# Refinement I


## Ketil Stølen

# Objectives for the lectures on refinement

- Motivate the role of refinement
- Introduce and related the following notions of refinement
  - supplementing
  - narrowing
  - detailing
- Illustrate the use of these notions of refinement
  - the interplay between specification and refinement
- Illustrate the translation of theory into practice

# Three main concepts of language theory

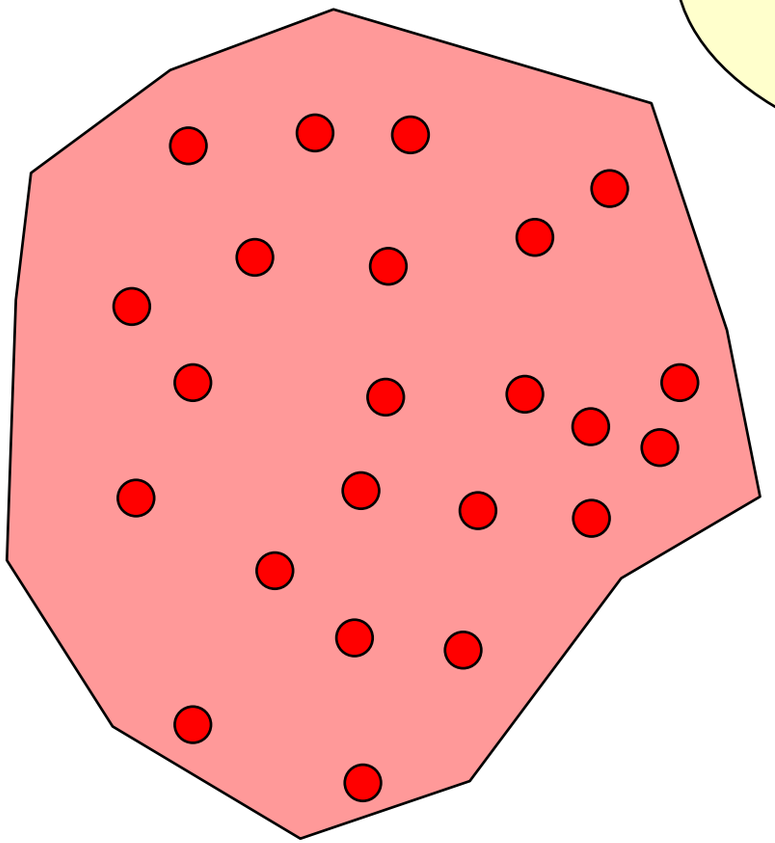
- **Syntax**
  - **The relationship between symbols or groups of symbols independent of content, usage and interpretation**
- **Semantics**
  - **The rules and conventions that are necessary to interpret and understand the content of language constructs**
- **Pragmatics**
  - **The study of the relationship between symbols or groups of symbols and their interpretation and usage**

# Semantic relation

Syntactically correct expressions in the language to be explained

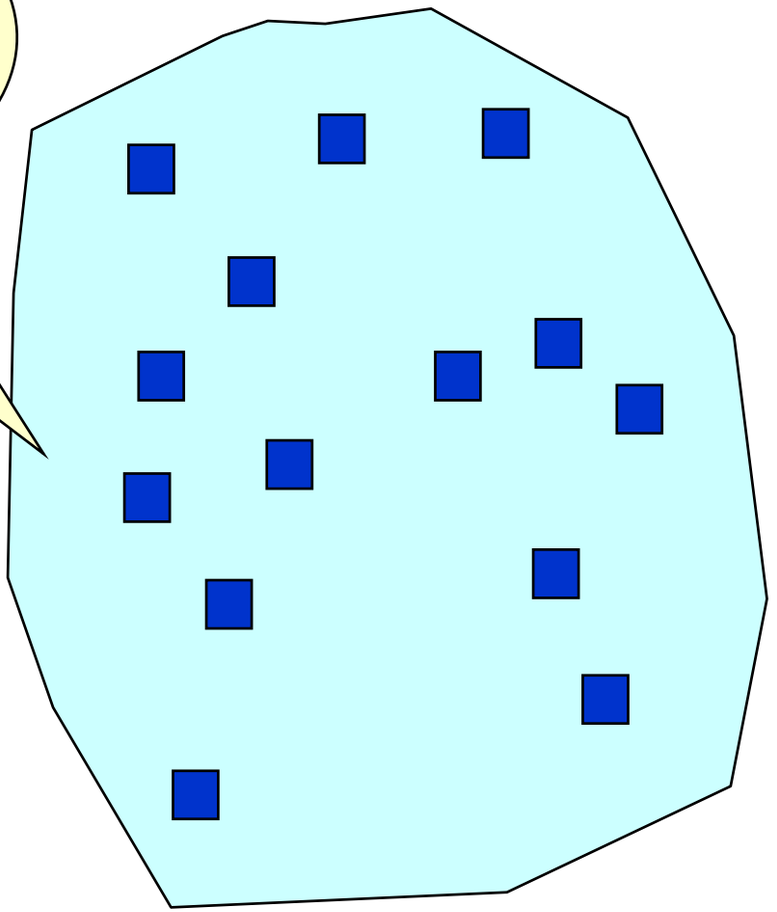
Syntactically correct expressions in a language that is well-understood

**What does it mean that a language is well-understood?**



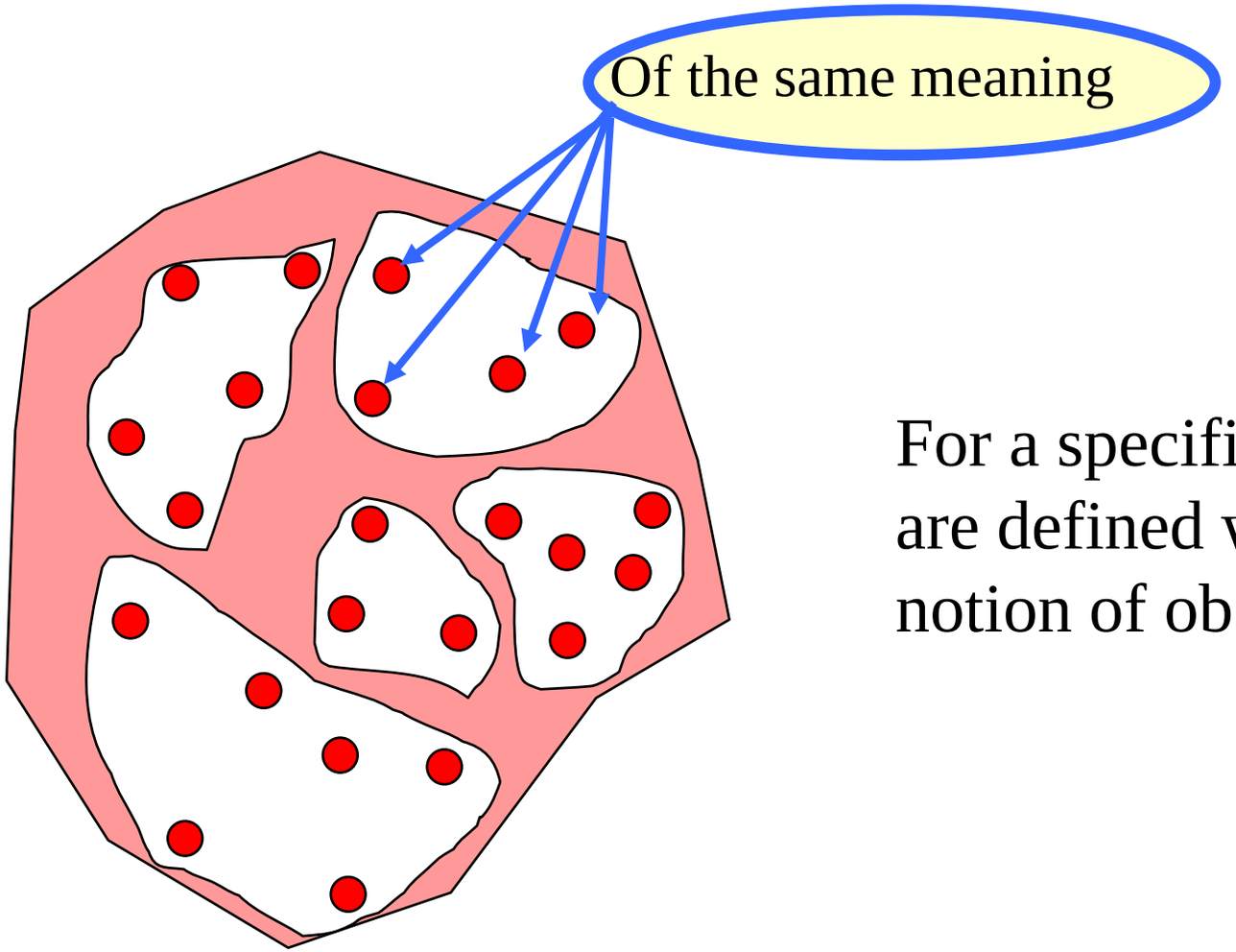
Semantic relation

Relates expressions that need interpretation to expressions that are well-understood



# The need for a notion of observation

- **A semantic relation will define an equivalence relation on the language that should be understood**




For a specification language these are defined with respect to a notion of observation

# Definition of a notion of observation

- May observe only external behavior
- May observe that nothing bad happens
- May observe that something eventually happens
- May observe any potential behavior
- May observe time with respect to a global clock

**May our notion of observation be implemented by a human being?**

# Pre-post specifications

# The origins of refinement

# 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

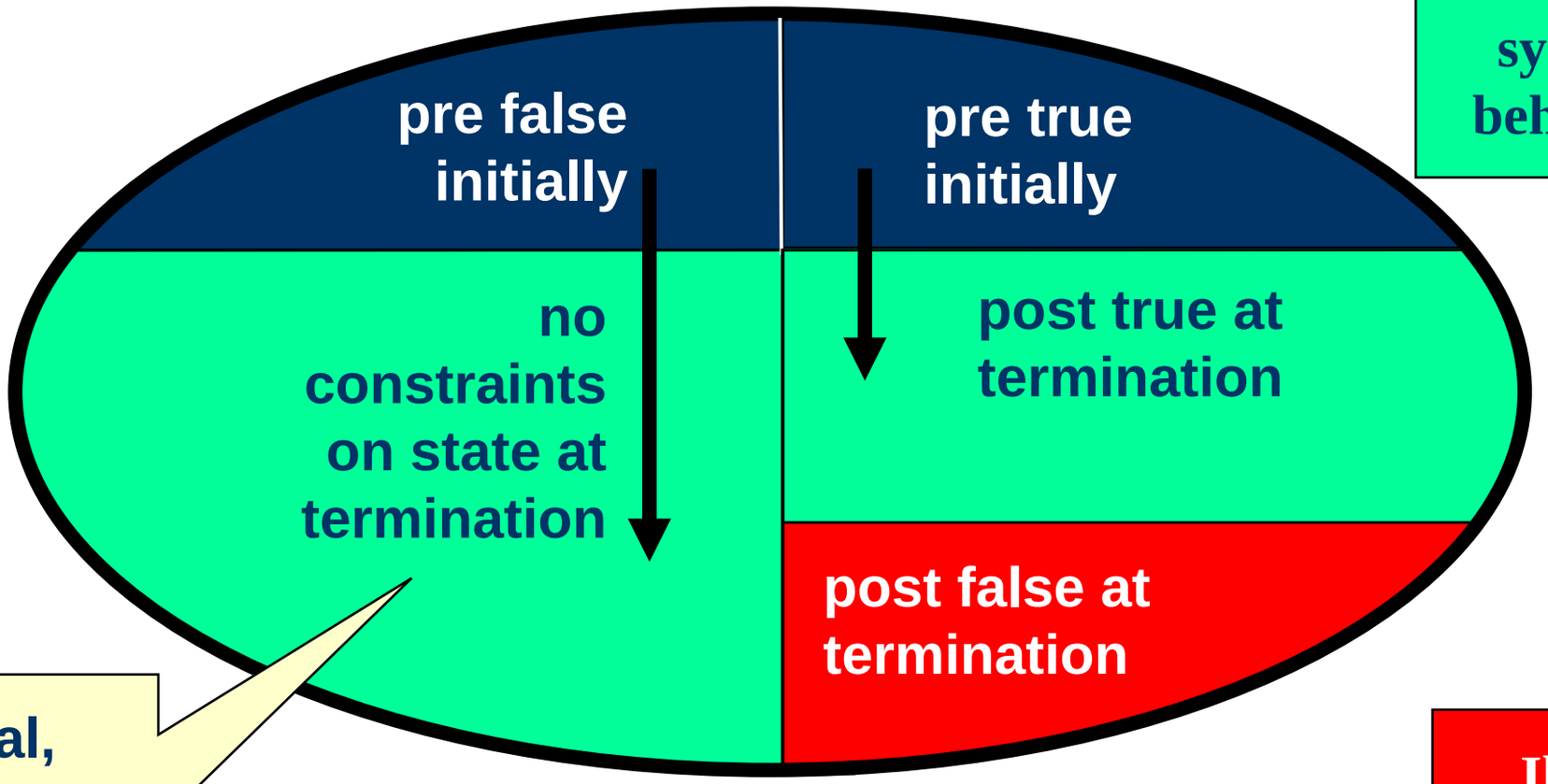
Assumption about the state at the moment the execution is initiated

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

rest' < divisor

Guarantee with respect to the state at the moment of termination

# Semantics of pre-post specifications

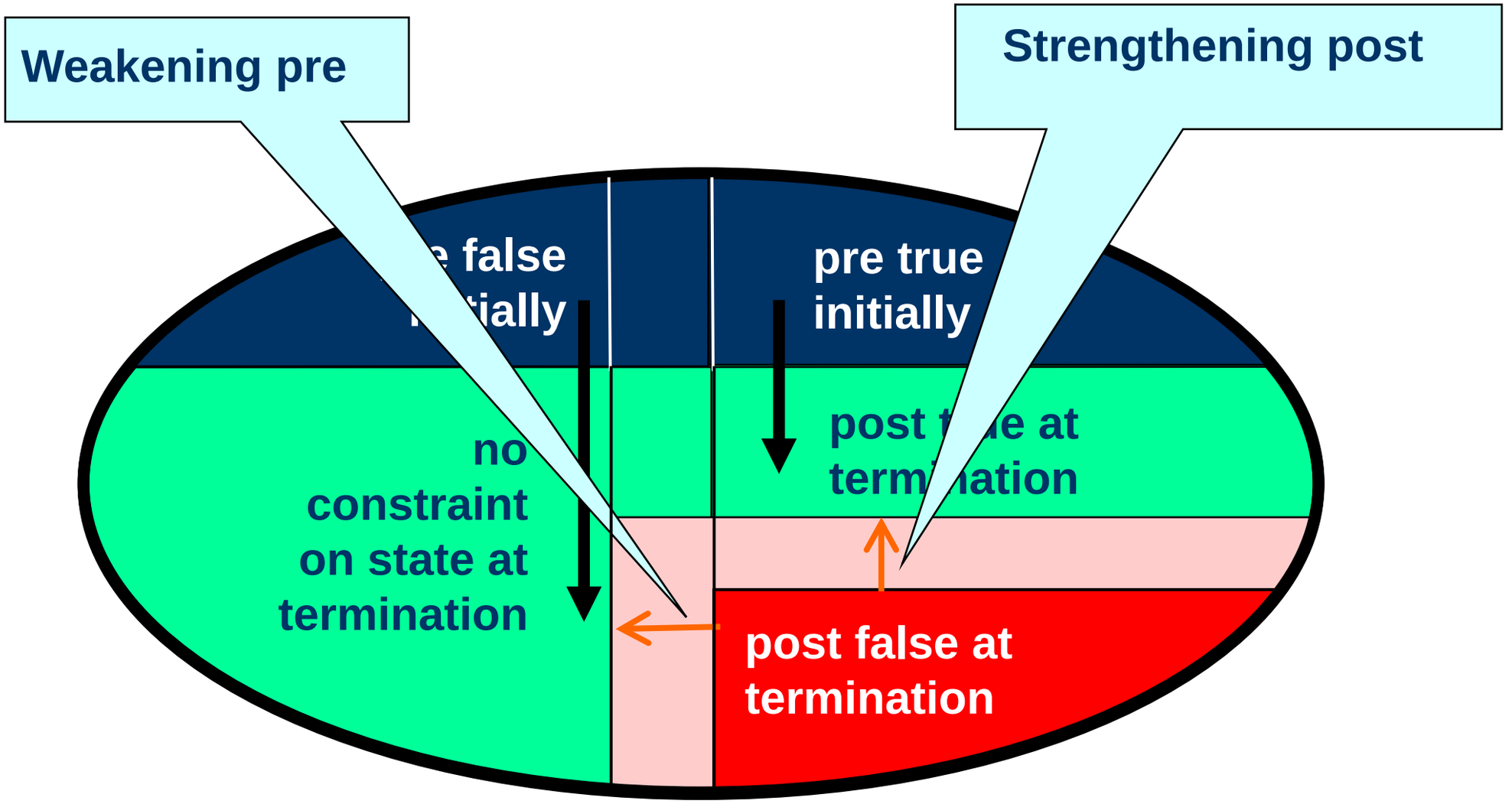



Legal system behavior

Legal, but arbitrary behavior

Illegal system behavior

# Refinement in pre-post



# Weakening the pre-condition (assumption)

Integer division

**var** dividend, divisor, quotient, rest : Nat

---

**pre** **true**

**post**

**if** divisor $\neq$ 0 then

( dividend = (quotient' * divisor) + rest' ) & rest' < divisor

**else** quotient' = 0

# Strengthening the post-condition (guarantee)

Integer division

**var** dividend, divisor, quotient, rest : Nat

---

**pre** divisor $\neq$ 0

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

rest' < divisor & dividend' = dividend &

divisor' = divisor

# STAIRS

# Refinement in UML

# Motivation

- Exploit classical theory of refinement in a practical UML setting
  - From theory to practice, and not the other way around
- Sequence diagrams can be used to capture the meaning of other UML description techniques for behavior
- By defining refinement for sequence diagrams we therefore implicitly define refinement for UML

# Traces for sequence diagrams summarized

- Traces for sequence diagrams are sequences of events

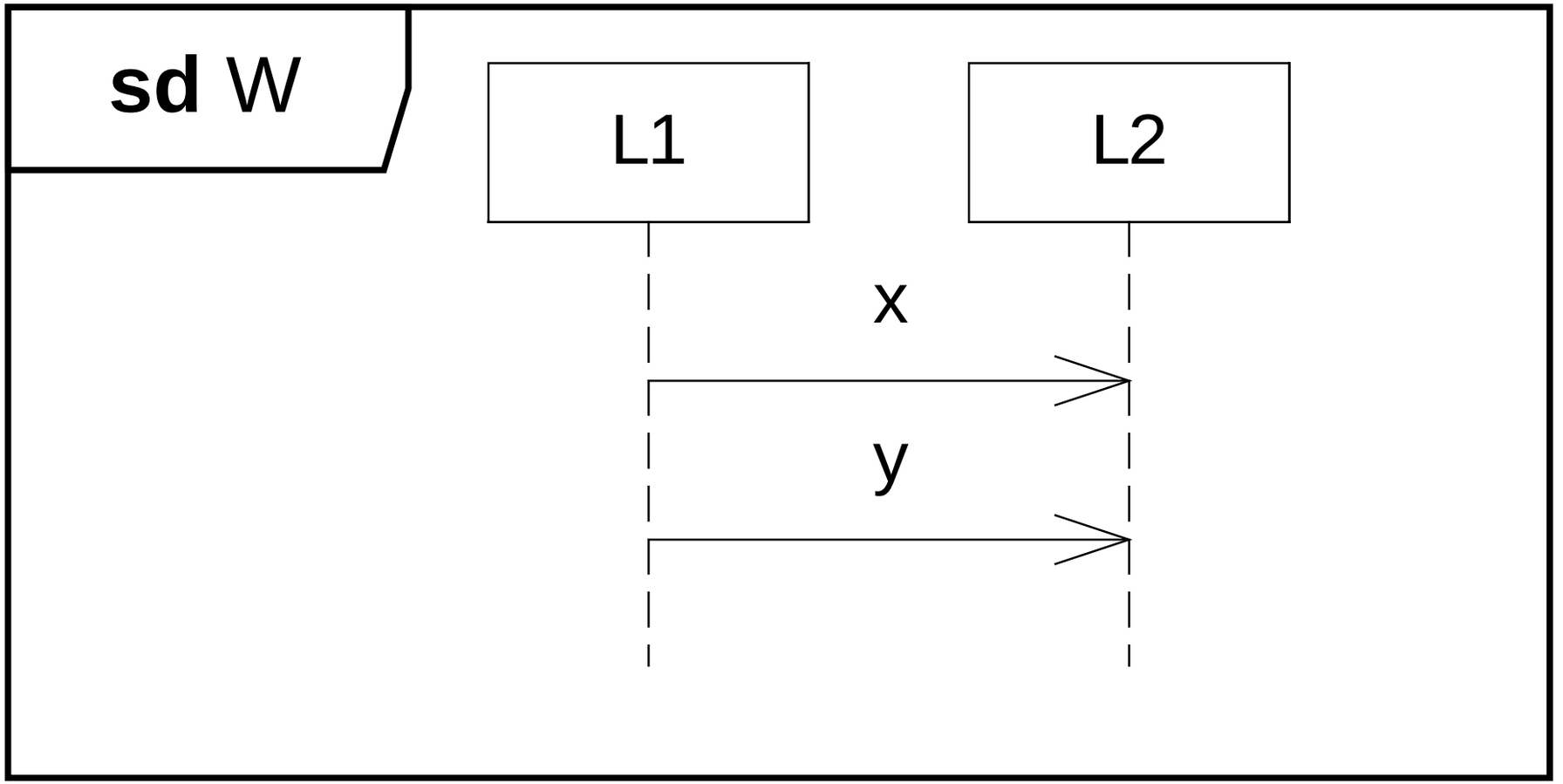
  <e1, e2, e3, e4, e4, e1, e2, e5, ……>

- An event represent either the transmission or reception of messages
  - ?m - reception of message m
  - !m - transmission of message m
- Events are instantaneous
- A trace may be finite
  - termination, deadlock, infinite waiting, crash
- A trace may also be infinite
  - infinite loop, intended non termination

# Causality and weak sequencing

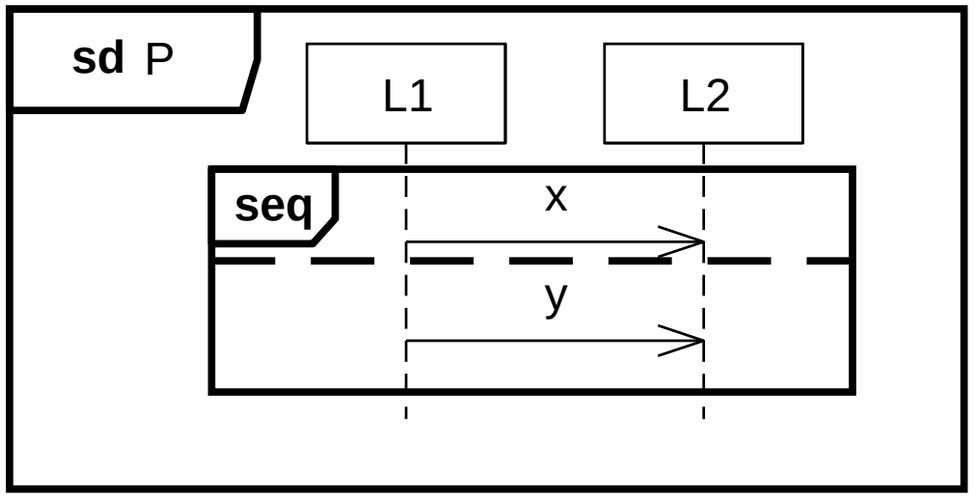
- Causality:
  - a message can never be received before it has been transmitted
  - the transmission event for a message is therefore always ordered before the reception event for the same message
- Weak sequencing:
  - events from the same lifeline are ordered in the trace in the same order as on the lifeline

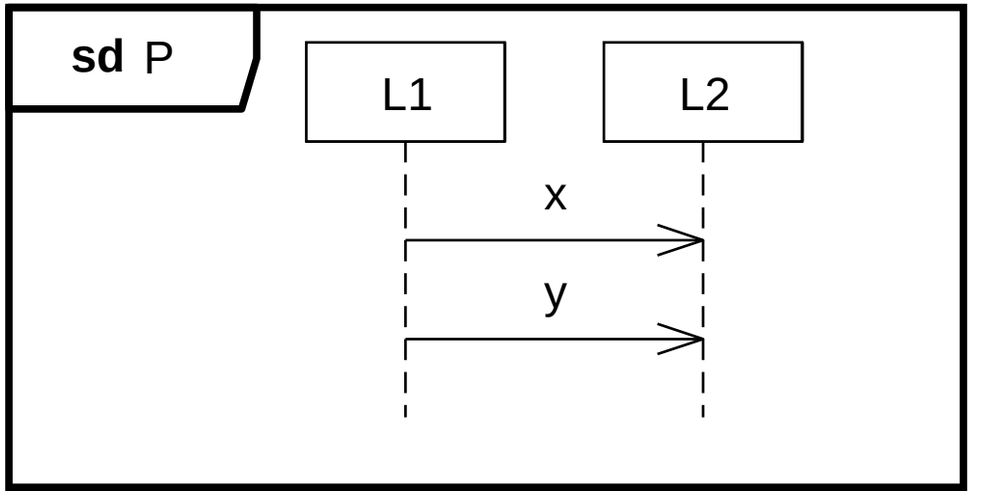
# Weak sequencing



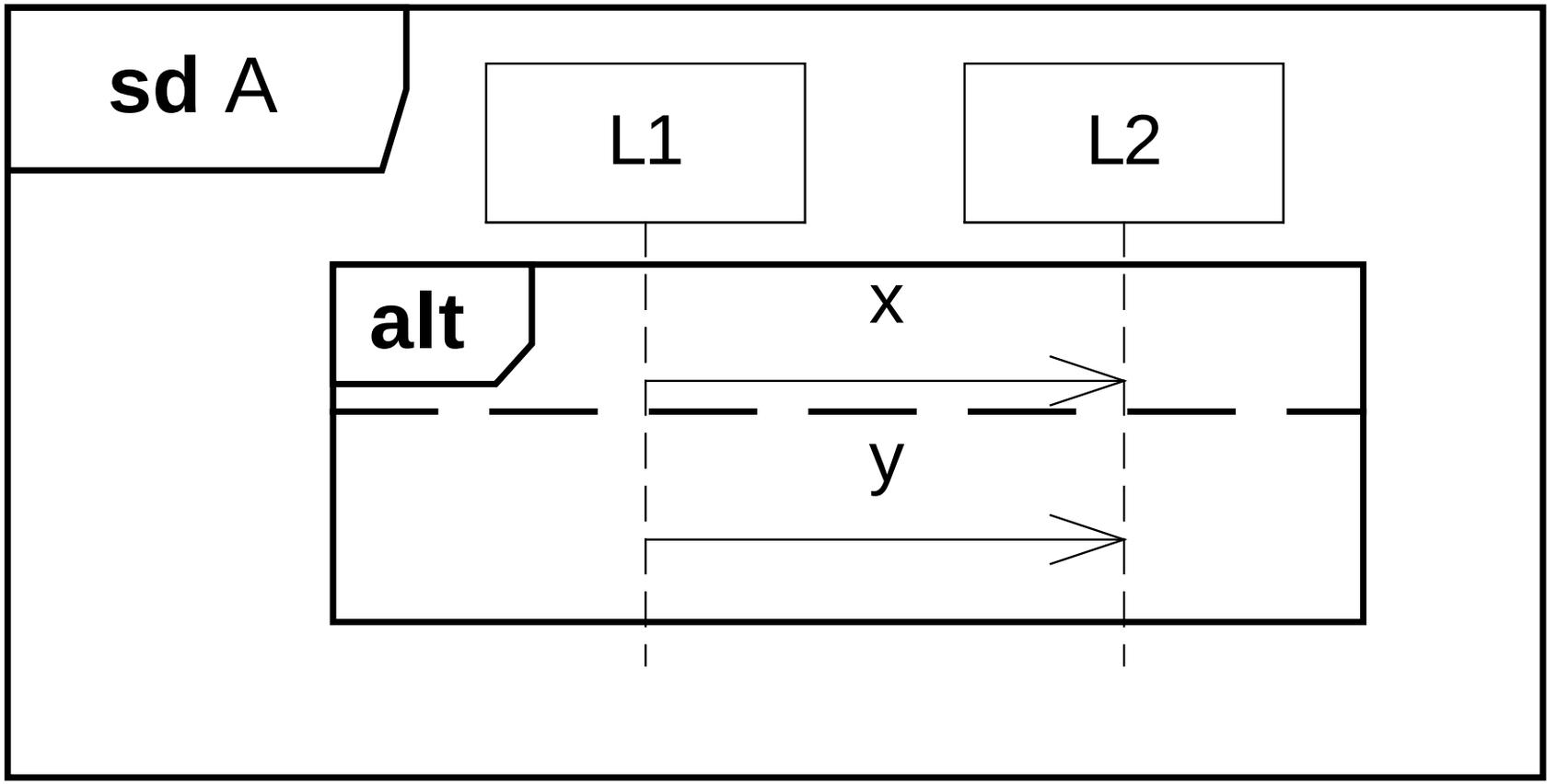

<!x,?x,!y,?y>
<!x,!y,?x,?y>

# These two diagrams are semantically the same

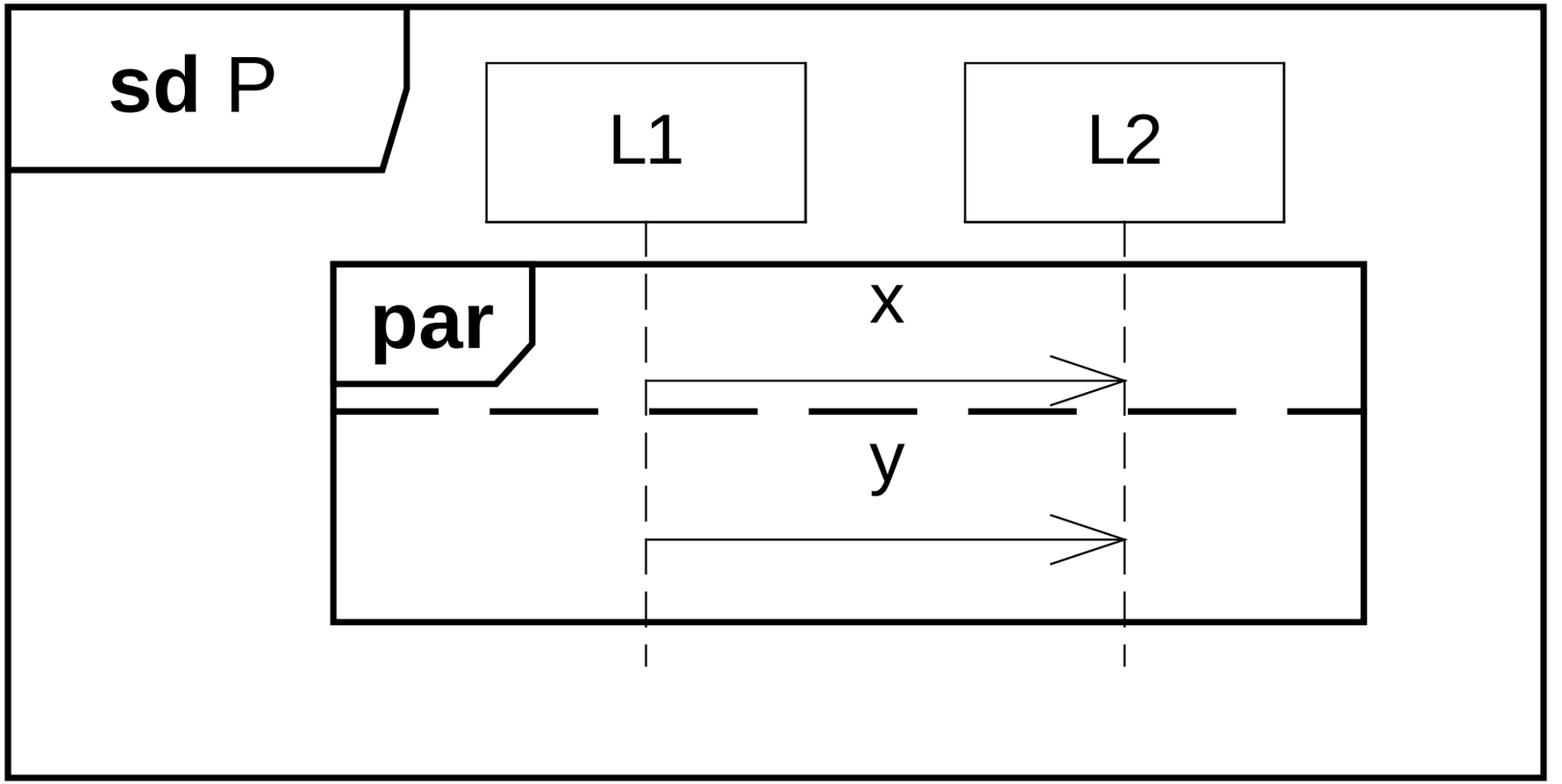


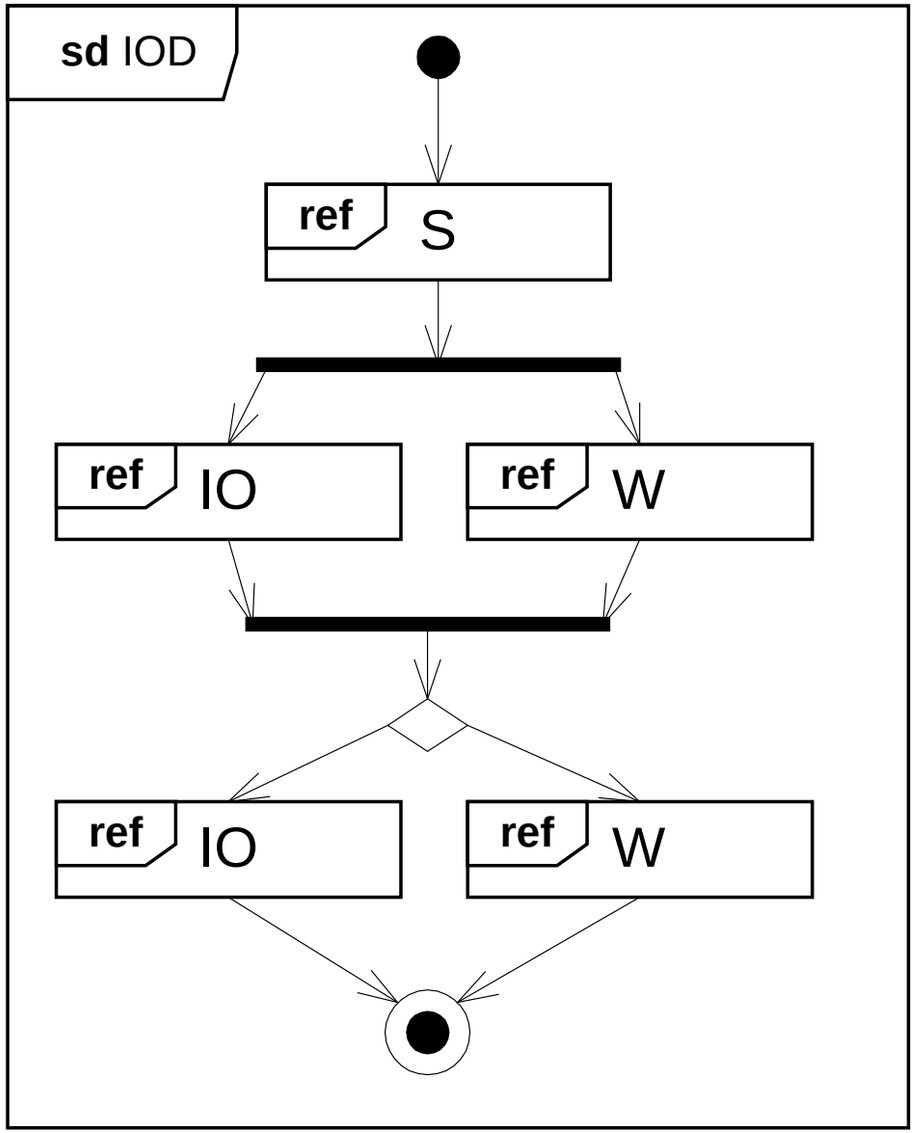
# Alternative composition



# Parallel composition

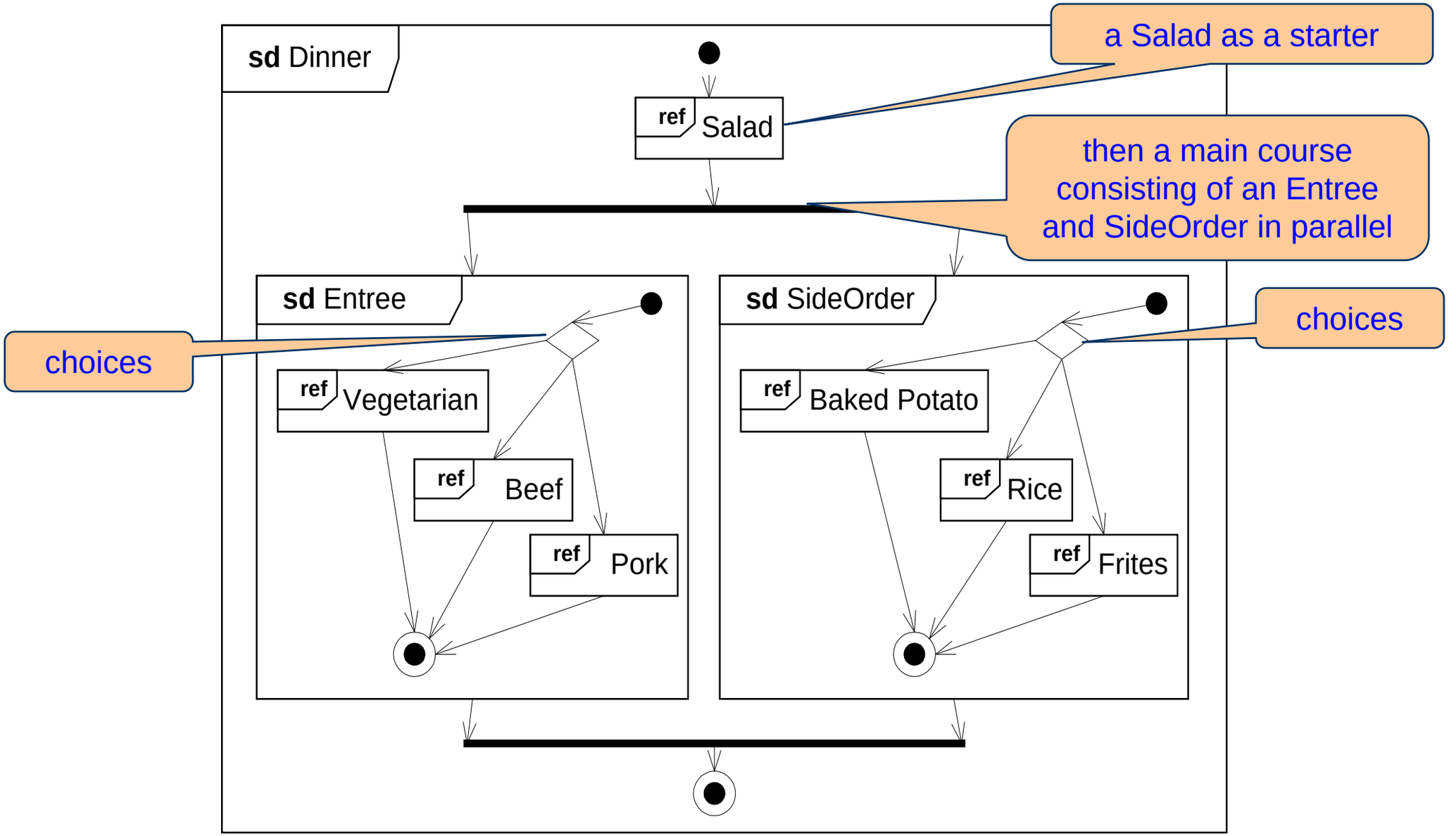


# Interaction overview diagram

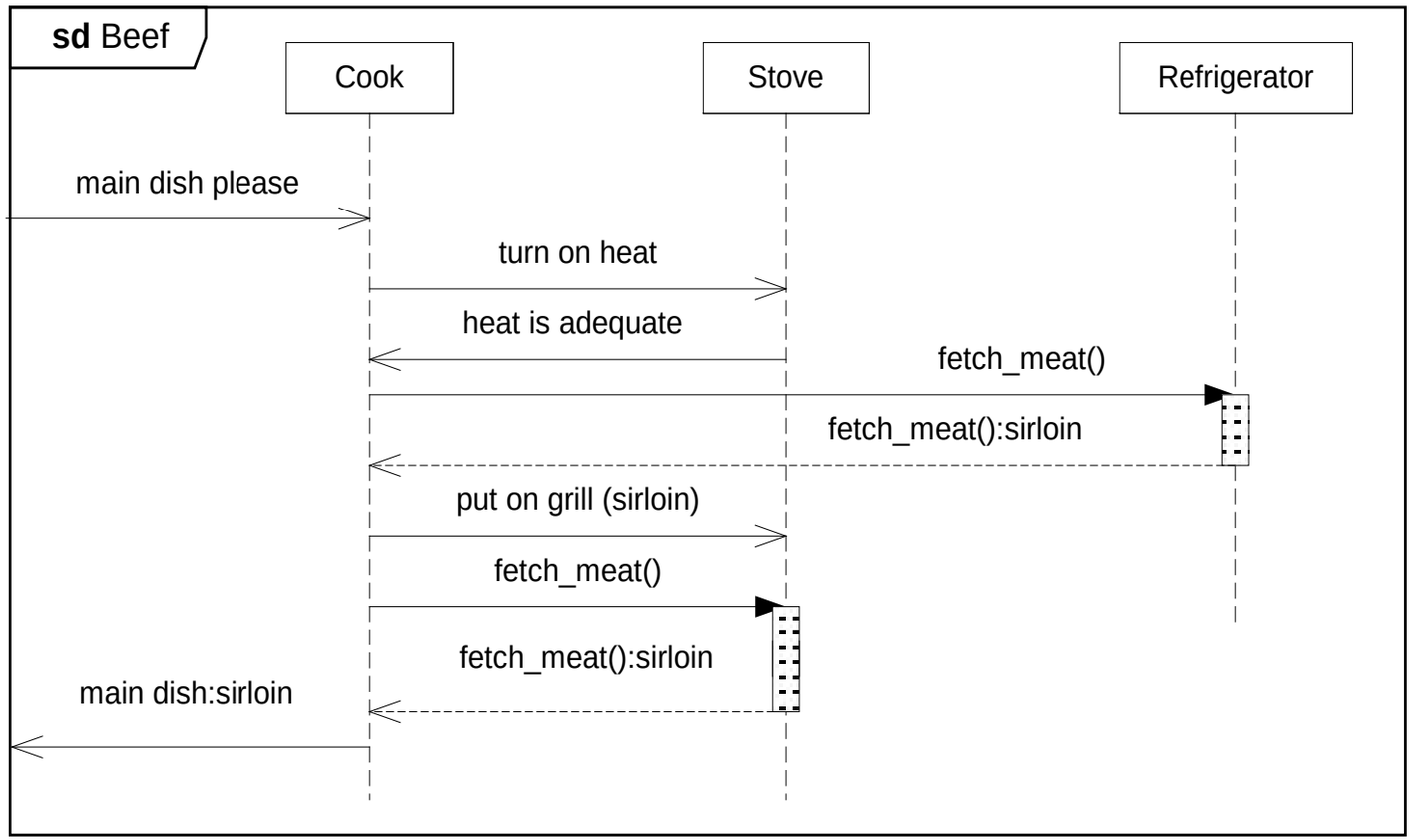



S **seq** (IO **par** W) **seq** (IO **alt** W)

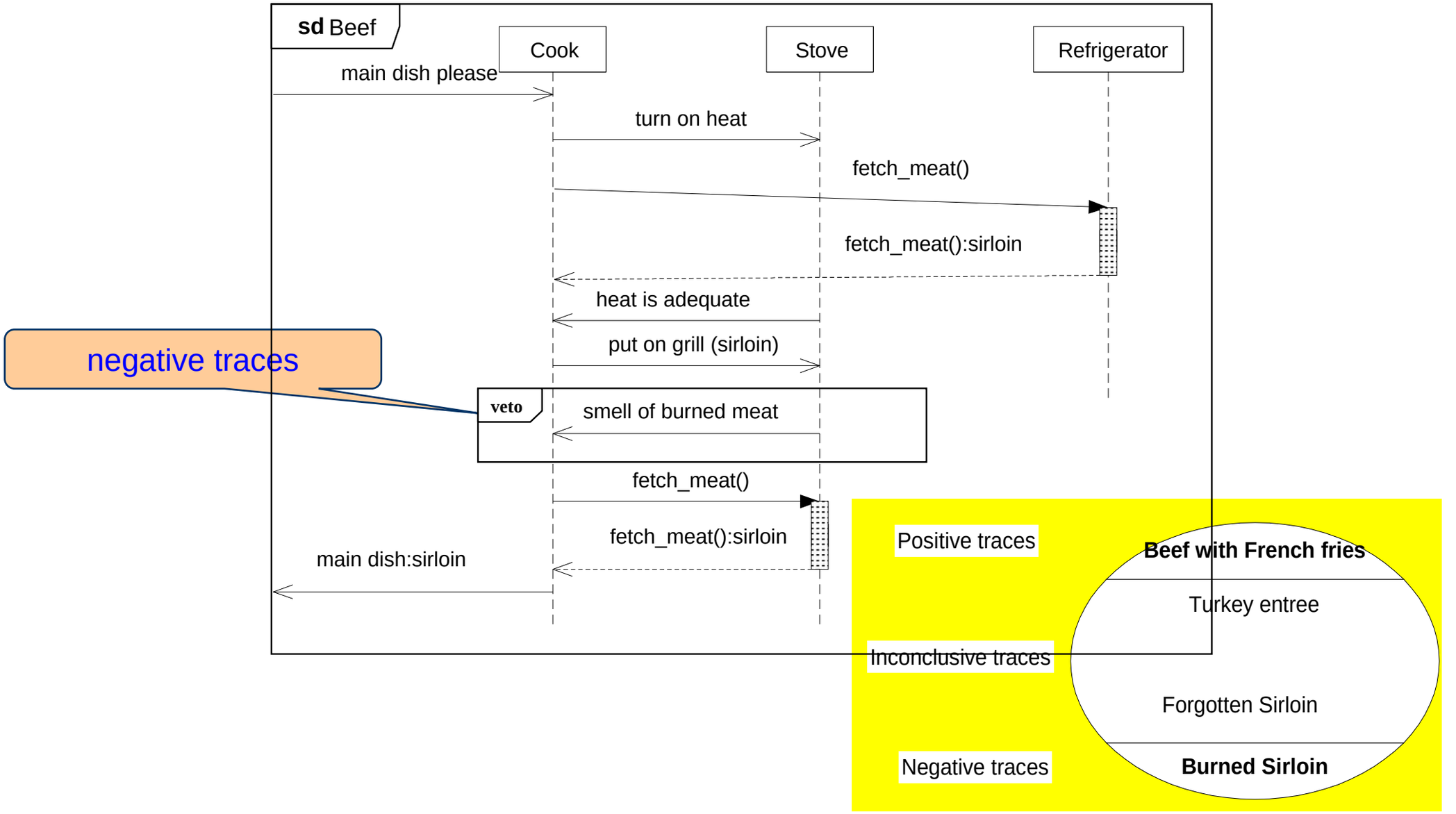
# Dinner


sd Dinner
ref Salad
a Salad as a starter
then a main course consisting of an Entree and SideOrder in parallel
sd Entree
sd SideOrder
choices
choices
ref Vegetarian
ref Beef
ref Pork
ref Baked Potato
ref Rice
ref Frites

# Some potential positive traces of Beef

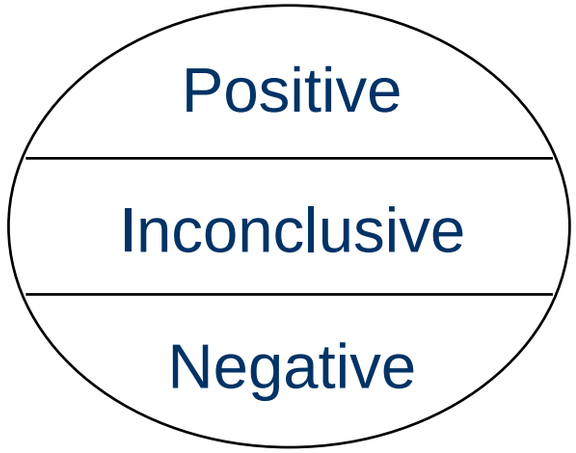


# Potential negative Beef experiences



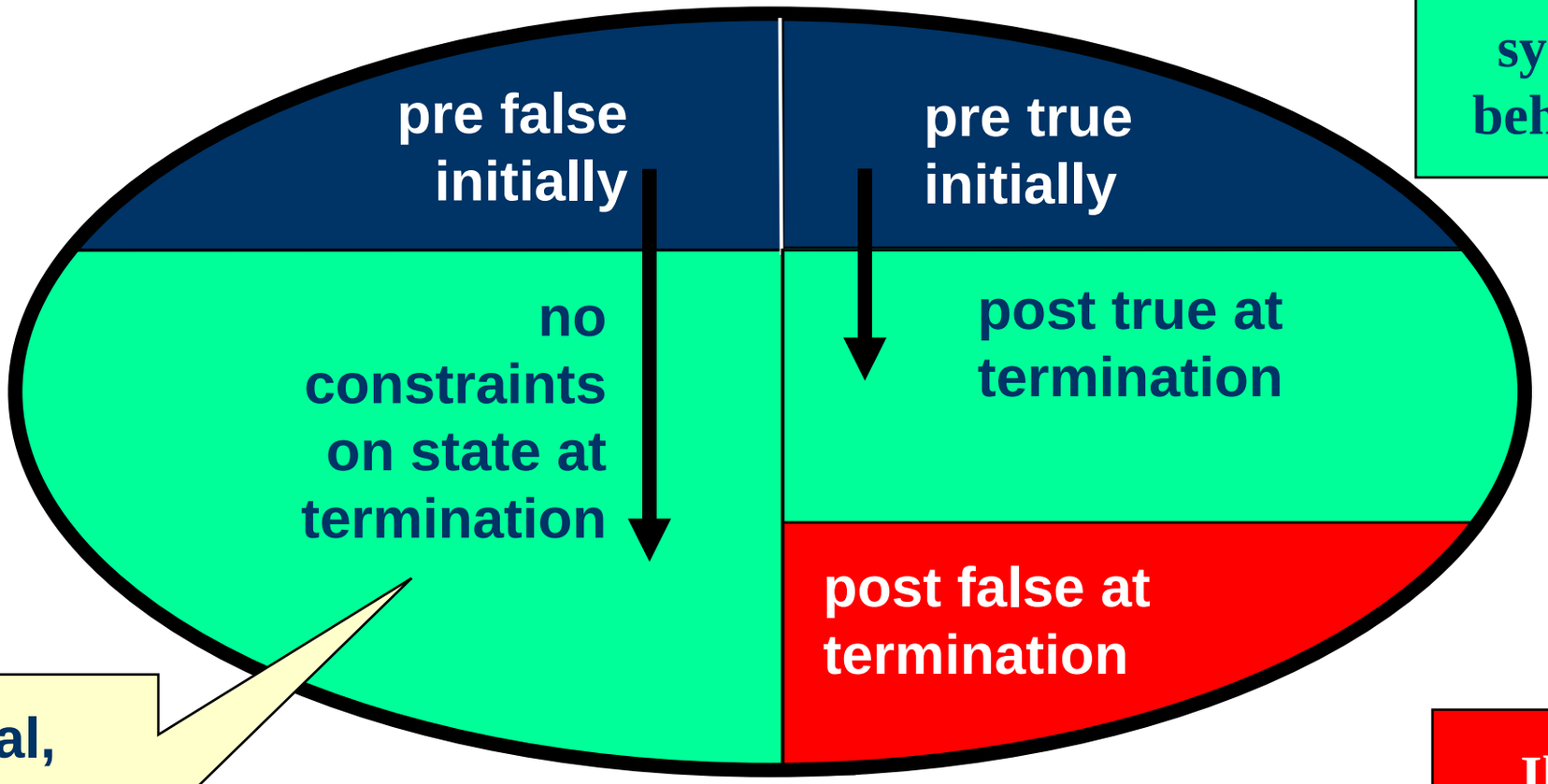
# STAIRS semantics: simple case

- Each positive execution is represented by a trace
- Each negative execution is represented by a trace
- The semantics of a sequence diagram is a pair of sets of traces (Positive, Negative)




- All other traces over the actual alphabet of events are inconclusive

# Semantics of pre-post specifications

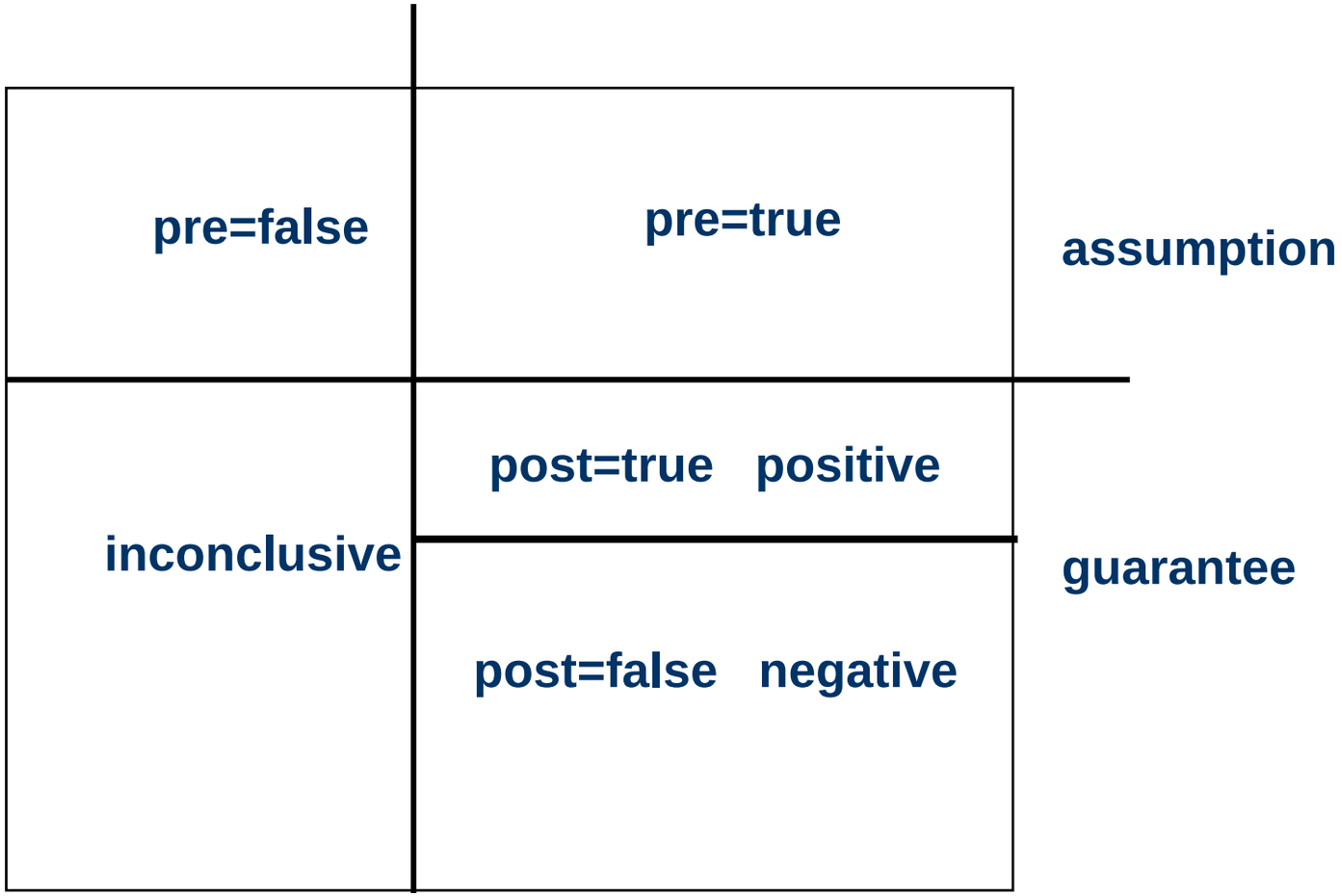



Legal system behavior

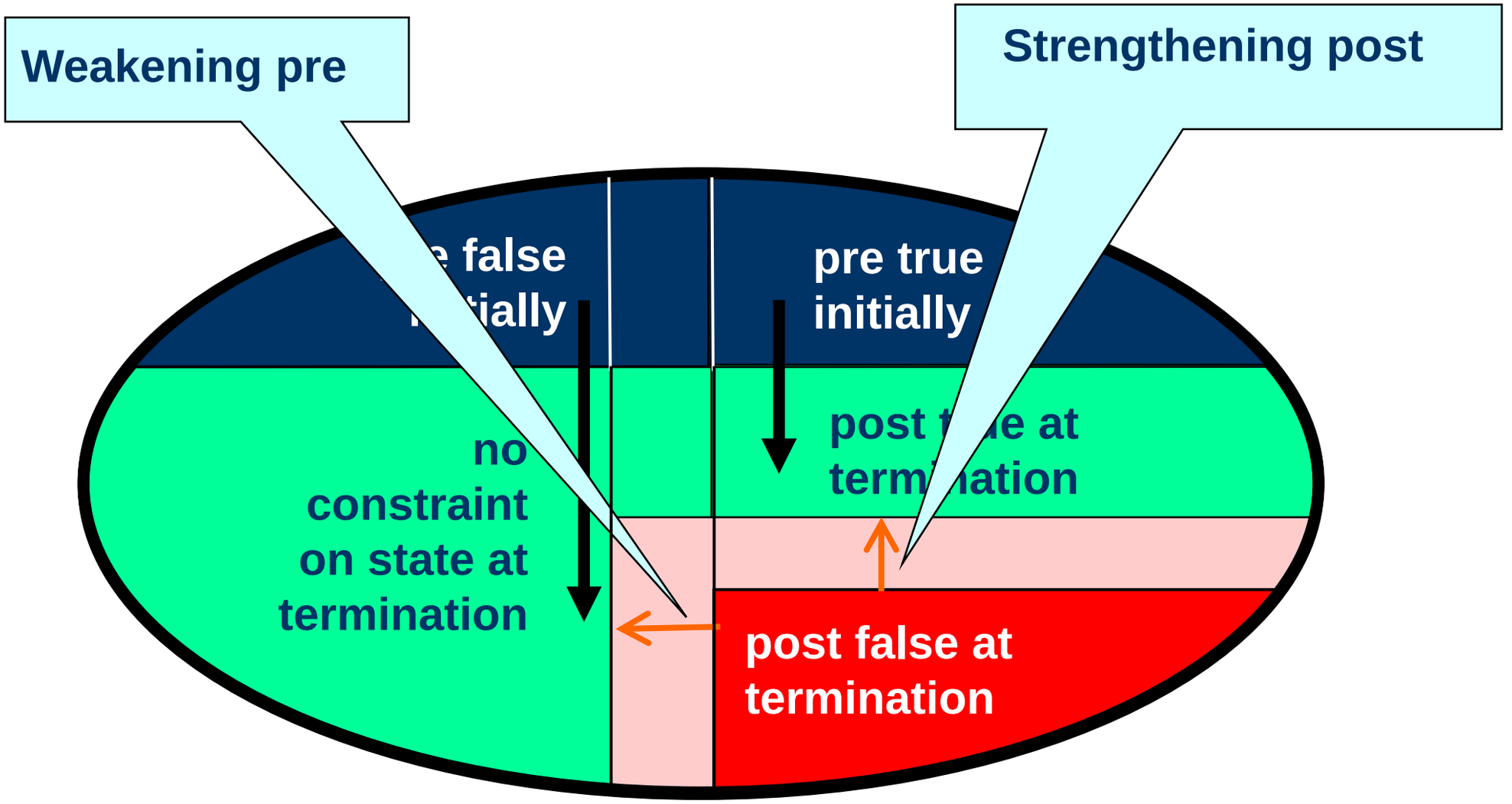
Legal, but arbitrary behavior

Illegal system behavior

# Comparing STAIRS with pre-post

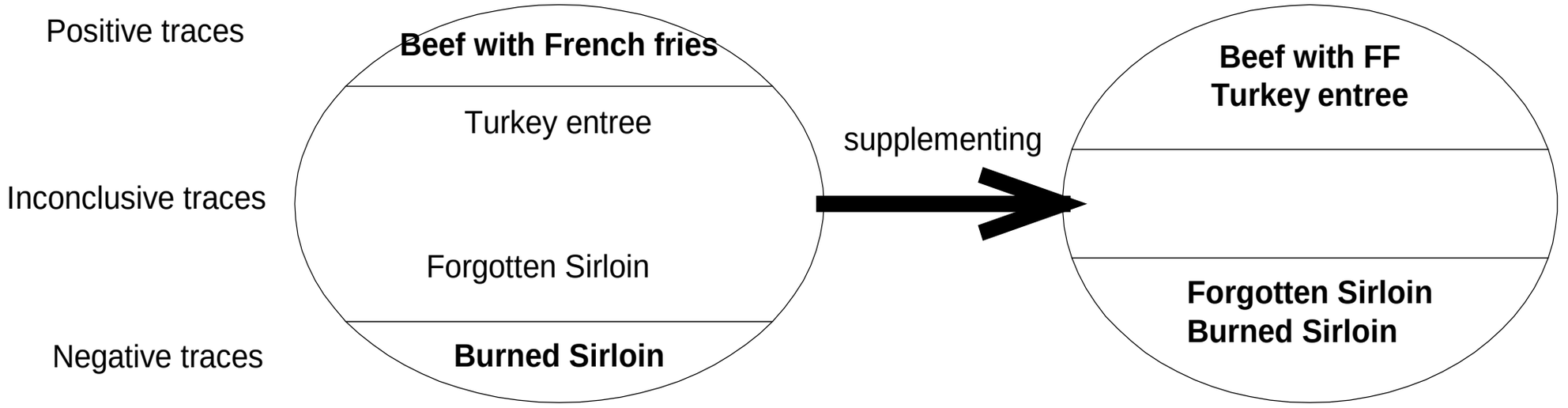


# Refinement in pre-post

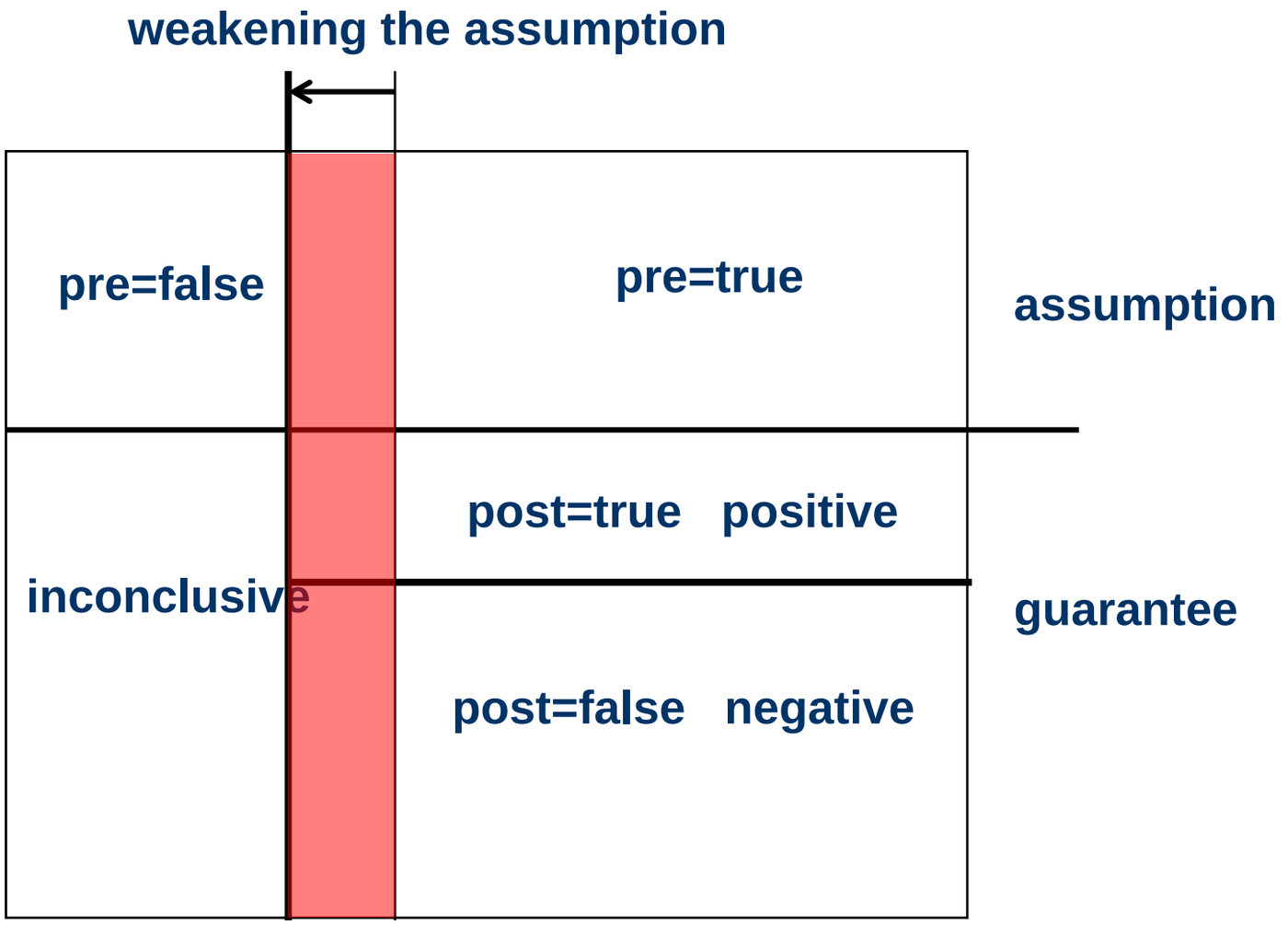


# STAIRS: supplementing

- Supplementing involves reducing the set of inconclusive traces by redefining inconclusive traces as either positive or negative
- Positive trace remains positive
- Negative trace remains negative

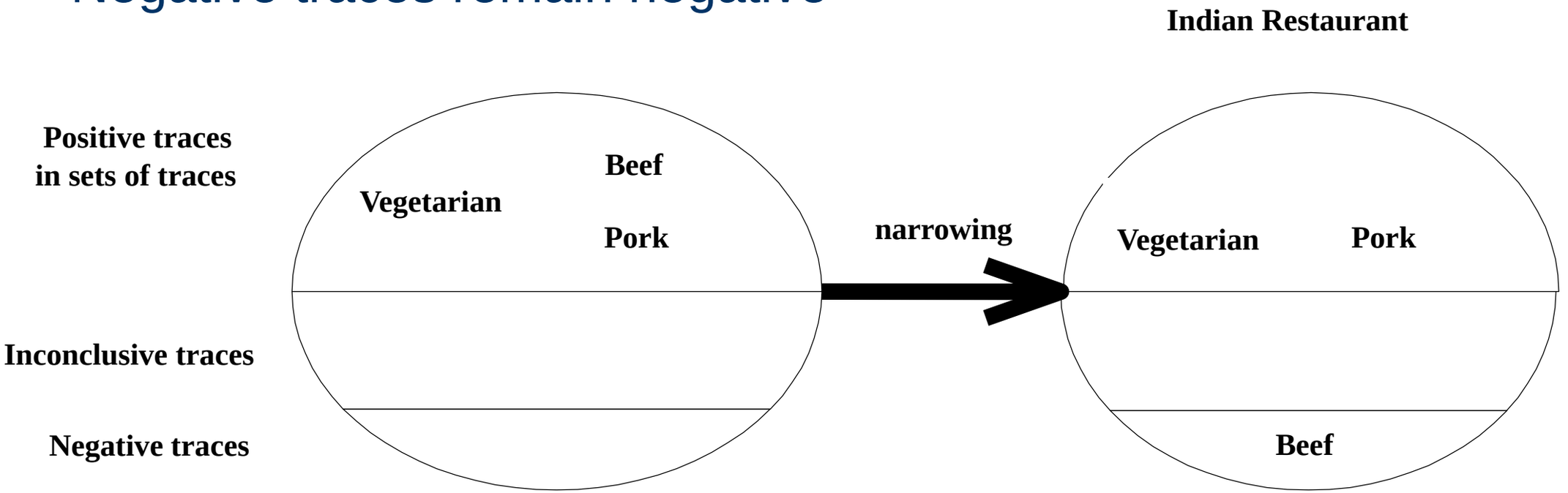


# Supplementing in pre-post

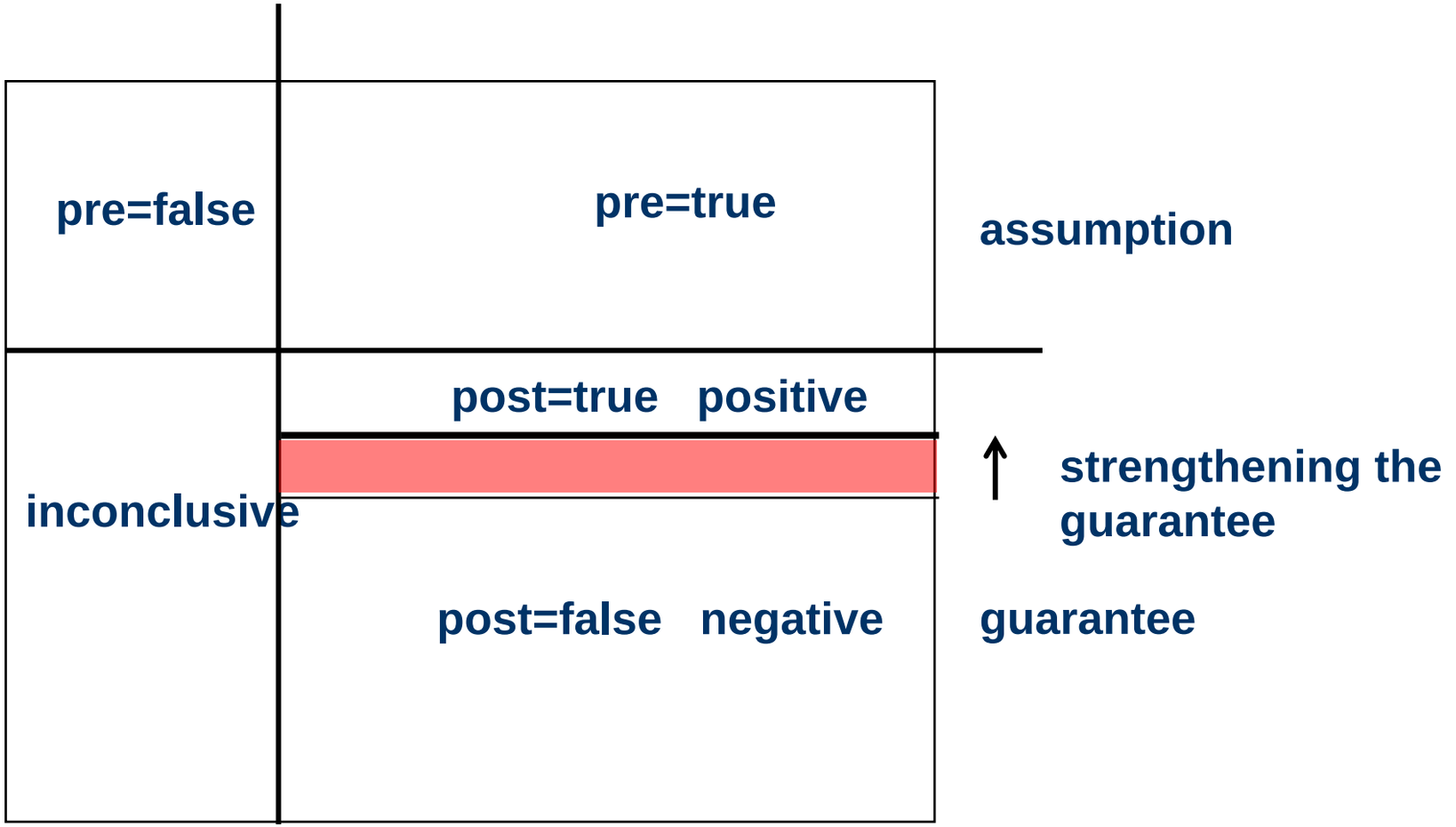


# STAIRS: narrowing

- Narrowing involves reducing the set of positive traces by redefining them as negative
- Inconclusive traces remain inconclusive
- Negative traces remain negative



# Narrowing in pre-post



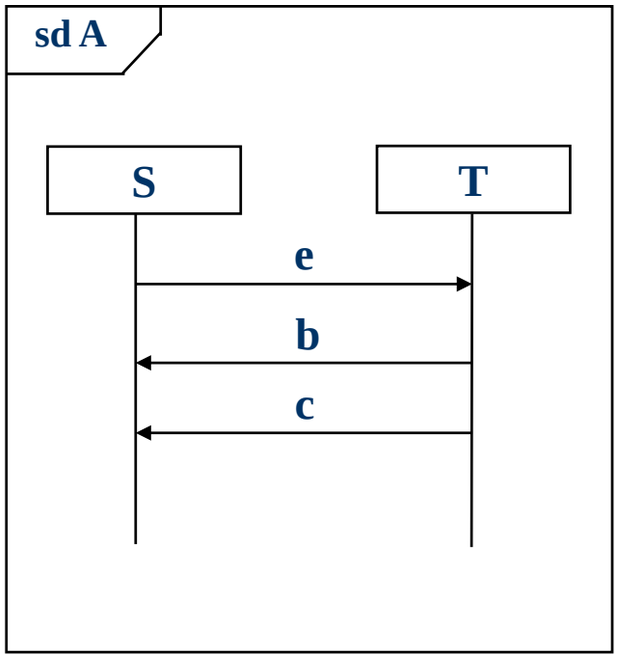
# Indirect definition: Refinement in STAIRS

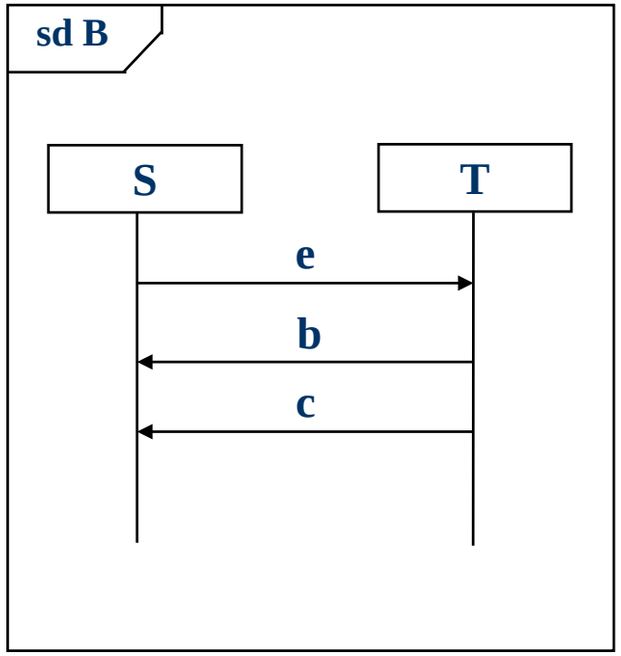
- A sequence diagram B is a general refinement of a sequence diagram A if
  - A and B are semantically identical
  - B can be obtained from A by supplementing
  - B can be obtained from A by narrowing
  - B can be obtained from A by a finite number of steps

    A -> C1 -> C2 -> …. ->Cn->B

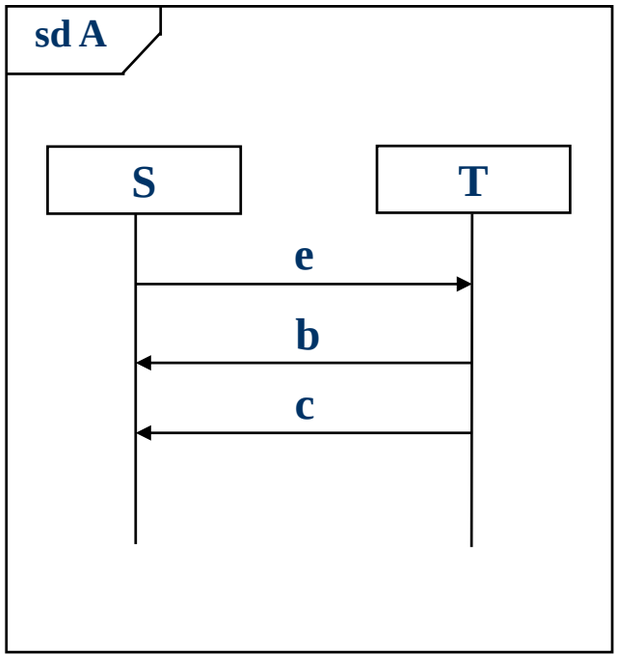
    each of which is either a supplementing or a narrowing

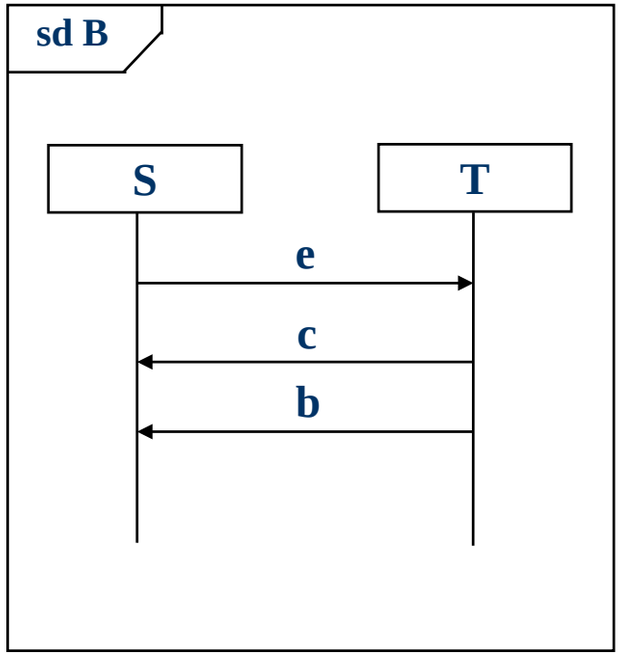
# Is B a refinement of A?



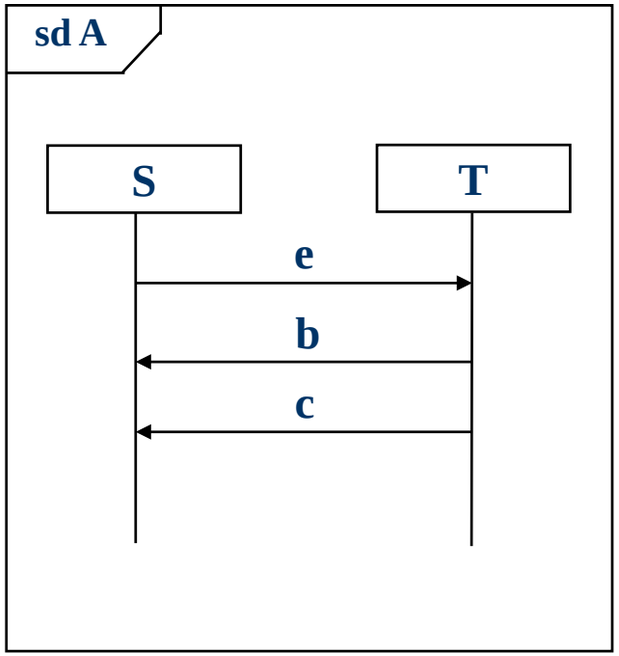



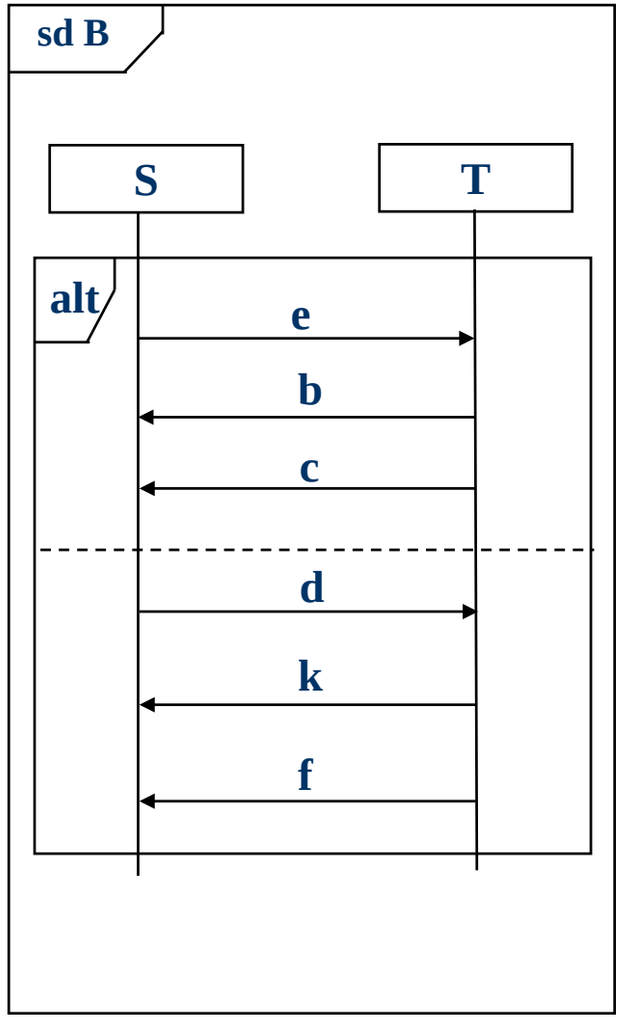
# Is B a refinement of A?



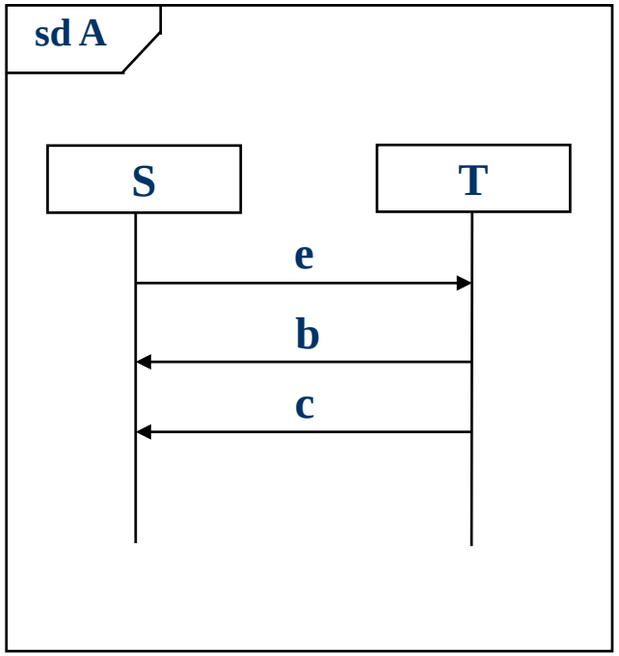



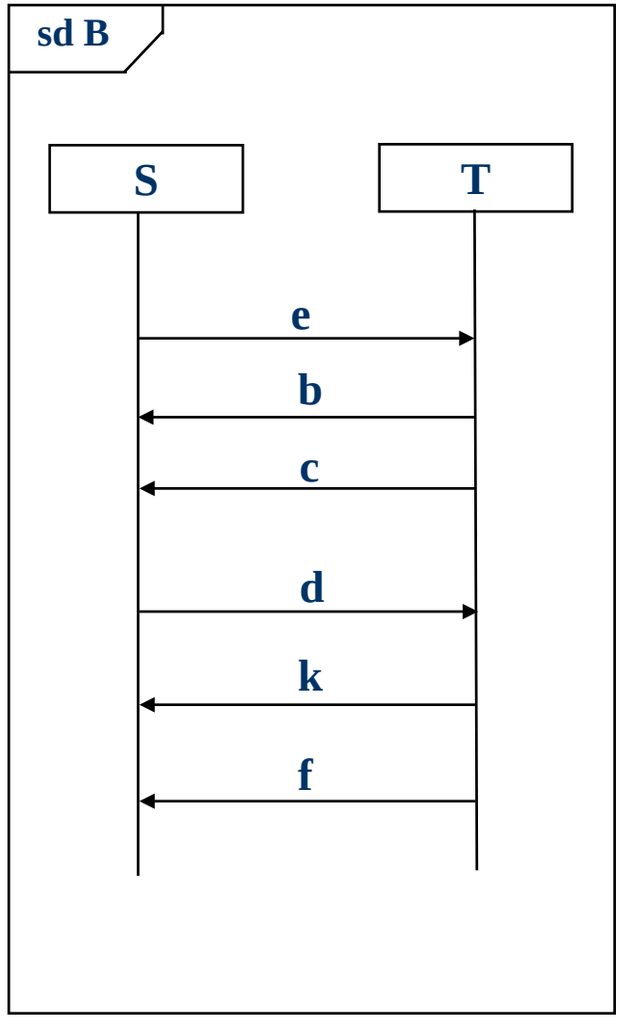
# Is B a refinement of A?



# Is B a refinement A?



# Is B a refinement of A?