Refining models expressed in UML sequence diagrams

October 7, 2005

– STAIRS –
Steps to Analyze Sequence Diagrams with Refinement Semantics

Øystein Haugen **Ketil Stølen**

Motivation

- Make use of classical refinement theory in a practical UML setting
  - From theory to practice, and not the other way around
- We aim to explain how classical theory of refinement can be used to refine specifications expressed with the help
- Sequence diagrams can be used to explain other kinds of UML diagrams
- By defining refinement for sequence diagrams we implicitly define refinement for the UML as a whole

Traces

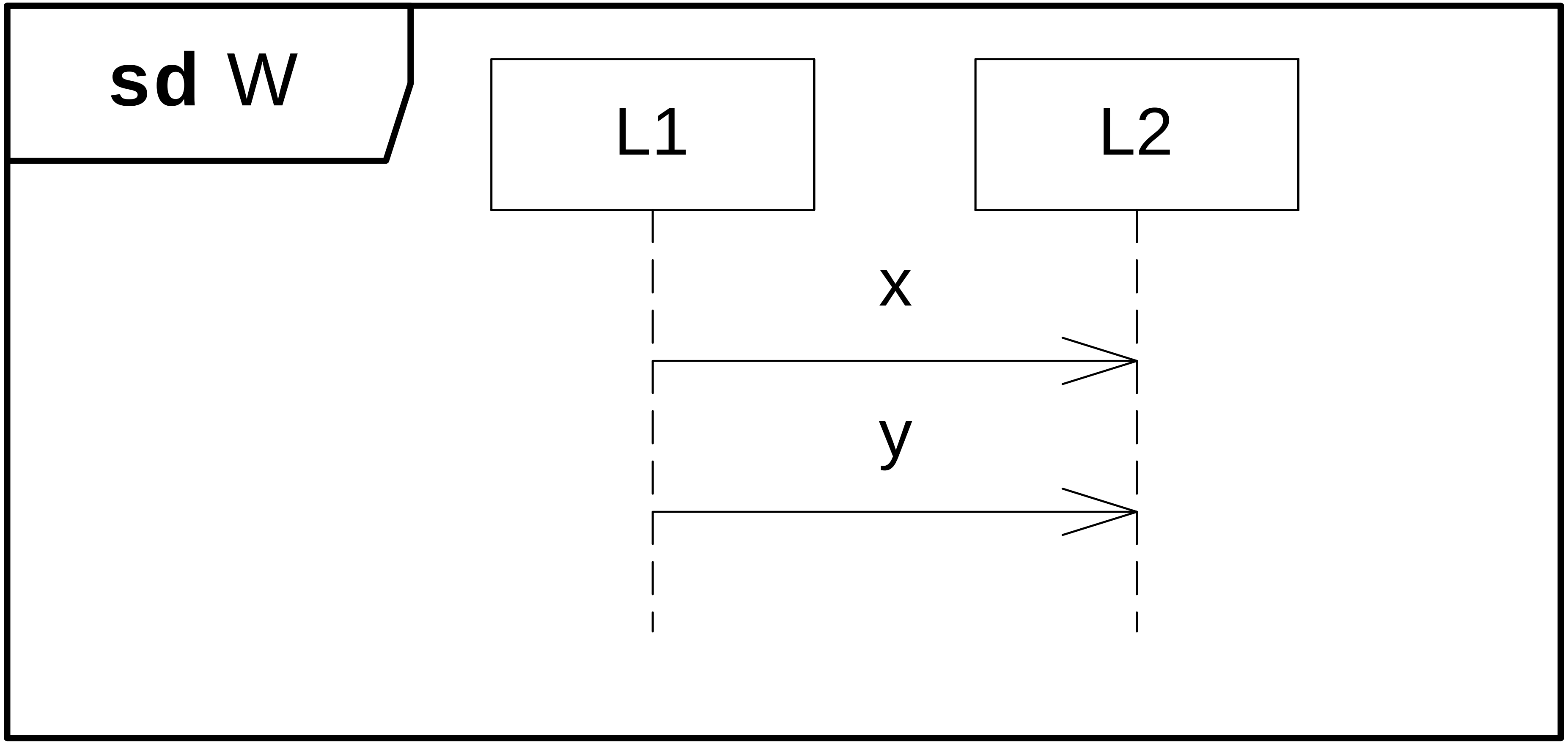
- Traces are used to represent system runs mathematically
- In the literature there are many different kinds of traces
- INF 5150 traces are sequences of events

$$\langle e1, e2, e3, e4, e4, e1, e2, e5, \ldots\ldots\ldots\ldots\ldots \rangle$$

- Events are instantaneous
- The number of events in a trace may be finite
  - may be caused by: termination, deadlock, infinite waiting, system crash
- The number of events in a trace may be infinite
  - May be cause by: non-termination, livelock, non-termination by purpose

Øystein Haugen Ketil Stølen

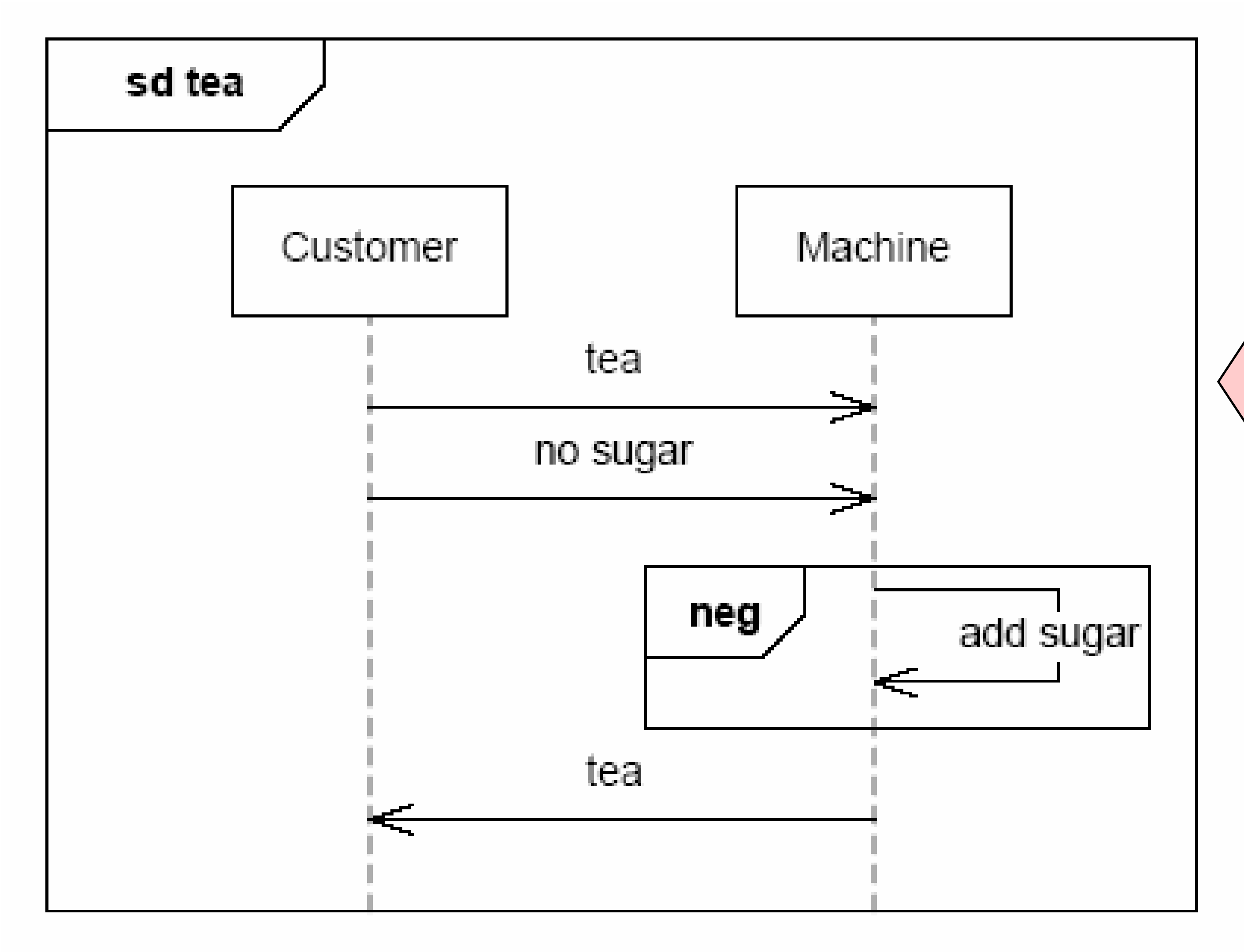
Weak sequencing



<!x,?x,!y,?y>

<!x,!y,?x,?y>

Positive and negative traces



What are the positive and negative traces of this diagram?

Semantics of sequence diagrams (without xalt)

- The formal semantics of a sequence diagram (without xalt) d is a pair (P,N) of sets of traces such that
  - P contains exactly the positive traces of d
  - N contains exactly the negative traces of d
- For any diagram d, we use [[d]] to denote its semantics
- The same trace may be both positive and negative with respect to the same diagram
  - But if this is the case, you have probably not specified what you intended to specify
- A trace that is neither positive nor negative for a sequence diagram d is inconclusive with respect to d

Pre-post specifications

Pre-post specifications are based on the assumption-guarantee paradigm

Integer division

var dividend, divisor, quotient, rest : Nat

pre divisor $\neq$ 0

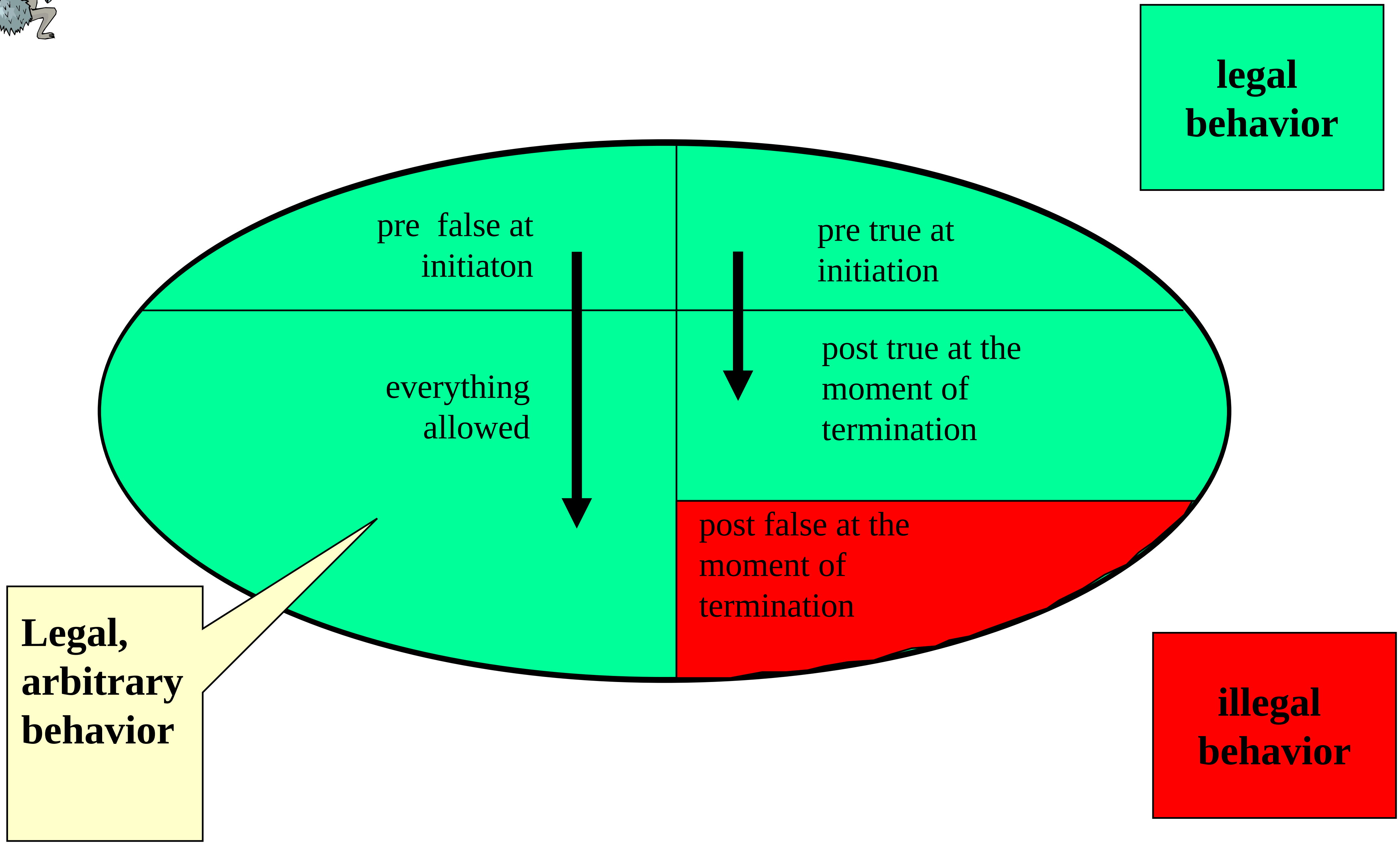
post (dividend = (quotient’ * divisor) + rest’) &

rest’ < divisor

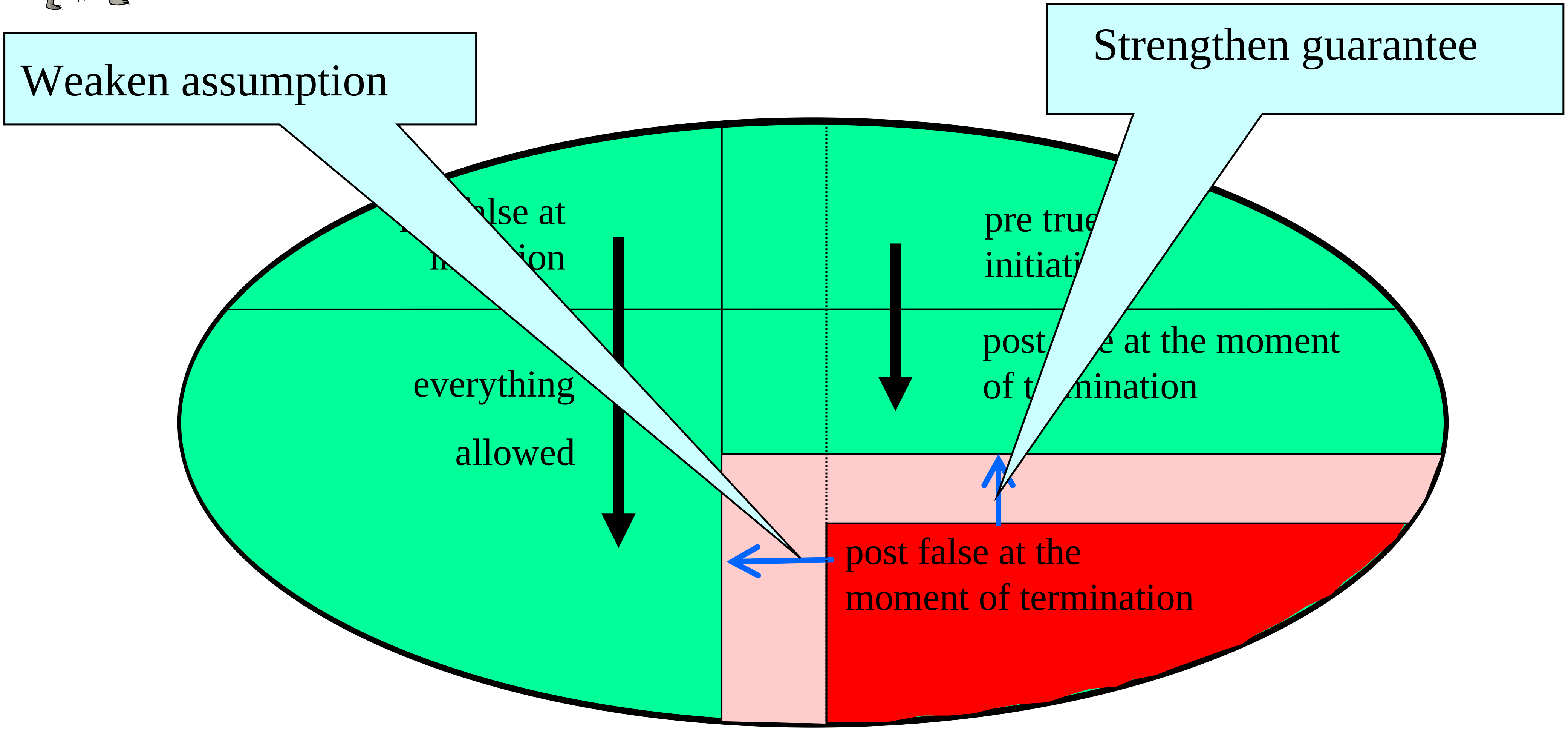
Assumption about the state at the moment the execution is initiated

Guarantee with respect to the state at the moment of termination

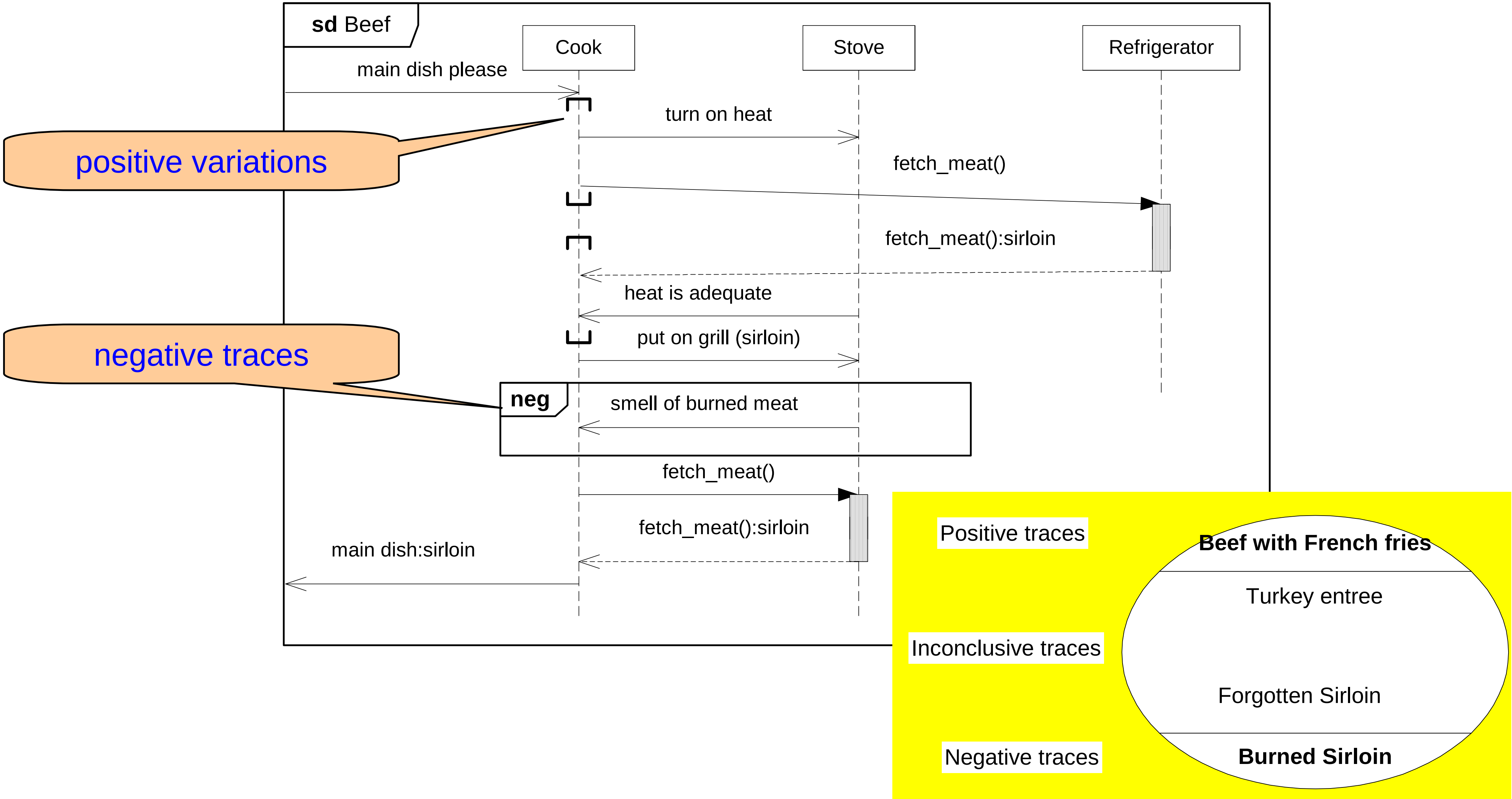
Semantics for pre-post specification



Property refinement for pre-post specifications



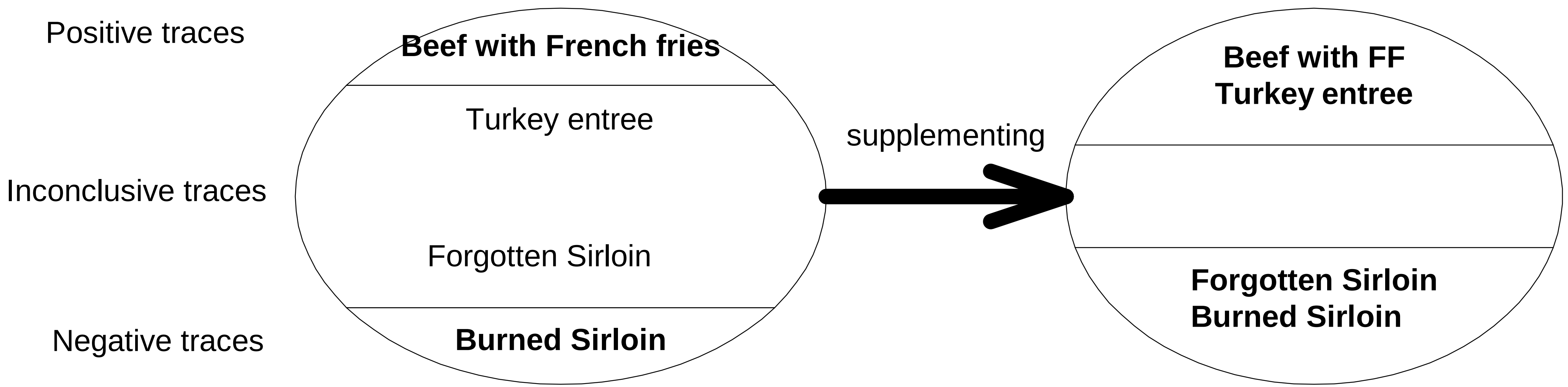
Property refinement of sequence diagrams



Øystein Haugen Ketil Stølen

Supplementing

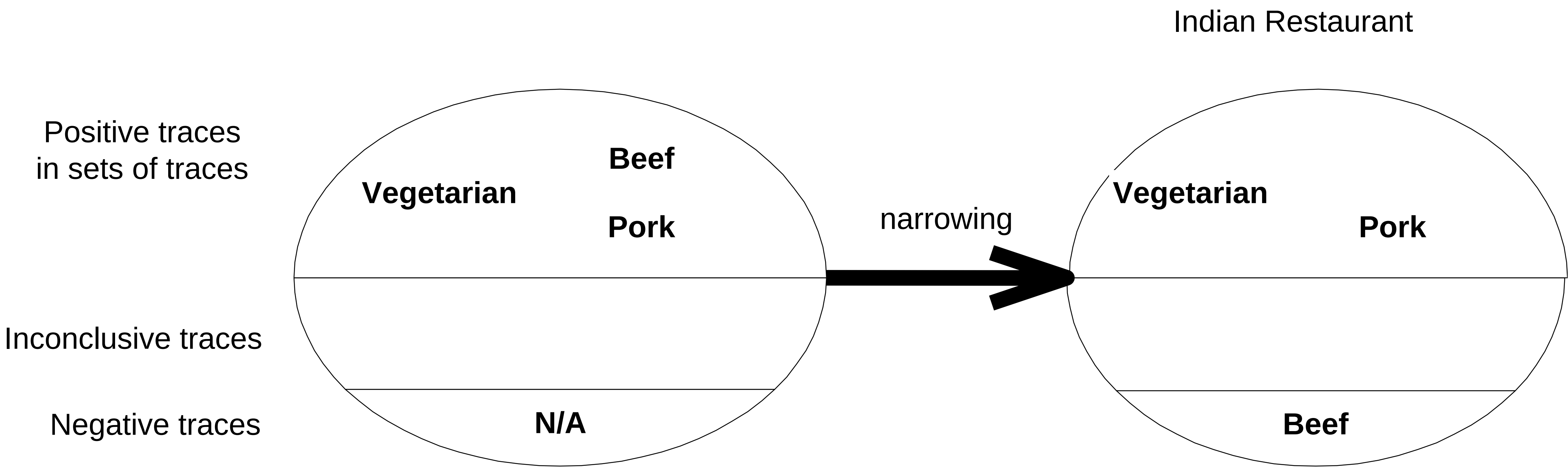
- To supplement means reducing the set of inconclusive traces by reclassifying inconclusive traces as either positive or negative
- The already positive traces remain positive
- The already negative traces remain negative



Øystein Haugen Ketil Stølen

Narrowing

- To narrow means reducing the set of positive traces by reclassifying positive traces as negative
- The inconclusive traces remain inconclusive
- The already negative traces remain negative



Øystein Haugen Ketil Stølen

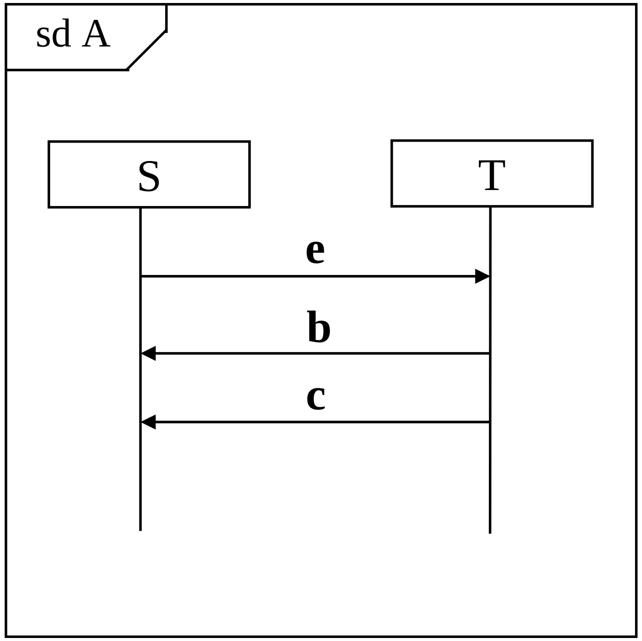
INDIRECT DEFINITION: Property refinement in STAIRS

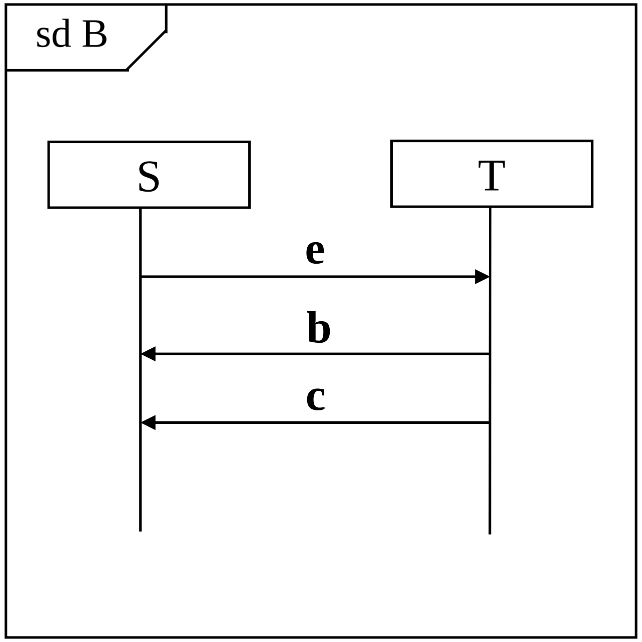
- **A sequence diagram B is a property refinement of a sequence diagram A if**
  - **A and B are semantically identical**
  - **B can be obtained from A by supplementing**
  - **B can be obtained from A by narrowing**
  - **B can be obtained from A by a finite number of steps**

    A -> C1 -> C2 -> …. ->Cn->B

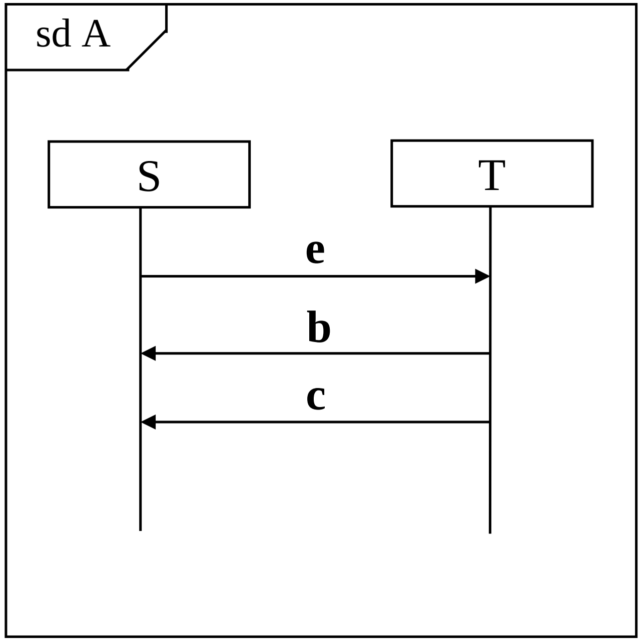
    each of which is either a supplementing or a narrowing

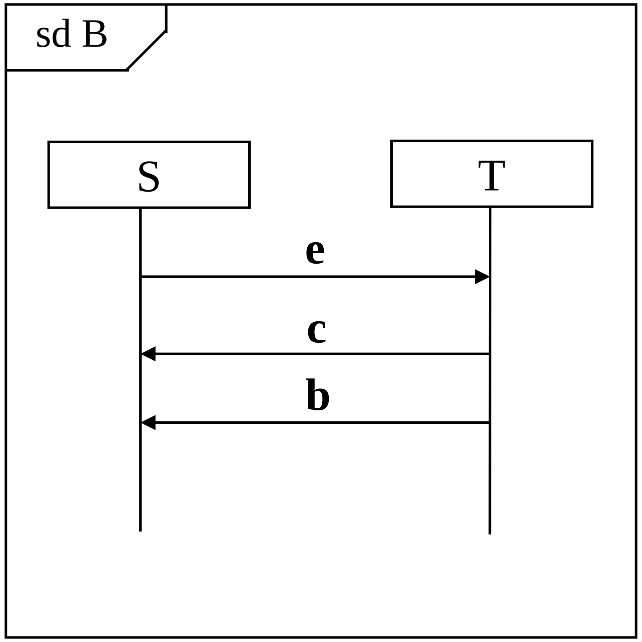
Is B a property refinement of A?



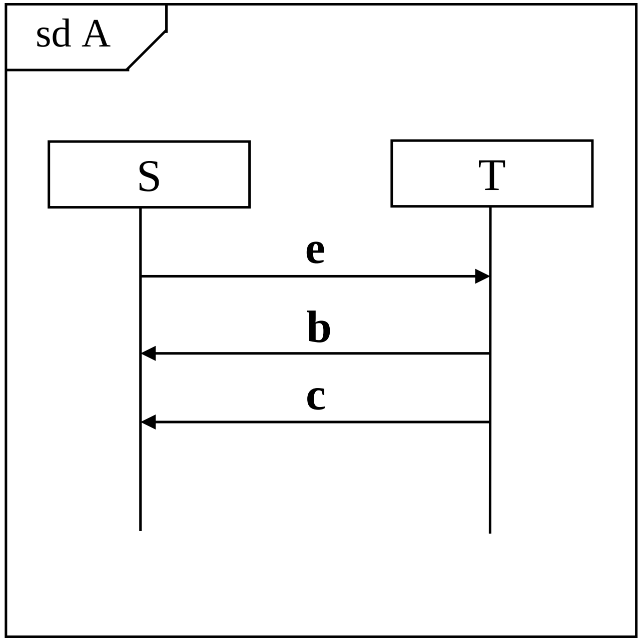


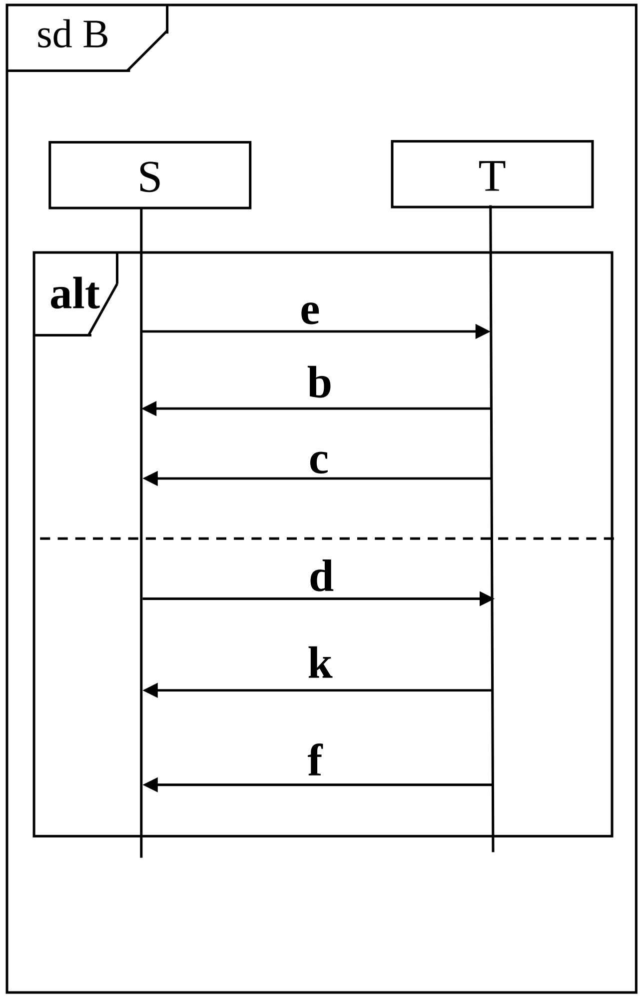
Is B a property refinement of A?





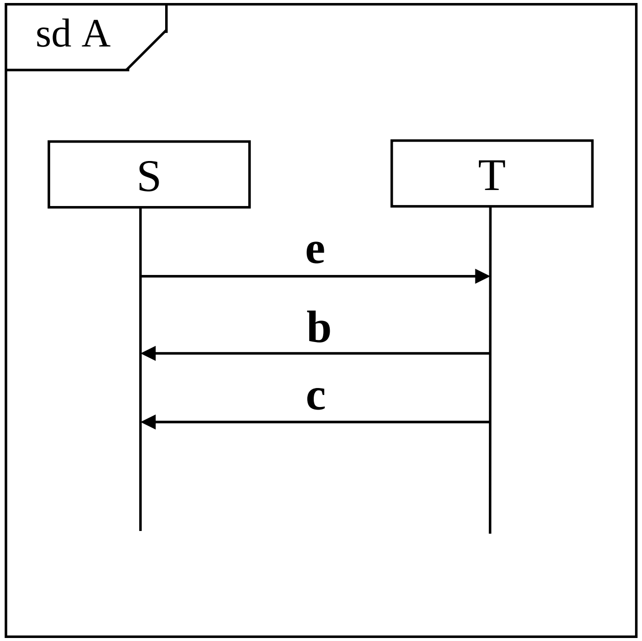
Is B a property refinement of A?

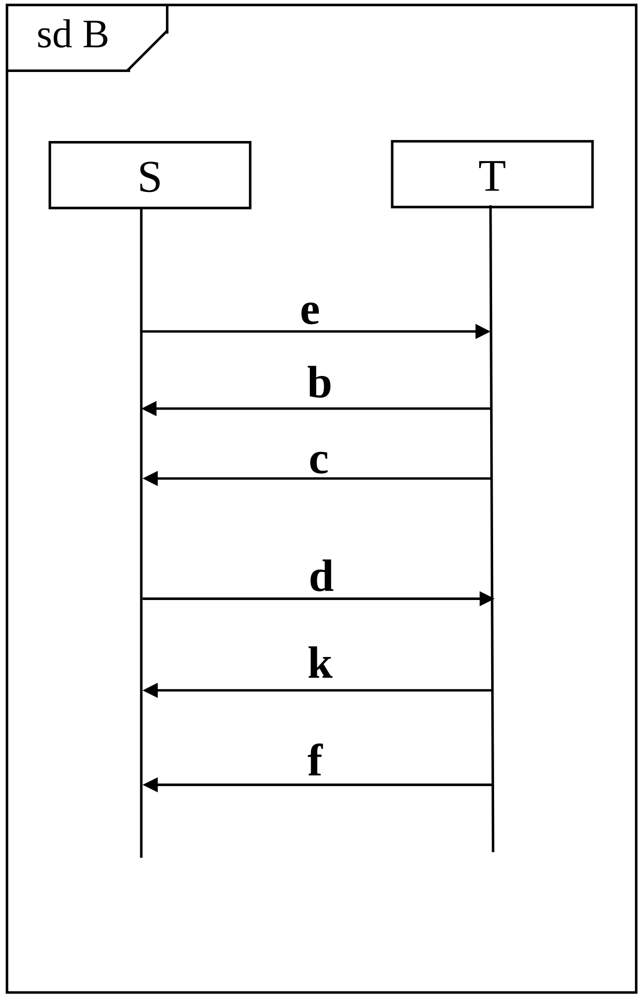




Øystein Haugen Ketil Stølen

Is B a property refinement A?

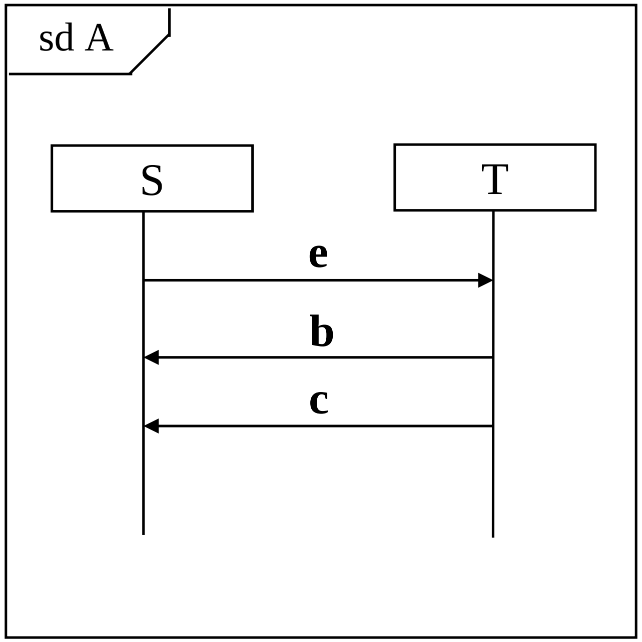


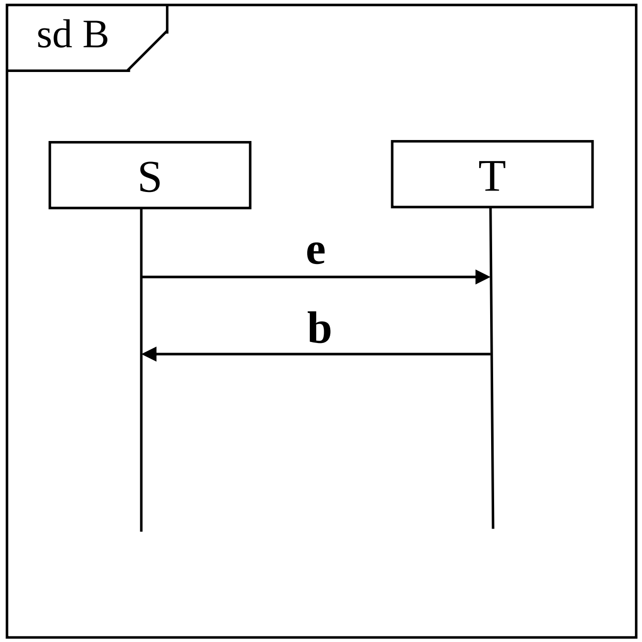


Øystein Haugen

Ketil Stølen

Is B a property refinement of A?

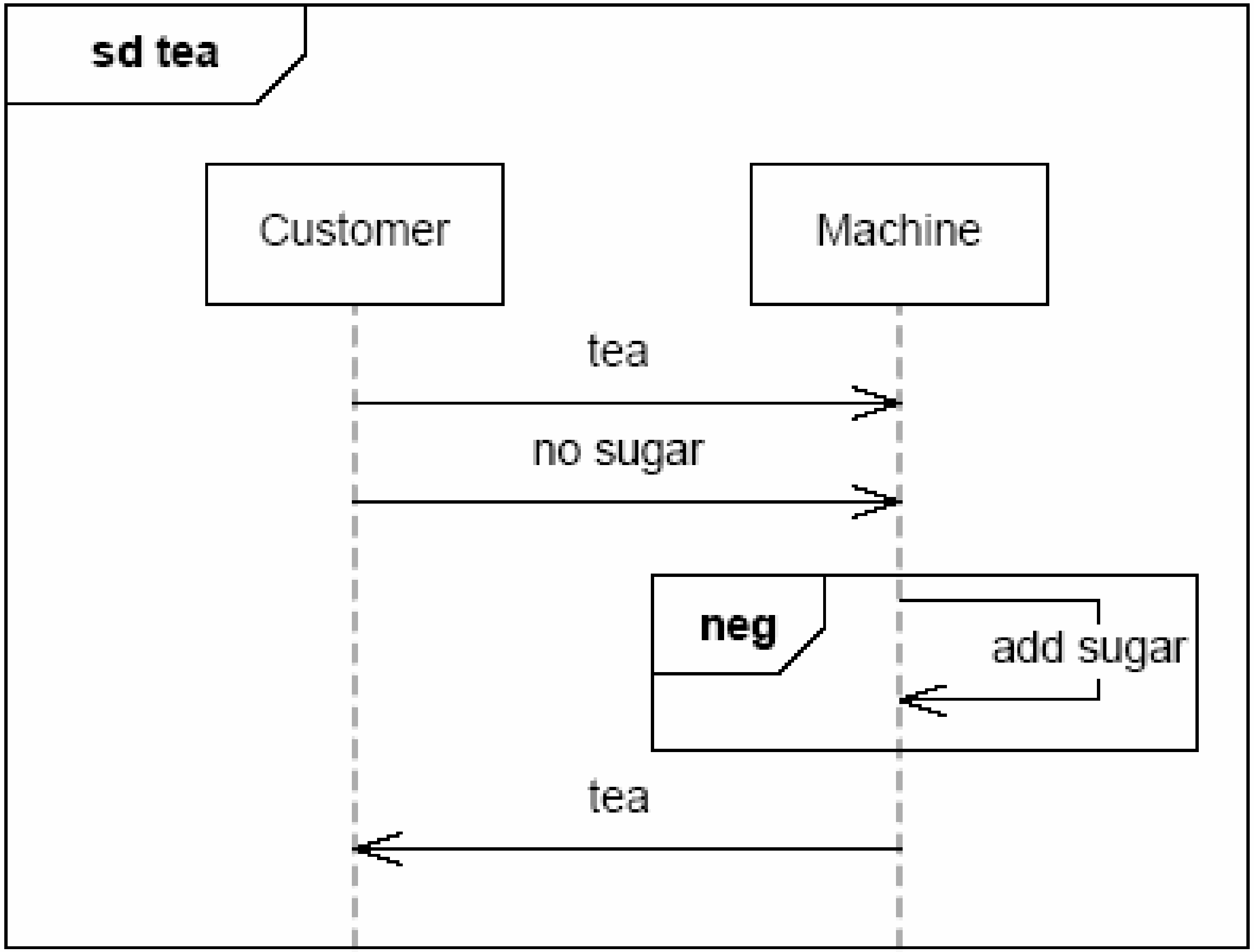




DIRECT DEFINITION: Property refinement in STAIRS

- A sequence diagram B is a property refinement of a sequence diagram A if
  - every trace classified as negative by A is also classified as negative by B
  - every trace classified as positive by A is classified as either positive or negative by B

Property refinement of “tea”



What does it mean to
- supplement?
- narrow?
- both in one go?

Distinguishing two different types of non-determinism

- **Classical distinction between mandatory and potential non-determinism**

- **Most specification languages cannot distinguish mandatory and potential non-determinism**

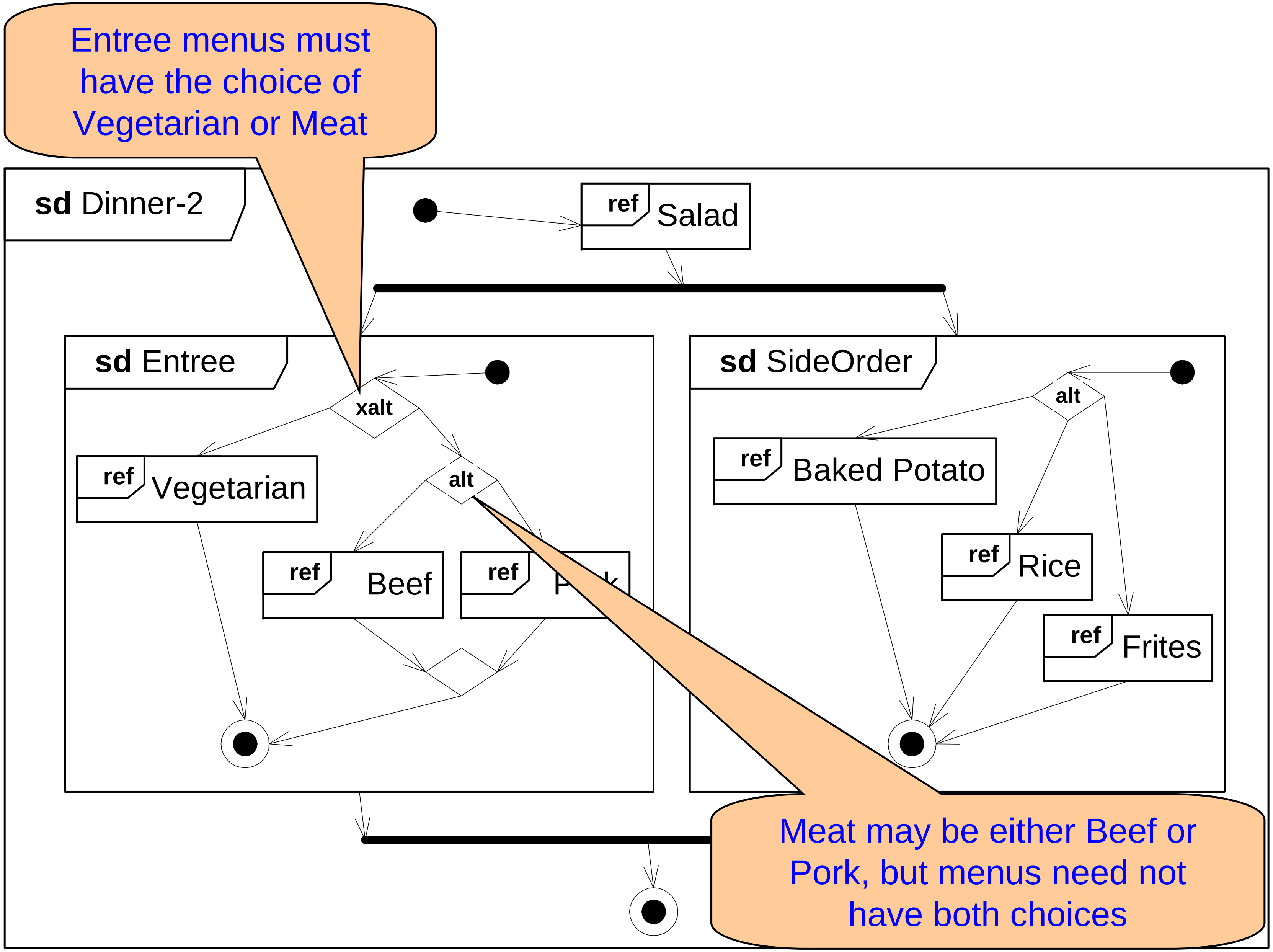
Øystein Haugen **Ketil Stølen**

The need for both *alt* and *xalt*

- **Potential non-determinism captured by alt allows abstraction and inessential non-determinism**
  - **Under-specification**
  - **Non-critical design decisions may be postponed**

- **Mandatory non-determinism captured by xalt characterizes non-determinism that must be reflected in every correct implementation**
  - **Makes it possible to specify games**
  - **Important in relation to security**
  - **Also helpful as a means of abstraction**

Restaurant example with both alt and xalt



Øystein Haugen Ketil Stølen

General STAIRS-semantics for sequence diagrams

- The semantics of a sequence diagram is a set of pairs of sets of traces
  - {(P1,N1), (P2,N2),..., (Pn,Nn)}
  - where for each index j
    - Nj is a set of negative traces
    - Pj is a set of positive traces
- We refer to these pairs as ``interaction obligations´´
- Each interaction obligation must be fulfilled by a correct implementation

The semantics of alt and xalt

$$[\![d_1 \; alt \; d_2]\!] \stackrel{\text{def}}{=} \{ (p_1 \cup p_2, n_1 \cup n_2) \mid (p_1, n_1) \in [\![d_1]\!] \wedge (p_2, n_2) \in [\![d_2]\!] \}$$

$$[\![d_1 \; xalt \; d_2]\!] \stackrel{\text{def}}{=} [\![d_1]\!] \cup [\![d_2]\!]$$

Property refinement of an interaction obligation

$$(p_1, n_1) \rightsquigarrow (p_2, n_2) \qquad \text{iff} \qquad n_1 \subseteq n_2 \wedge p_1 \subseteq p_2 \cup n_2$$

An interaction obligation (p2,n2) is a property refinement of an Interaction obligation (p1,n1) if and only if

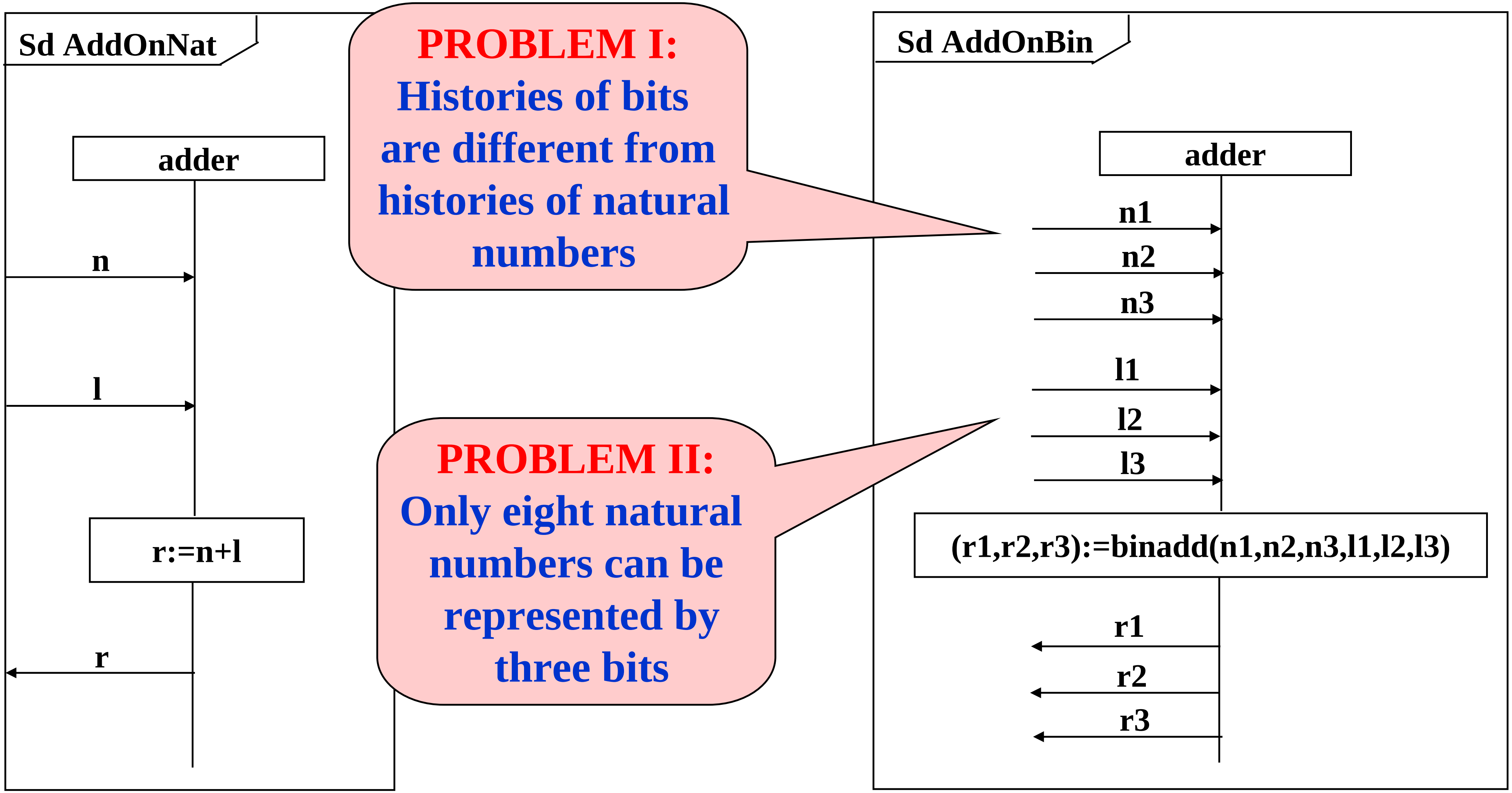
> **n1 is a subset of or equal to n2, and**
> **p1 is a subset of or equal to the union of p2 and n2**

Property refinement in STAIRS – general case

A sequence diagram d' is a refinement of a sequence diagram d, written $d \leadsto d'$, iff

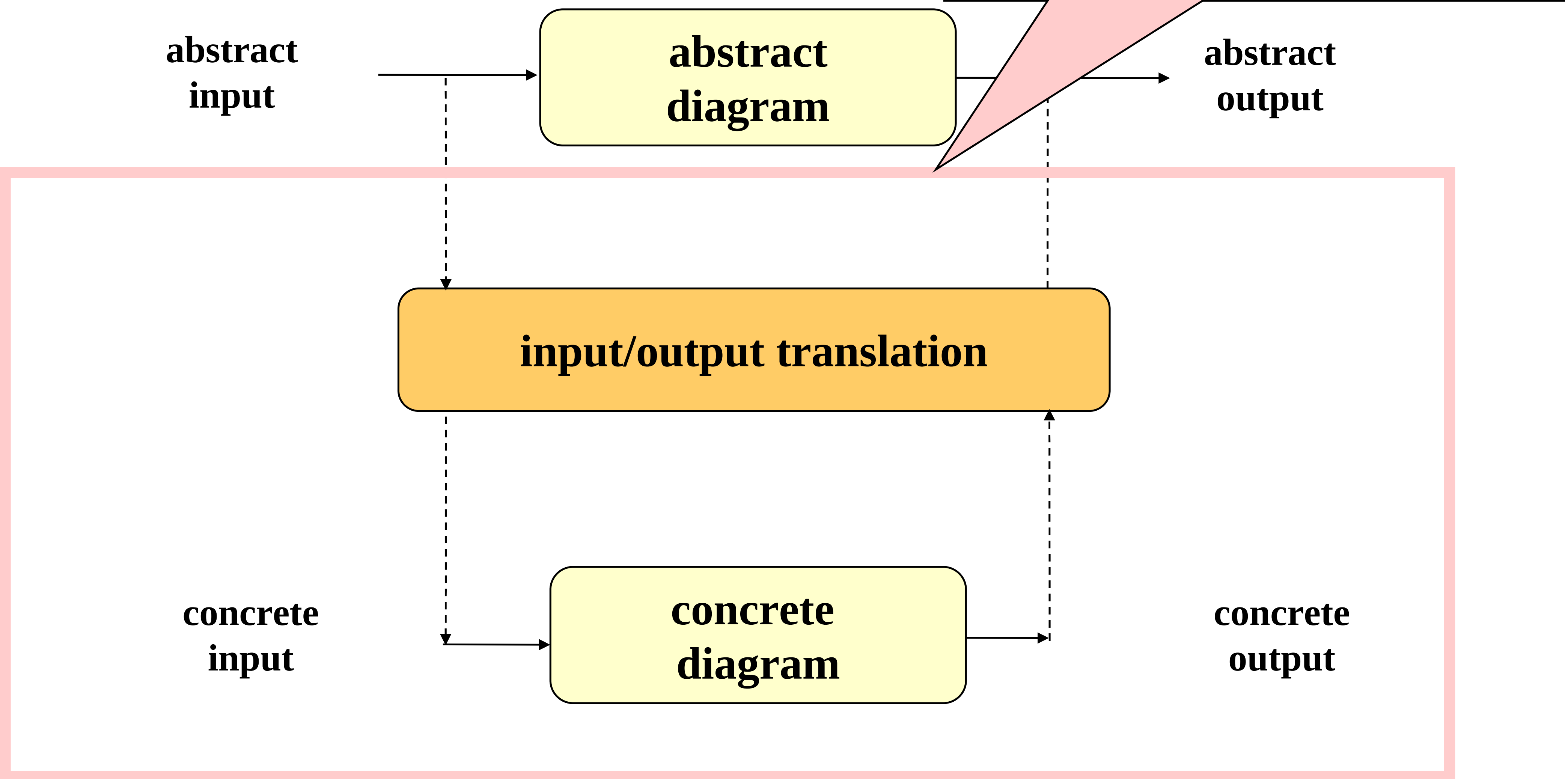
$$\forall o \in [\![d]\!] : \exists o' \in [\![d']\!] : o \leadsto o'$$

Property refinement is not enough



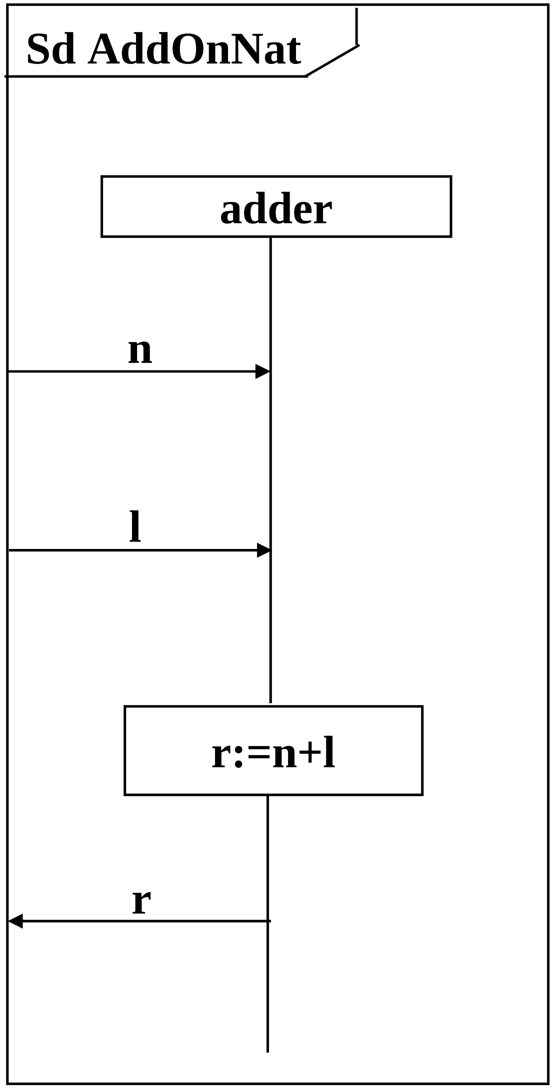
Interface refinement

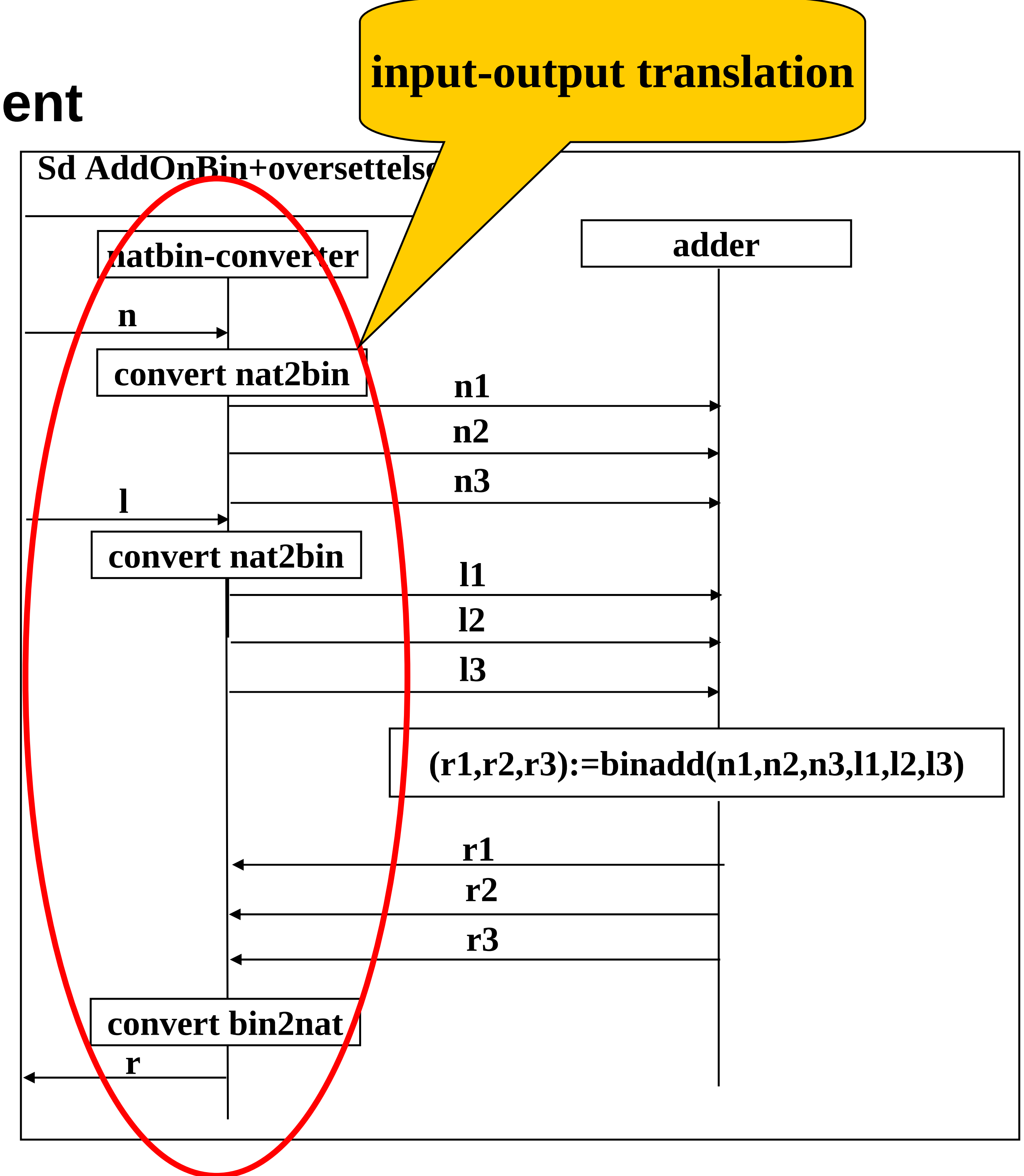
Requirement: Composition of concrete diagram and translation is a property refinement of abstract diagram



Øystein Haugen
Ketil Stølen

Interface refinement





Øystein Haugen Ketil Stølen

Interface refinement generalizes property refinement

interface refinement

property refinement

=

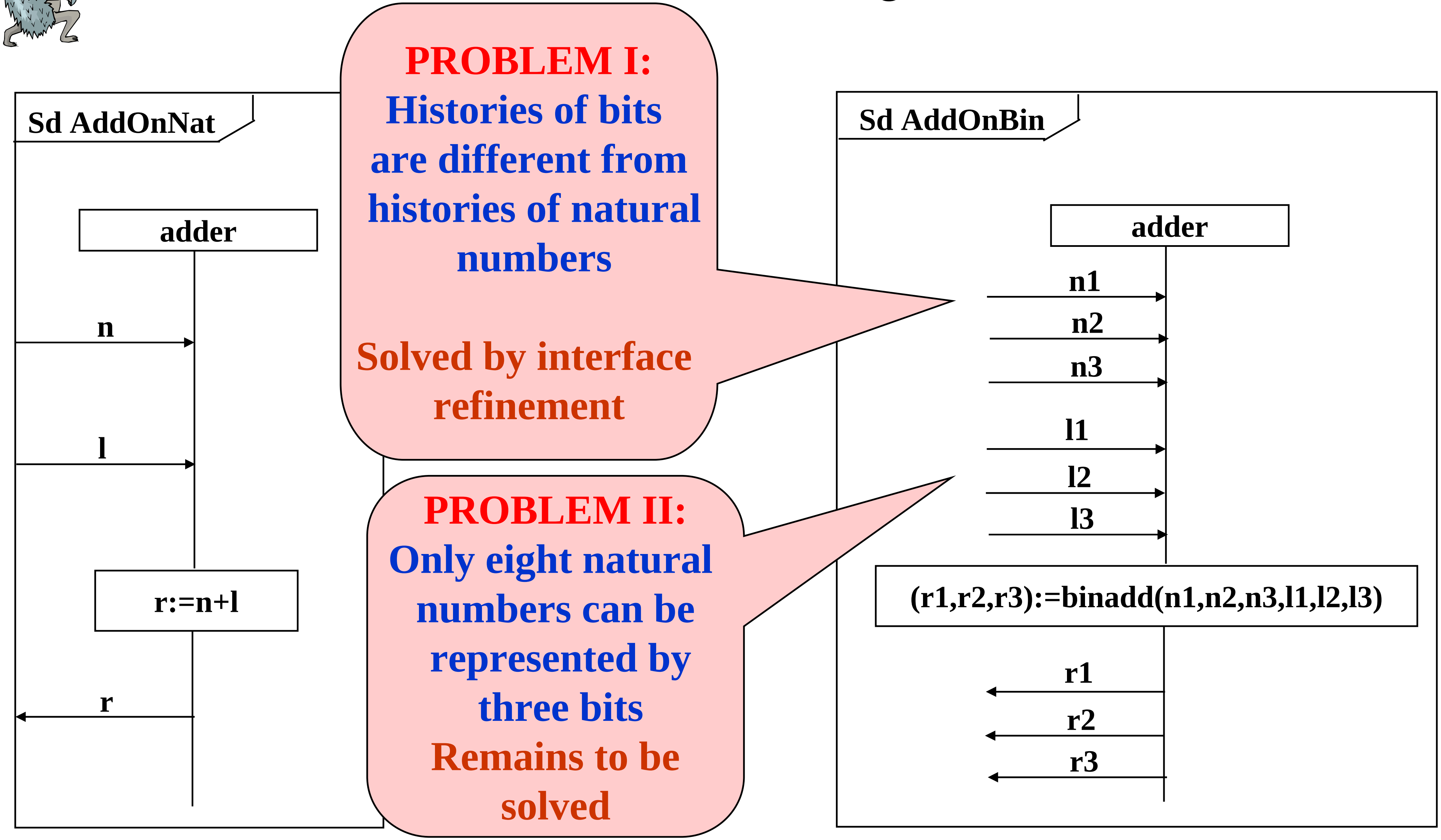
Interface refinement
wrt
identity translation

In STAIRS interface refinement is a special case of detailing

Øystein Haugen

Ketil Stølen

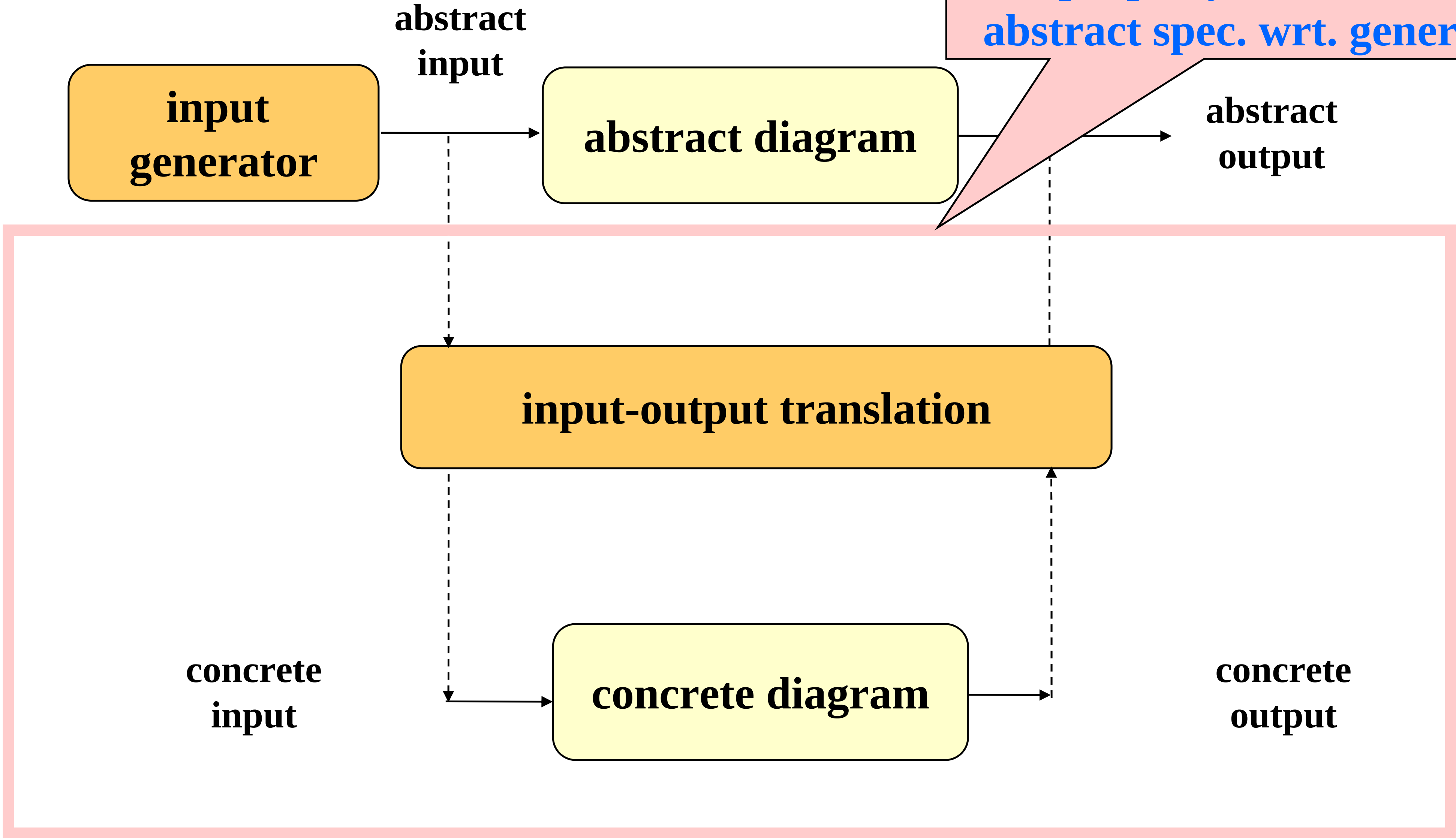
Interface refinement is not enough



Øystein Haugen Ketil Stølen

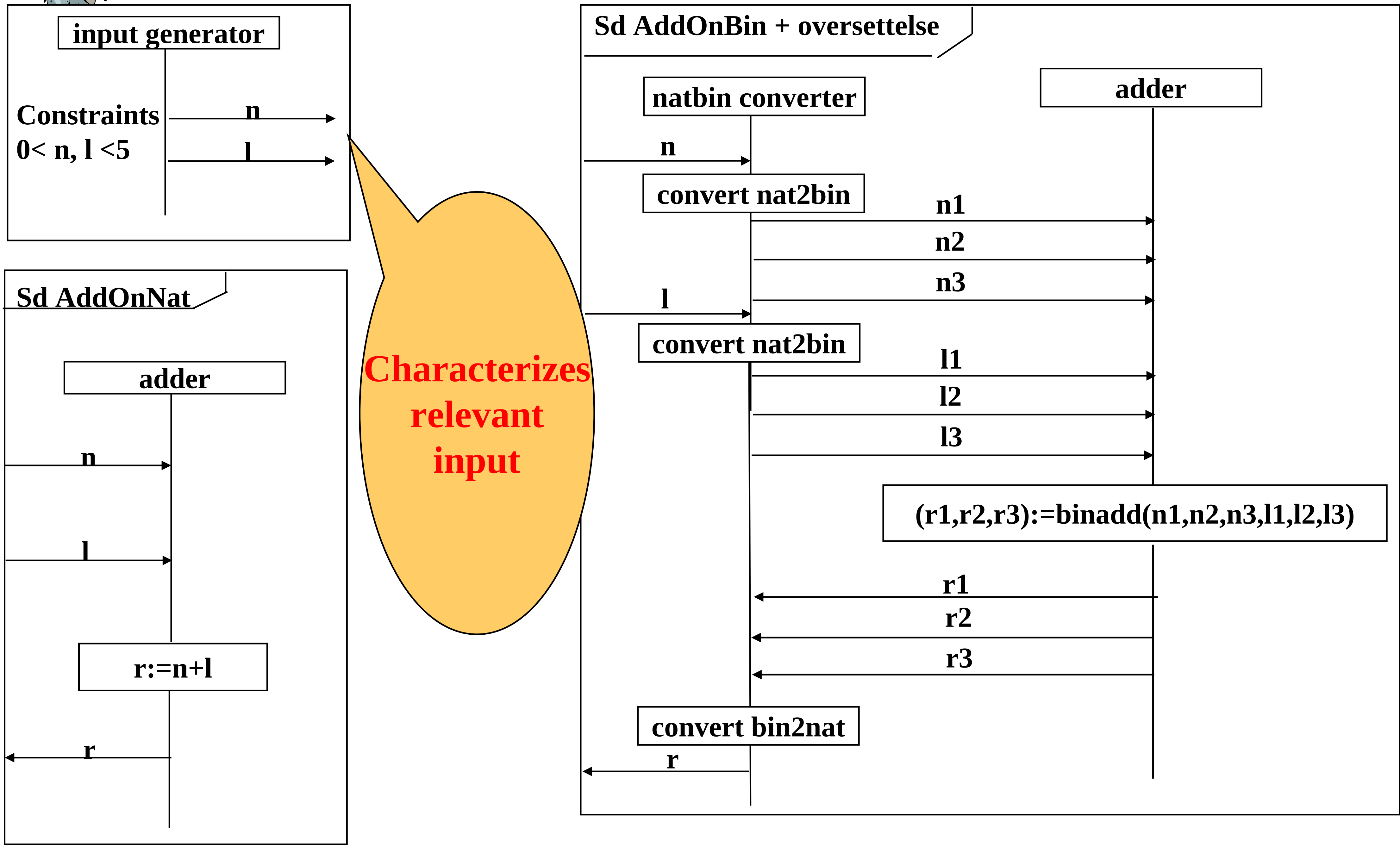
Conditional refinement

Requirement: Composition of concrete spec. and translation is a property refinement of abstract spec. wrt. generator



Øystein Haugen
Ketil Stølen

Conditional interface refinement (another form of detailing)



Øystein Haugen Ketil Stølen

Literature on refinement

- **The classics:**
  - C.A.R. Hoare. An axiomatic basis for computer programming. Communications of the ACM, 12:576-583, 1969
  - C.A.R. Hoare. Proof of correctness of data representations. Acta Informatica, 1:271-282, 1972
  - O-.J. Dahl, C.A.R. Hoare. Hierarchial program structures. In structured programming, pages 175-220. Academic Press, 1972
  - E.W.Dijkstra. A discipline of programming. Prentice-Hall, 1976

- **STAIRS is to a large extent based on ideas taken from:**
  - Manfred Broy, Ketil Stølen. Specification and Development of Interactive Systems - FOCUS on Streams, Interfaces and Refinement. ISBN 0-387-95073-7, Springer, 2001

Security analysis

- **First of three lectures October 28**

- **We will**
  - **classify dependability concepts**
  - **introduce motivate and explain a basic terminology for risk management and risk analysis**
  - **relate risk management to system development**
  - **describe and motivate the different processes of risk management and how they interact**