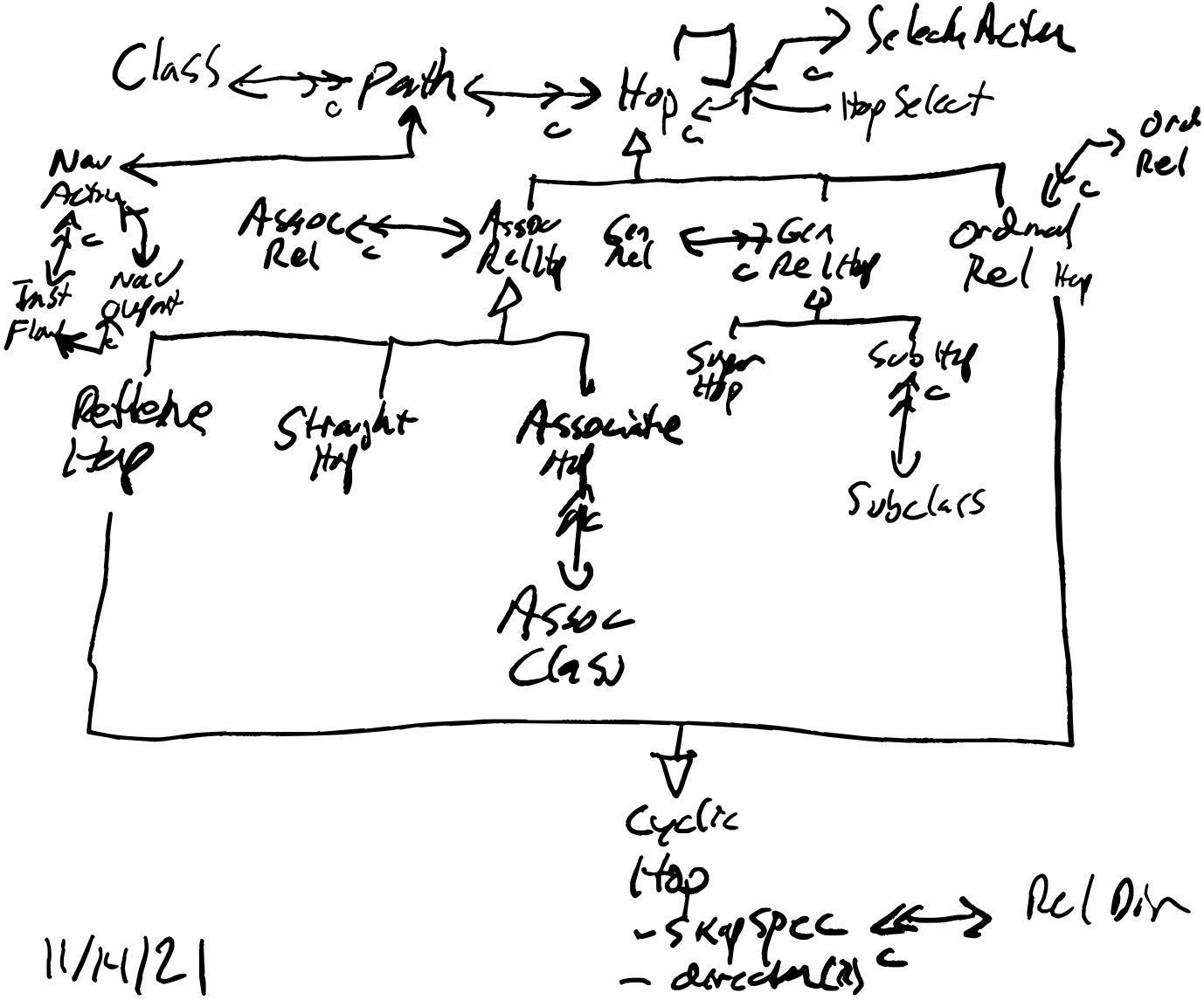
③ al/R1/R2/R3

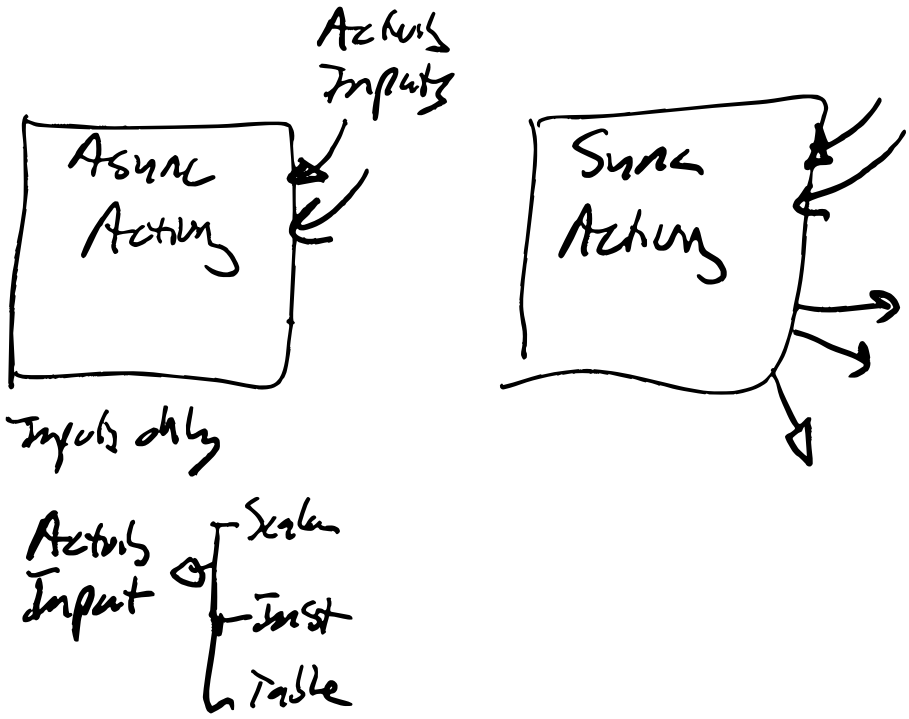
④ al/R1(m)/R2/R3



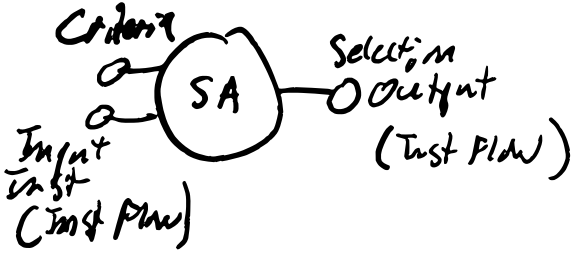


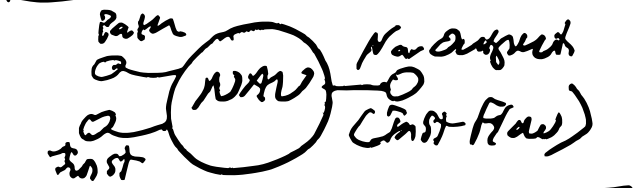


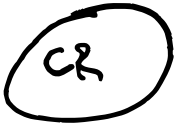

Class
Path
Hop
Select Action
Hop Select
Nav Action
Inst Flag
Nav Qualif
Assoc Rel
Assoc Rel Hop
Gen Rel
Gen Rel Hop
Ordnal Rel
Ordnal Rel Hop
Reflexive Hop
Straight Hop
Associative Hop
Assoc Class
Super Hop
Sub Hop
Subclass
Cyclic Hop
- skipSpec
- direction
Rel Dir
11/14/21

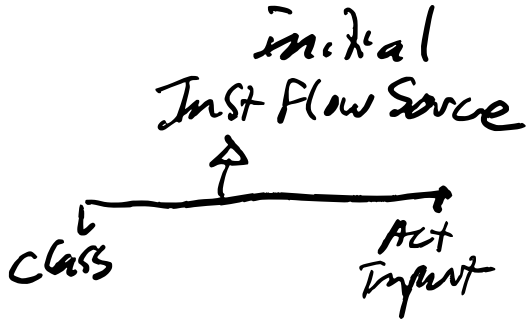

Async
Actions
Actions
Inputs
Inputs only
Sync
Actions
Actions
Input
Scalar
Inst
Table

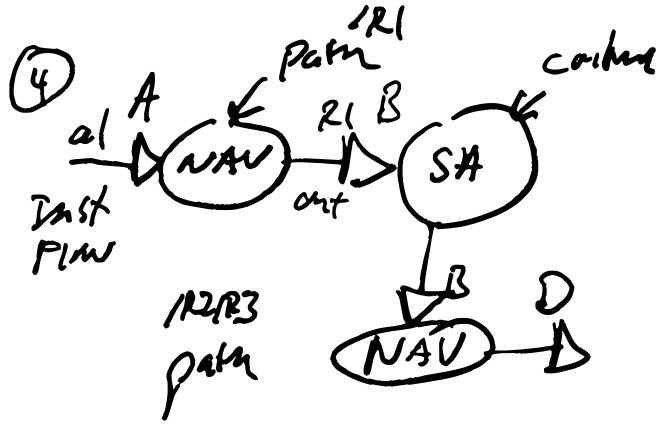
Actions

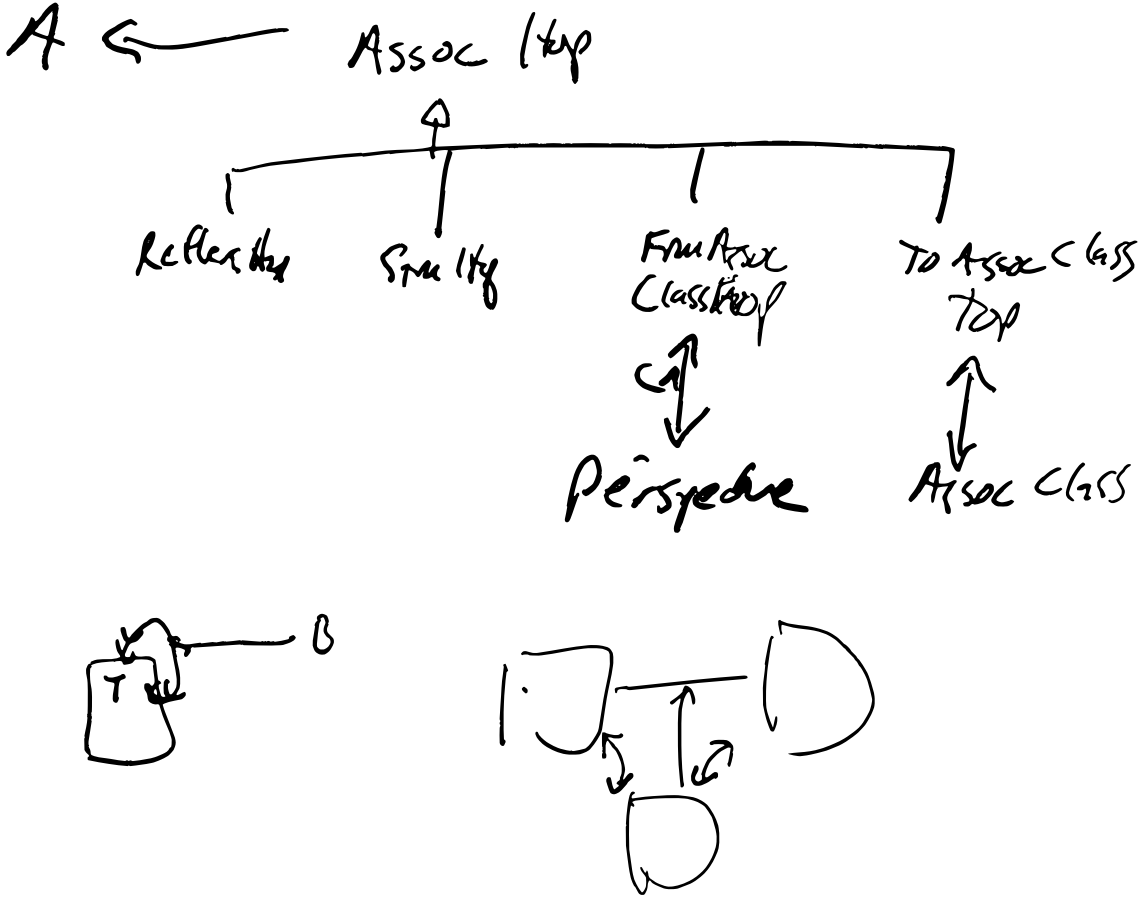

Criteria
SA
Selection
Output
(Inst Flow)
Input
Inst
(Inst Flow)


Path
NAV
Navigated
(Inst Flow)
Inst


CR


Initial
Inst Flow Source
Class
Act
Input


(4)
A
al
Inst
Flow
NAV
Path
1R1
R1
B
out
SA
criteria
B
D
1R2R3
path
NAV


A
Assoc Hop
Reflexivity
Symmetry
From Assoc Class Hop
To Assoc Class Hop
Perspective
Assoc Class
T
B

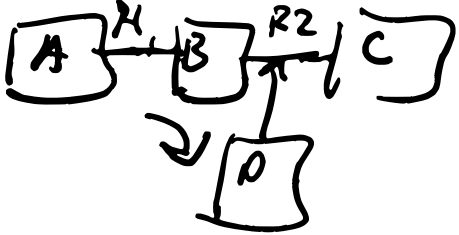
Each kind of Assoc Hop

① Straight Hop

cset.. = cset/R1/R2

No class accessed

② To Assoc Class Hop

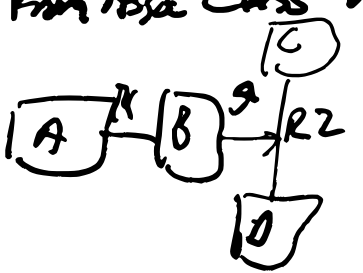


dset.. = cset/R1/R2/D

Assoc Class

To Assoc class hop

③ From Assoc Class Hop



cset.. = cset/R1/R2/C

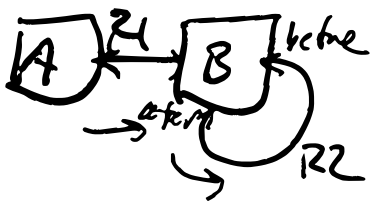
To Pside

From Assoc class hop

L Stern / Mint
2021-12-19

L Starr / Mint
2021-12-19

④ Circular Hop



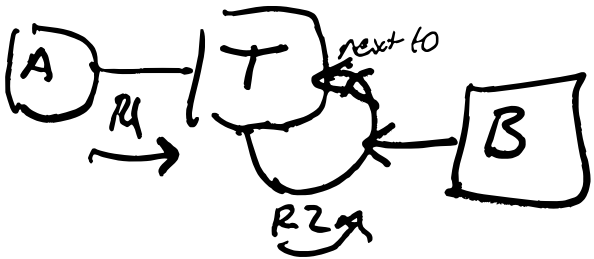
perspective

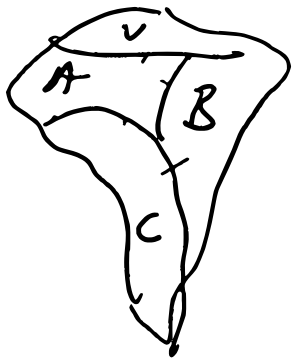
bset..aset/R1/R2/before

circular hop

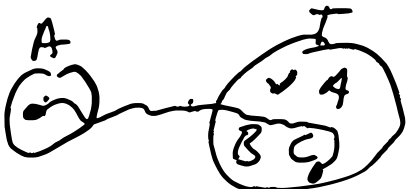
(both ordinal & referential reflexive associations)

⑤ Symmetric (only one perspective)



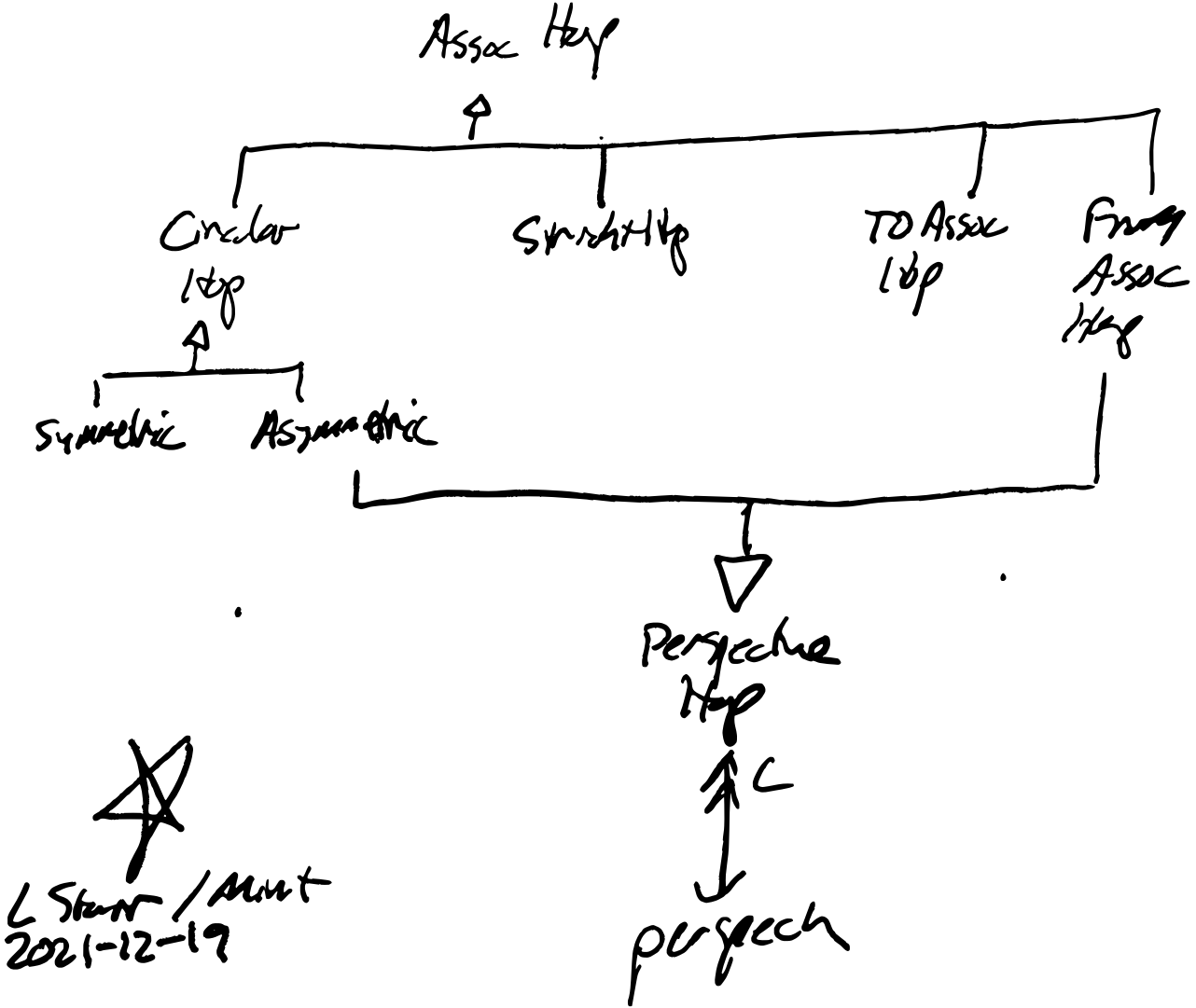


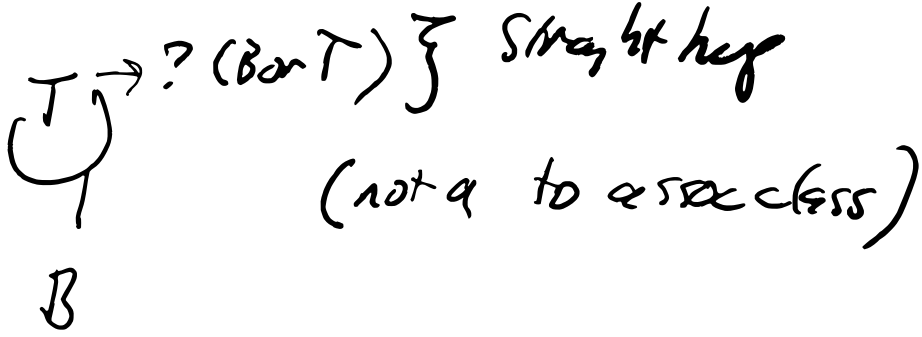
VA
AB
AC
BC

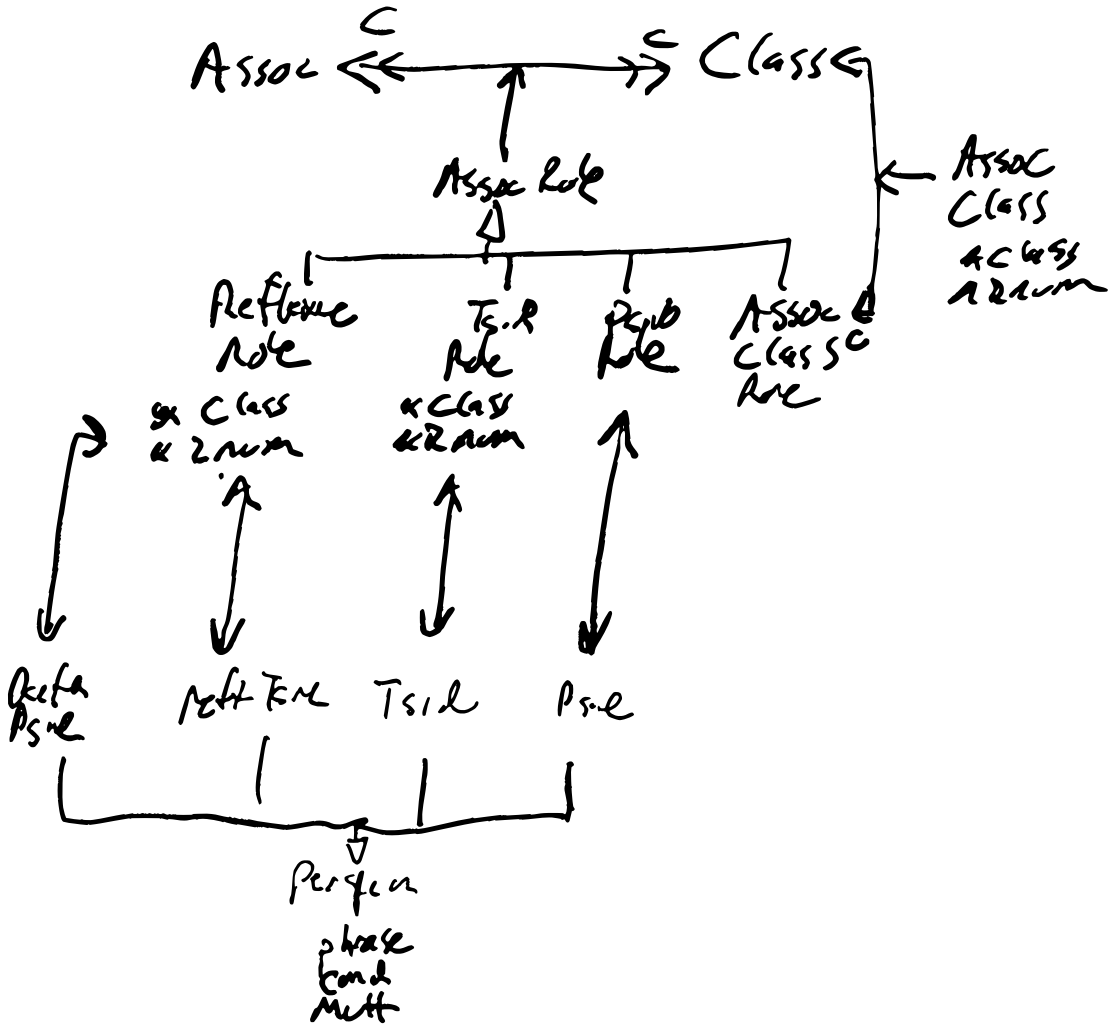


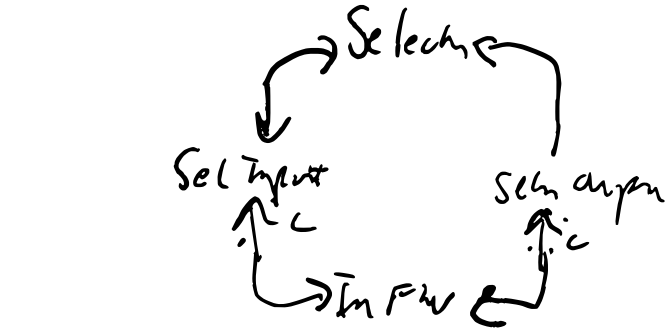
tset.. = aset/R1/R2

no need to specify like a symmetric


Assoc Hyp
Circular Hyp
Symmetric
Asymmetric
TO Assoc Hyp
Persecho Hyp
C
persech


T
→ ? (B or T)
B
Straight here
(not a to access)

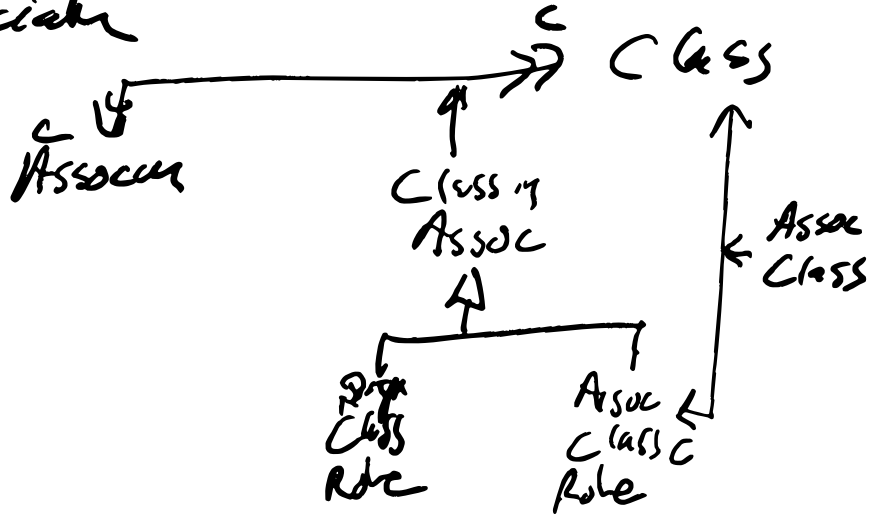

Assoc
Class
Assoc Role
Assoc Class
Reflexive Role
Tgt Role
Pub Role
Assoc Class Role
Reflexive Psvc
Tgt Psvc
Pub Psvc
Person

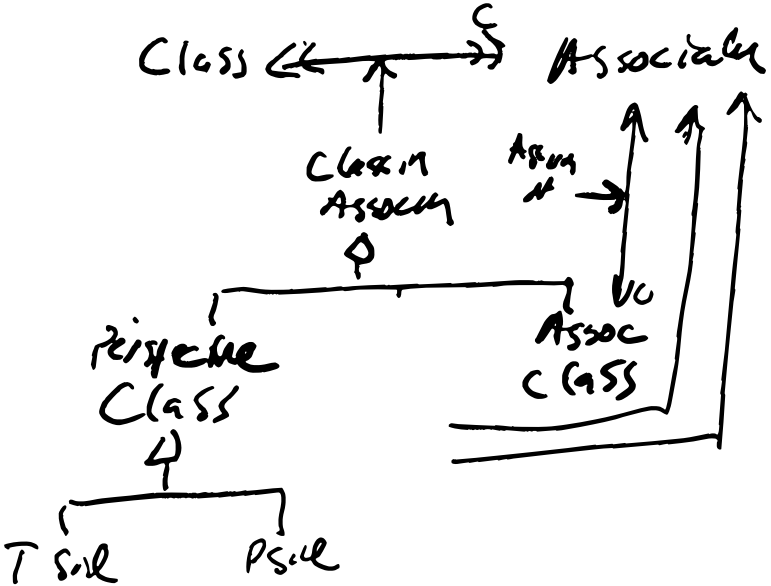

Selection
Sel Input
Sel output
Inst Flow


Selection
Inst
Inst Flow
Inst Flow
Source
Inst Flow

A class cannot be the assoc class in more than one association

A class cannot be both an assoc class & part class in the same association




Class
Associate
Class in Assoc
Assoc
1
Assoc Class
Persistent Class
Tsql
Psql