# UpSTAIRS with Sequence Diagrams

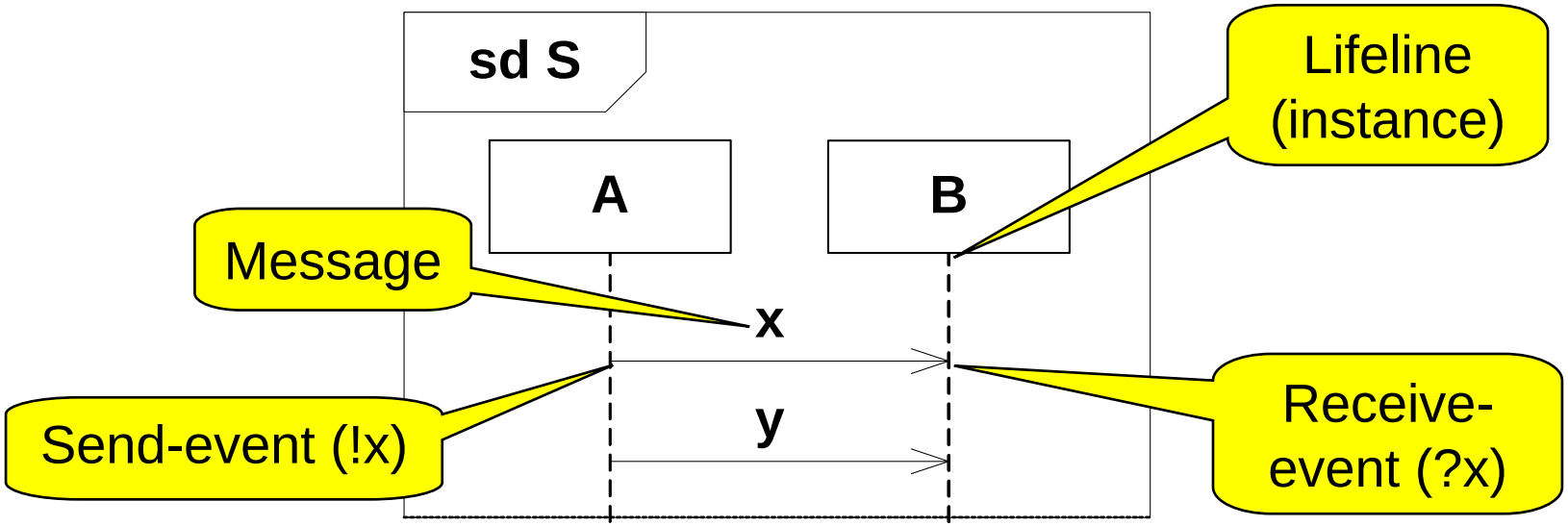
*Øystein Haugen,*
*Ragnhild Kobro Runde, Ketil Stølen*

University of Oslo and SINTEF ICT,
Norway

# Overview

- Interactions and trace semantics
- Interactions as example runs
- Underspecification and nondeterminism
- Refinement

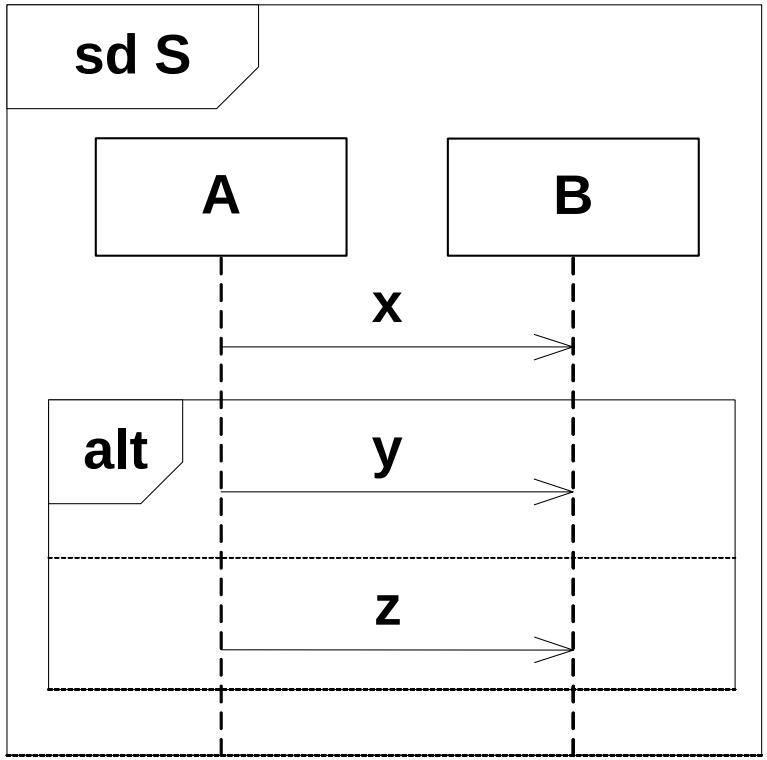
# Background: UML interactions




- Partial ordering of events:
  - The send event is ordered before the corresponding receive event.
  - Events on the same lifeline are ordered from the top and downwards.
- S specifies the two traces:
  - < !x, ?x, !y, ?y >
  - < !x, !y, ?x, ?y >


**Ragnhild Kobro Runde**

# Alternatives



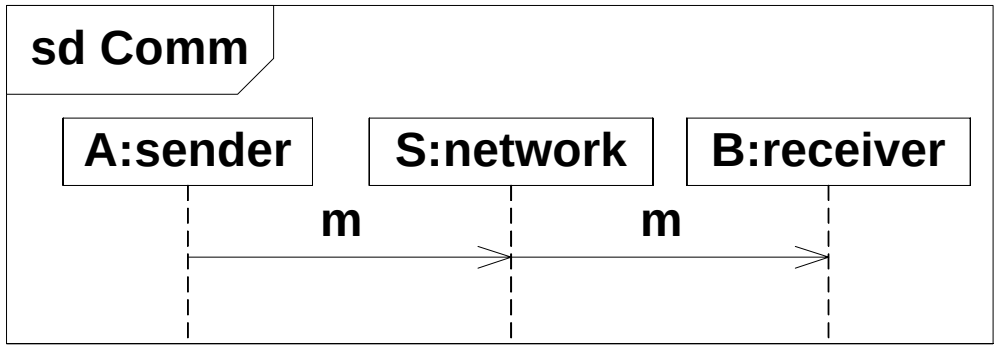

- S specifies the four traces:
  - < !x, ?x, !y, ?y >
  - < !x, !y, ?x, ?y >

  } First alternative

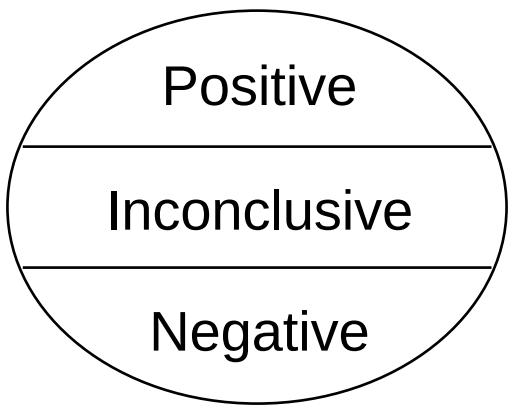
  - < !x, ?x, !z, ?z >
  - < !x, !z, ?x, ?z >

  } Second alternative

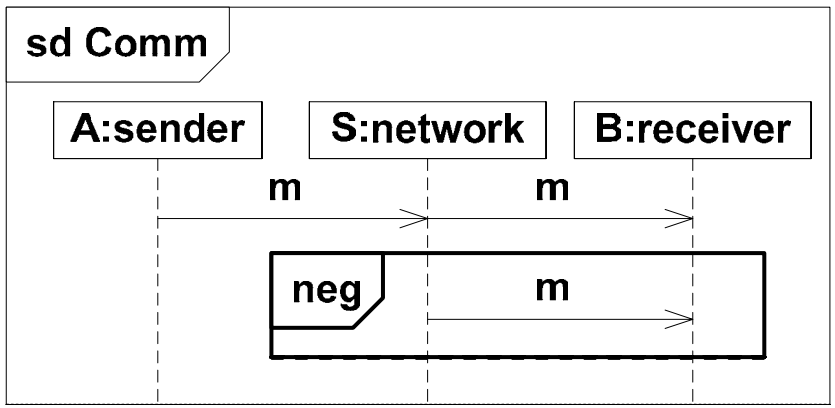
# Example: Network communication




- Interactions = example runs!
  - Specifies a set of positive and/or negative behaviours.



# Negative behaviour



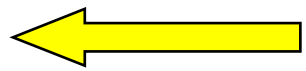

Positive:
$<!m_{AS},?m_{AS},!m_{SB},?m_{SB}>$

Negative:
$<!m_{AS},?m_{AS},!m_{SB},?m_{SB},!m_{SB},?m_{SB}>$
$<!m_{AS},?m_{AS},!m_{SB},!m_{SB},?m_{SB},?m_{SB}>$

- Formally:
  $(p1,n1) \gtrsim (p2,n2) =$
  $(p1 \gtrsim p2, (p1 \gtrsim n2) \cup (n1 \gtrsim p2) \cup (n1 \gtrsim n2))$
- Note:
  - Inconclusive + positive/negative = inconclusive
  - Positive + negative = negative


Ragnhild Kobro Runde

# Overview

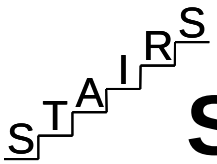
- Interactions and trace semantics
- Interactions as example runs
- Underspecification and nondeterminism 
- Refinement
- Data and guards

# Underspecification and non-determinism

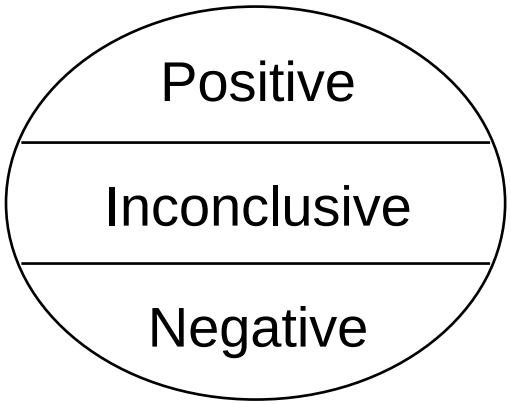
- Underspecification: Several alternative behaviours are considered equivalent (serve the same purpose).
- Inherent non-determinism: Alternative behaviours that must all be possible for the implementation.

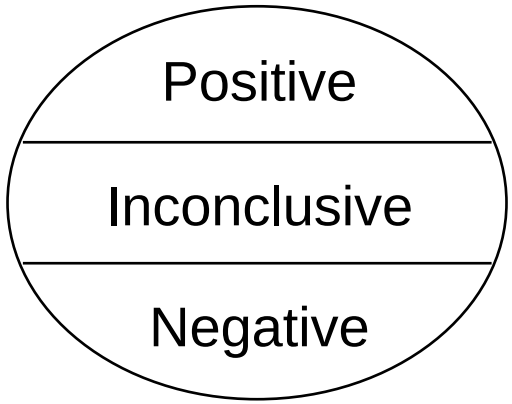
- These two should be described differently!

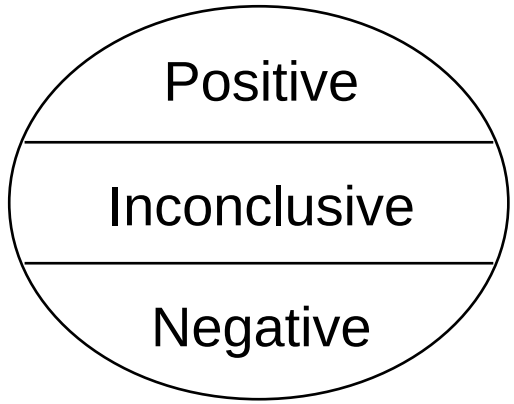

**Ragnhild Kobro Runde**



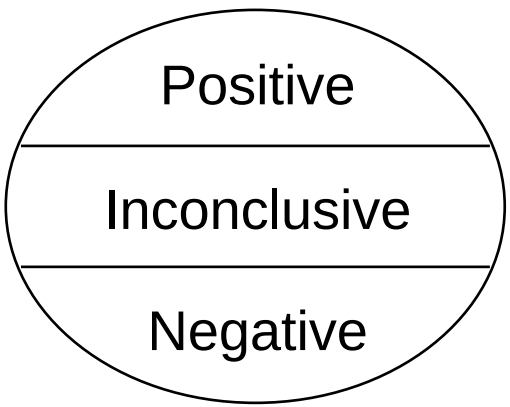

# STAIRS

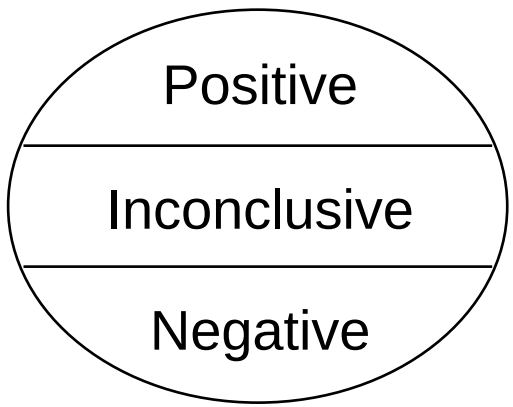


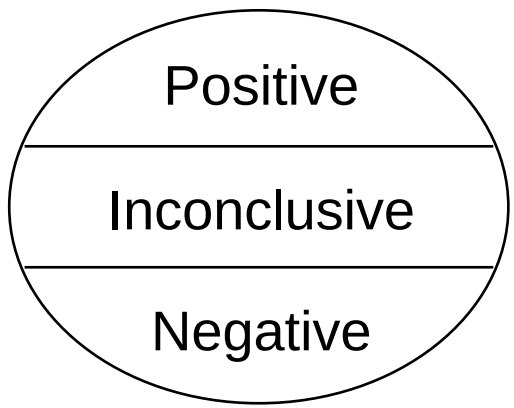




**xalt**






**Ragnhild Kobro Runde**

# alt vs xalt

- Assume

  [[ d1 ]] = (p1,n1)

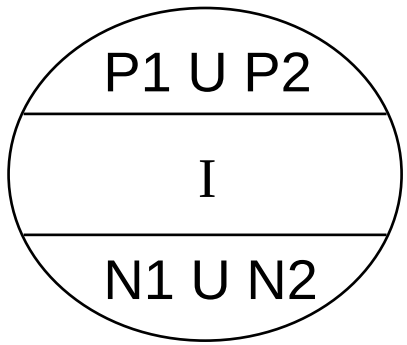
  [[ d2 ]] = (p2,n2)

- alt specifies potential behaviour:

  [[ d1 alt d2 ]]

  = [[ d1 ]] + [[ d2 ]]

  = (p1 U p2, n1 U n2)

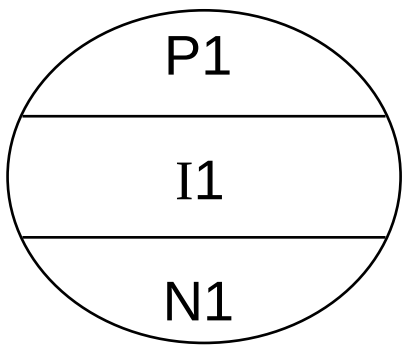


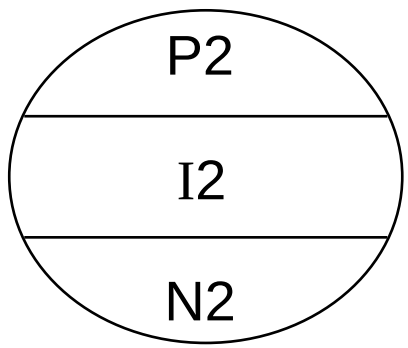

- xalt specifies mandatory behaviour:

  [[ d1 xalt d2 ]]

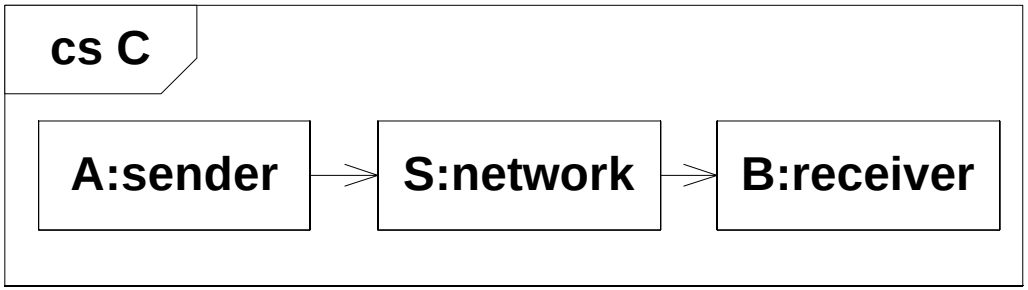
  = [[ d1 ]] U [[ d2 ]]

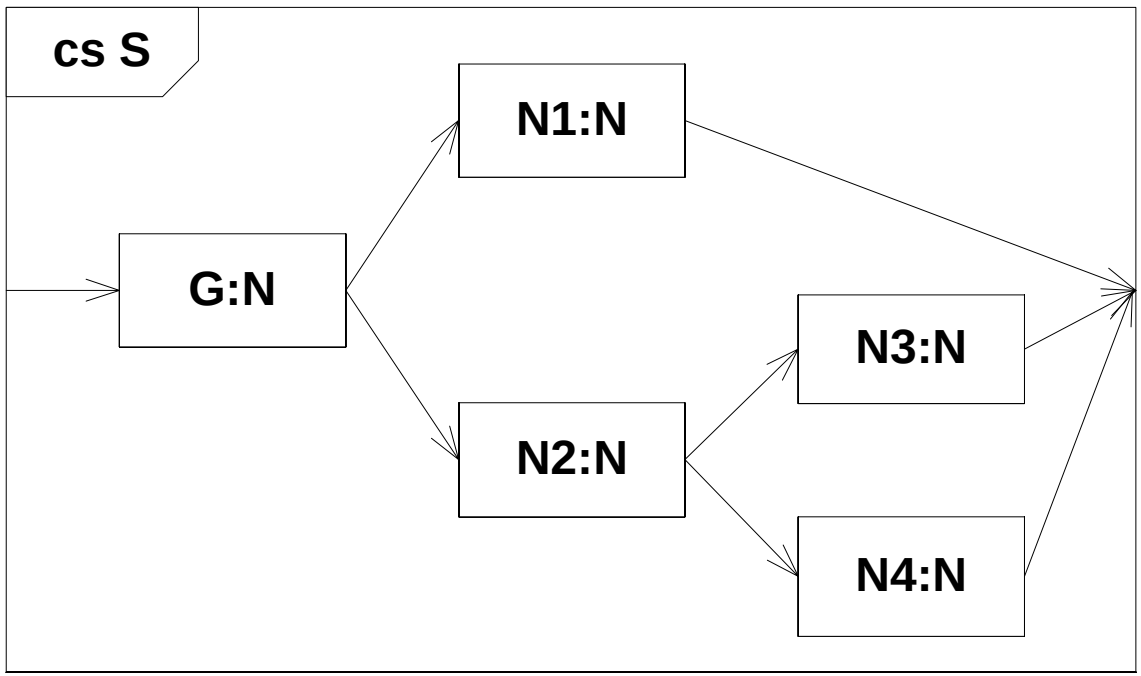
  = (p1,n1) U (p2,n2)



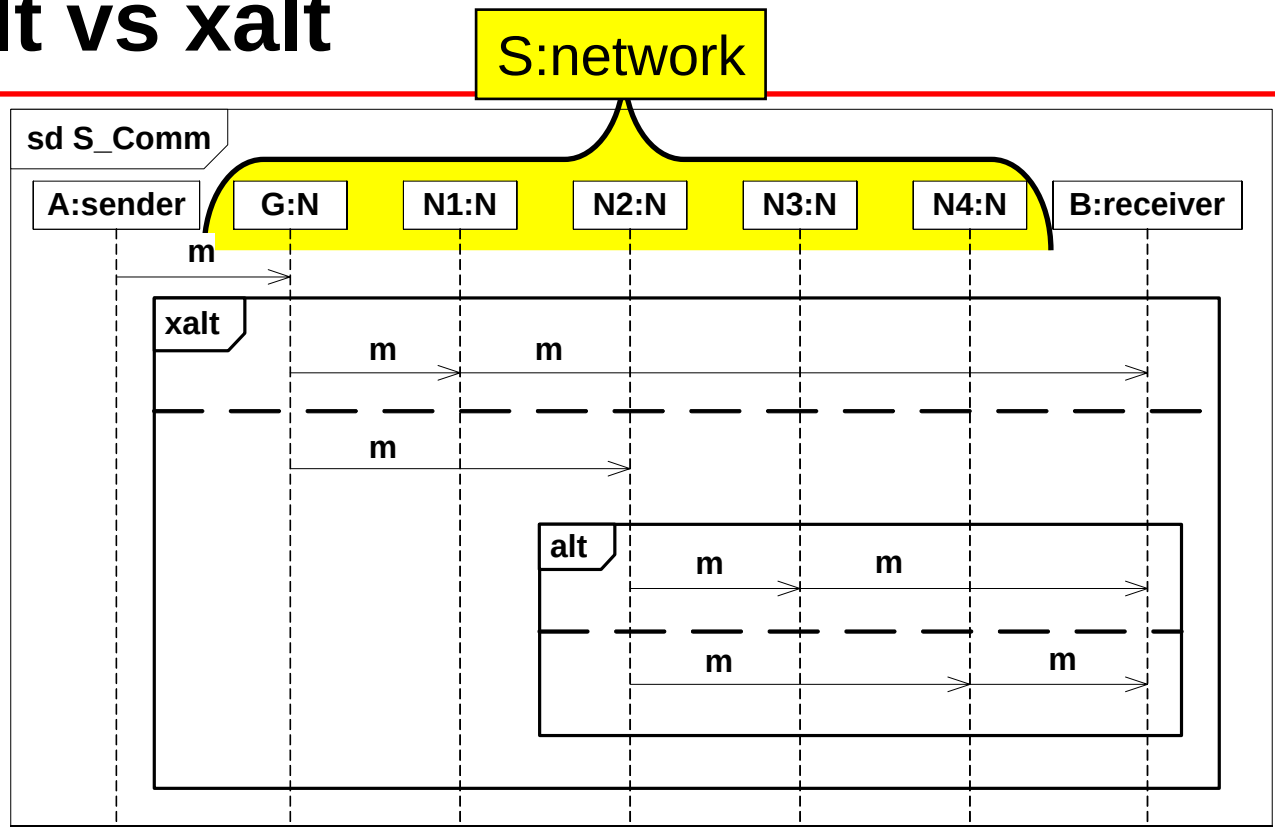



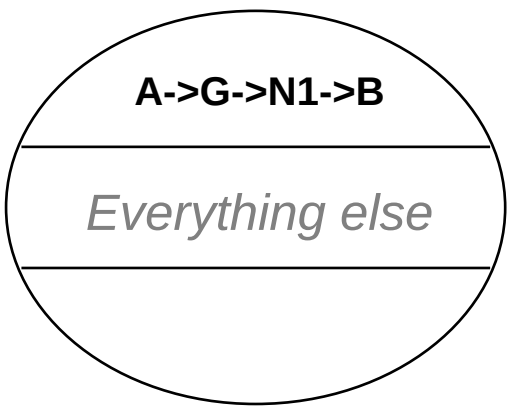
# Example: Network communication

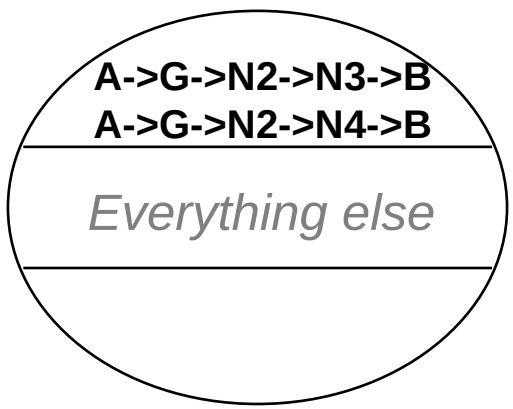


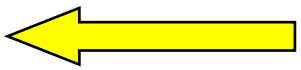
# alt vs xalt


S:network
sd S_Comm
A:sender
G:N
N1:N
N2:N
N3:N
N4:N
B:receiver
m
xalt
m
m
m
alt
m
m
m
m

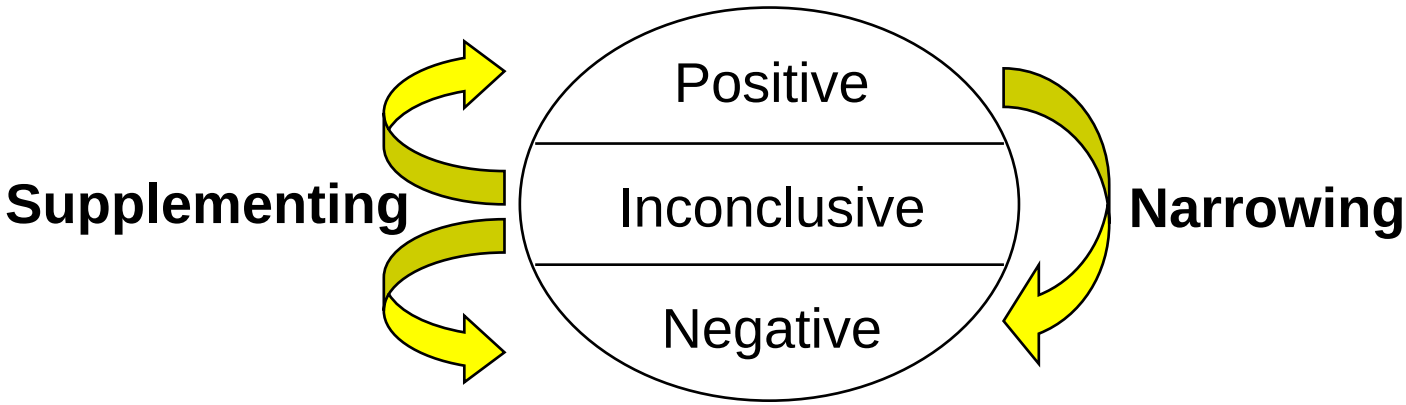

A->G->N1->B
Everything else


A->G->N2->N3->B
A->G->N2->N4->B
Everything else

# Overview

- Interactions and trace semantics
- Interactions as example runs
- Underspecification and nondeterminism
- Refinement 
- Data and guards

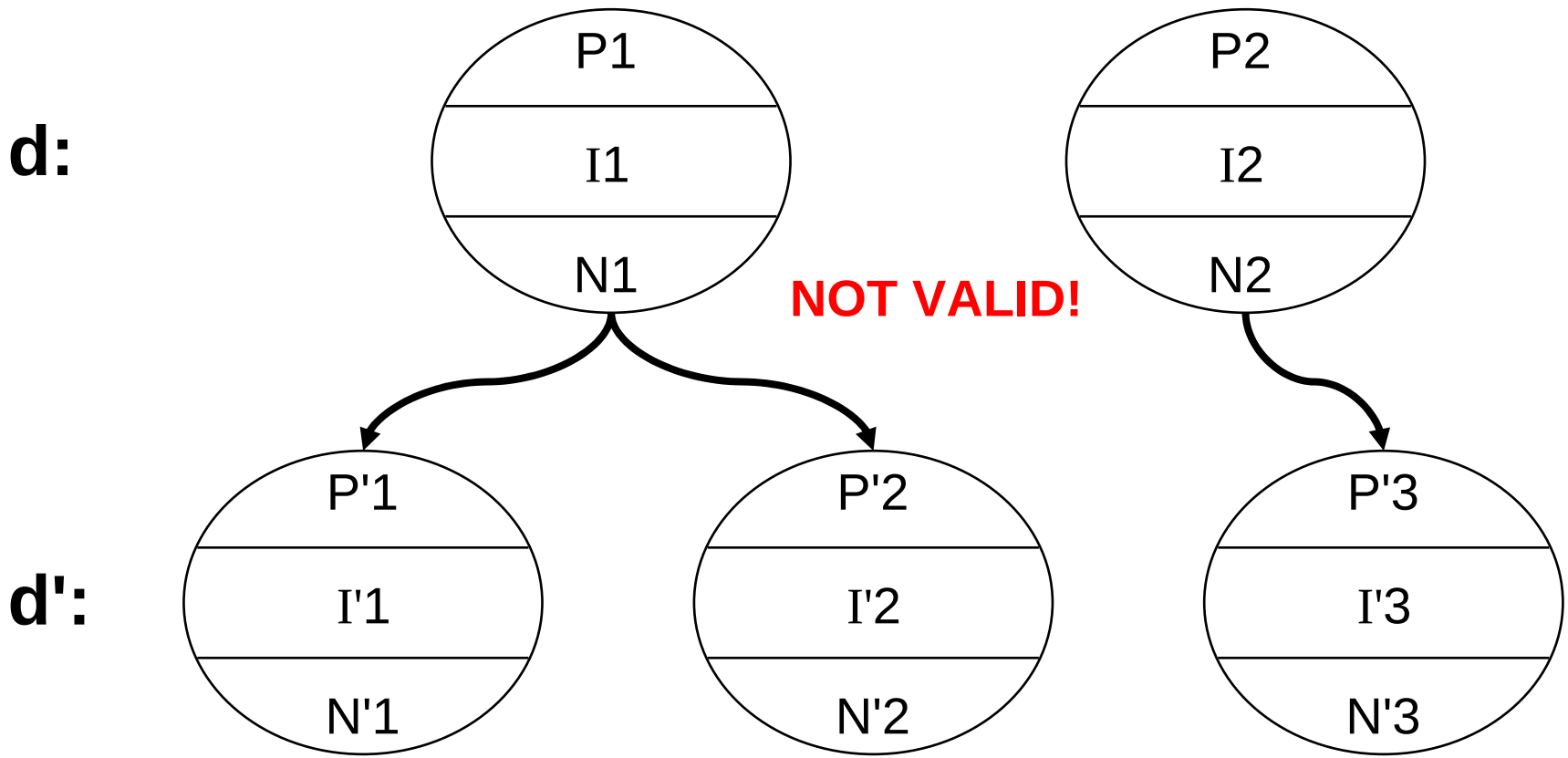
# Refinement in STAIRS



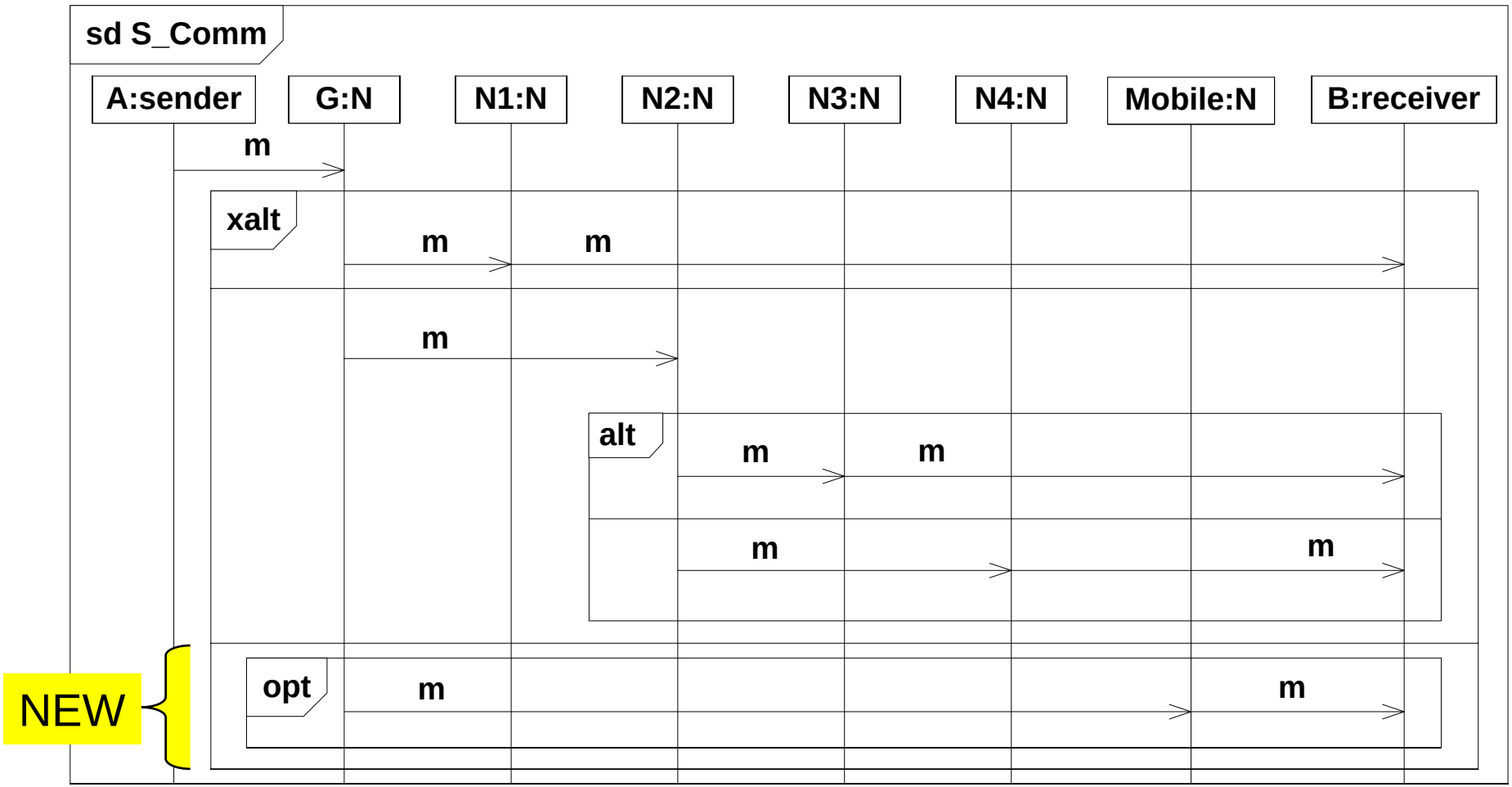

- An interaction obligation o'=(p',n') is a refinement of an interaction obligation o=(p,n) iff
  - n ⊆ n'
  - p ⊆ p'Un'

# Refinement contd.

- An interaction d' is a refinement of an interaction d iff
  $\forall o \in [[ d ]]: \exists o' \in [[ d' ]]: o \rightsquigarrow o'$



# Adding new obligations






Ragnhild Kobro Runde

# Supplementing