Abstract Interpretation

a unified lattice model for static analysis of programs by construction or approximation of fixpoints

Patrick Cousot and Radhia Cousot, 1977

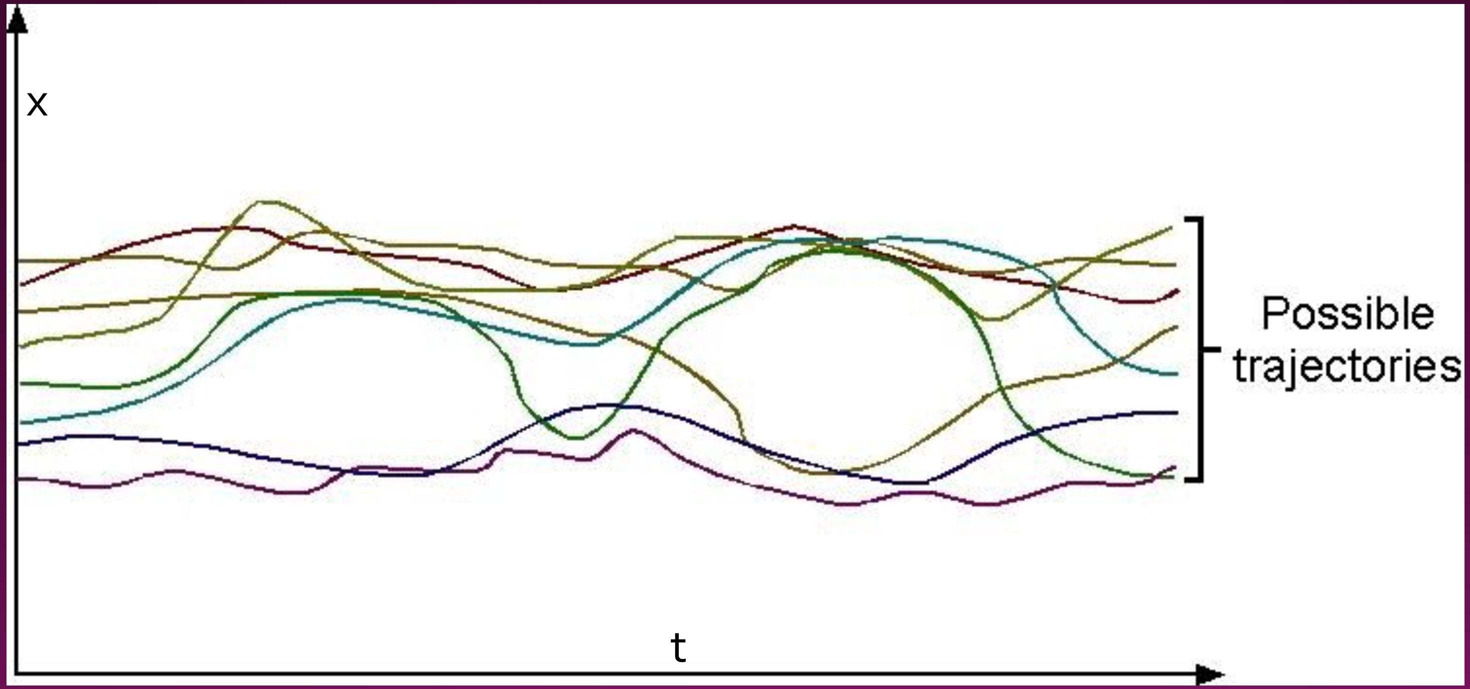
Motivation (for static analysis)

Say you've written code that you *really* don't want bugs in...

....like the controls for some rocket boots.

Motivation

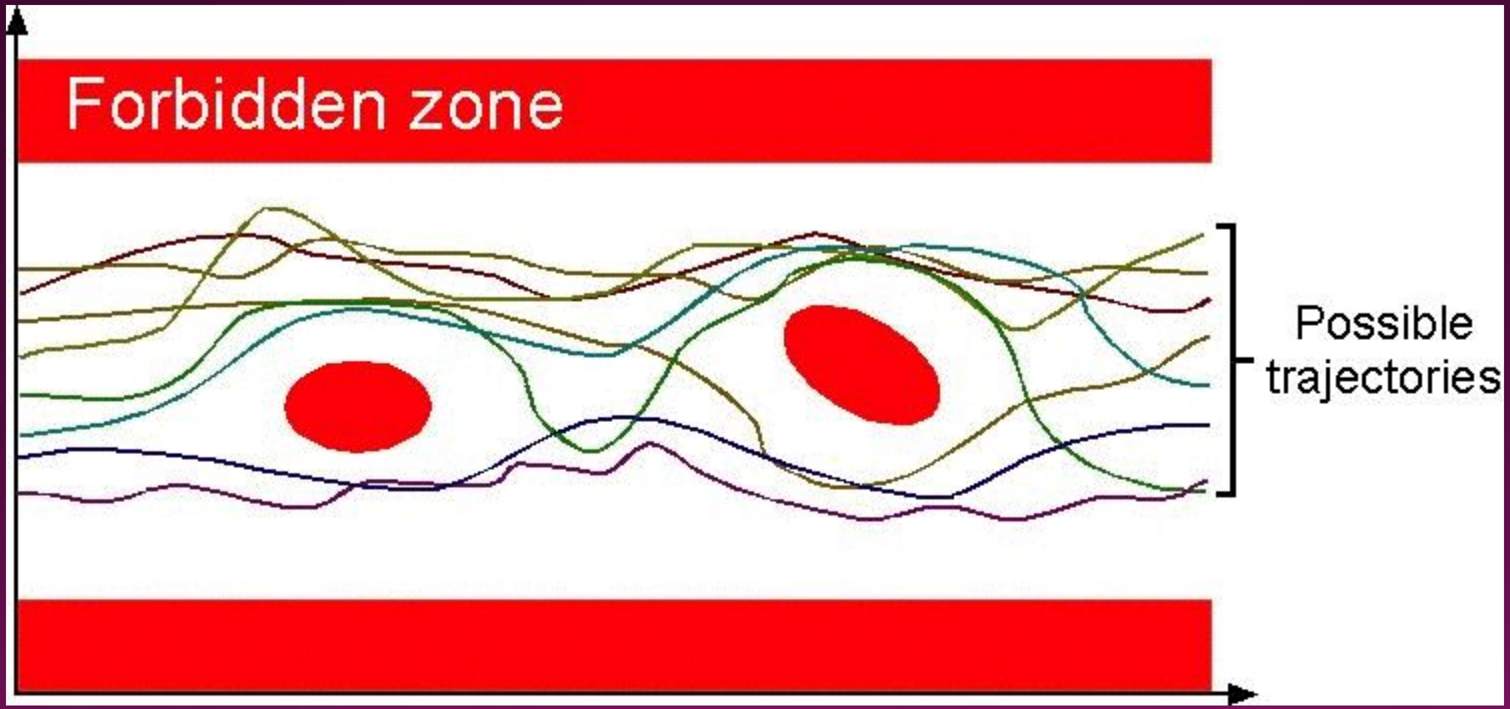
You want to reason about



Note: These sketches, and the intuition behind them, are from Patrick Cousot's website!

Motivation

To make sure you're safe



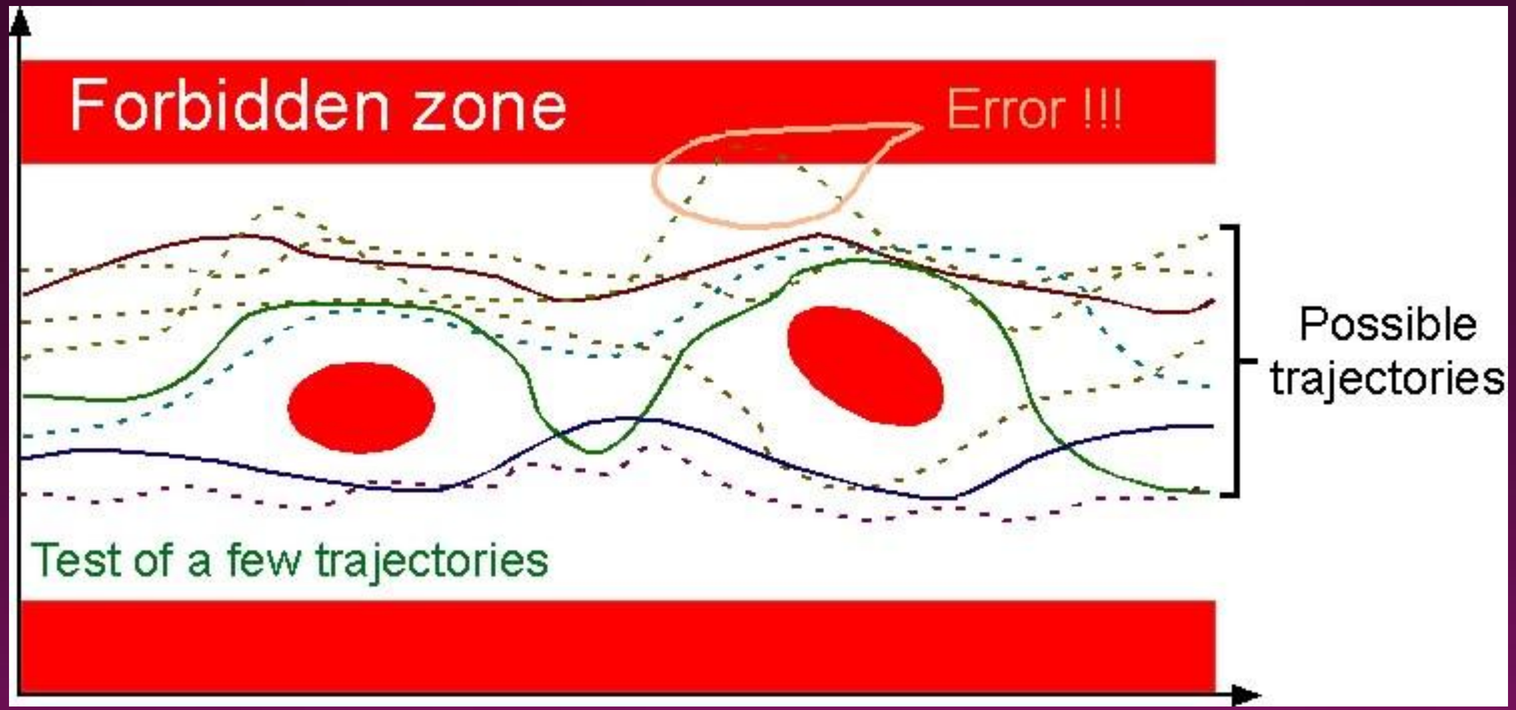
Motivation

....but you can't analyze code perfectly

Halting Problem

Motivation

Testing is dangerous…

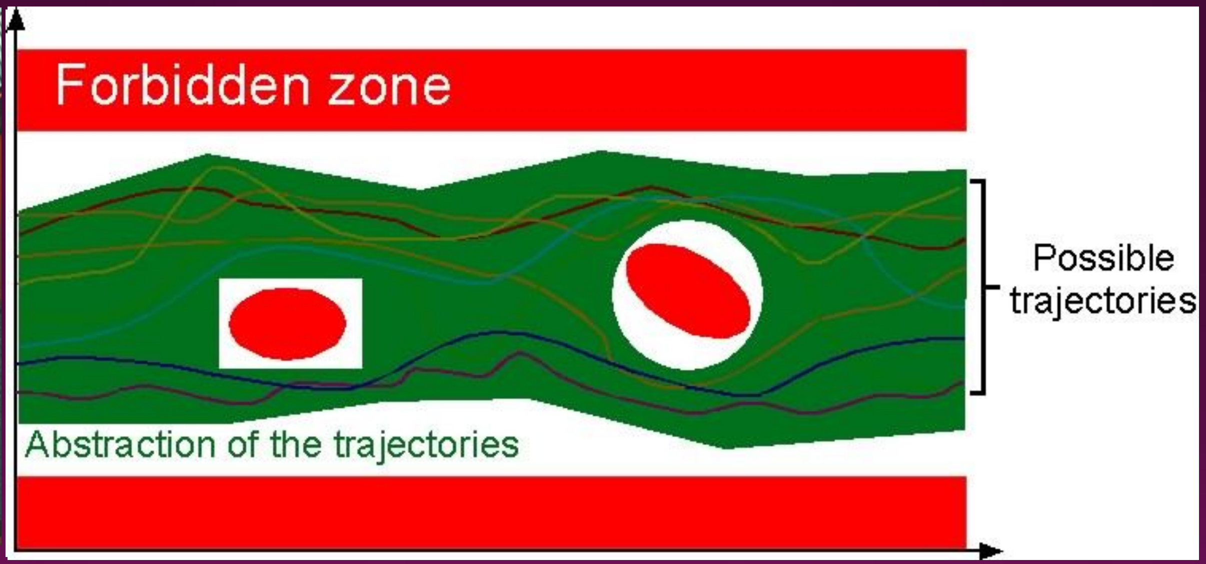


Motivation

Luckily you have an ally…

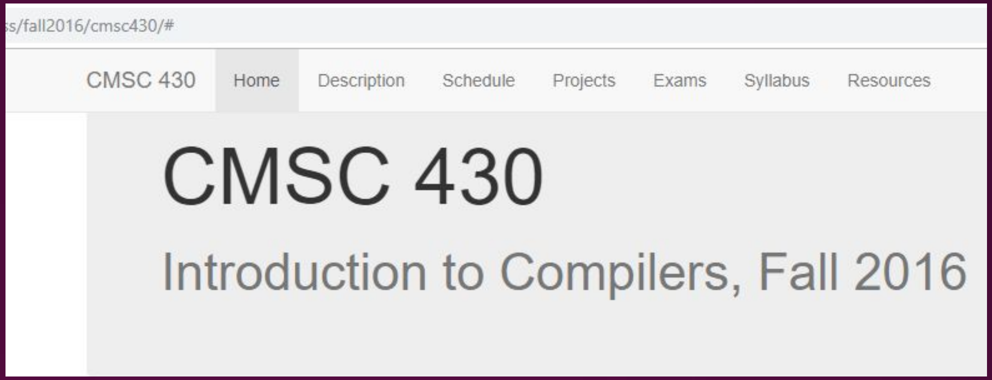
Motivation

With the power of… Abs



History – before this paper

- Early 70s work in data flow, type systems, etc
- As well as mathematical semantics



This paper

Uses mathematical semantics to give a grand unified theory of static analysis

Trivia:

Based on authors' work in interval analysis

Initially a 100 page handwritten manuscript submitted to the 4th POPL

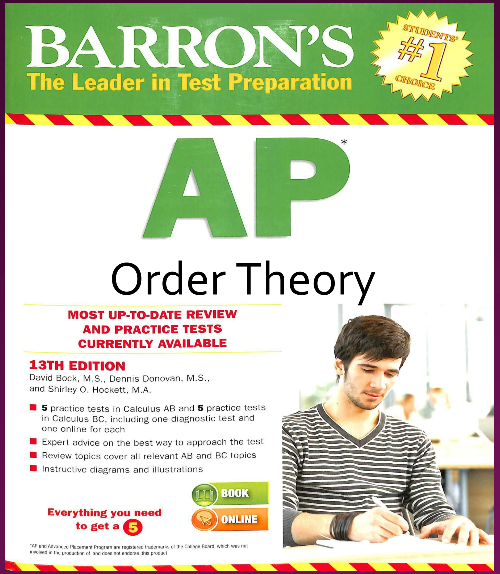
After this paper

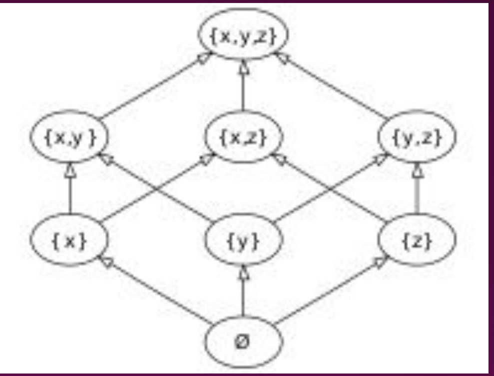
- Rich literature on static analysis in just about any domain you want
- Further theoretical exploration of AI

- Future, more computer-aided design.

Some Definitions

- A **lattice** is a partial order < L, ≤ > such that every two elements have a unique supremum (join) and infimum (meet)
- A **complete lattice** has a unique join and meet for every non-empty subset of L
- A **semi-lattice** only has join (or meet)





(from Wikipedia)

Abstraction

Properties of α, γ?

- Say we have a abstract, c concrete corresponding
- Somehow want maps to pick the "best"
  - $\alpha(c) \leq a,\ c \leq \gamma(a)$
- Monotone
  - $p \leq q \Rightarrow \alpha(p) \leq \alpha(q)$
- $\forall x, x = \alpha(\gamma(x))$
- $\forall y, y \leq \gamma(\alpha(y))$

Examples of Abstractions

- Sets of Integers
- (unbounded) Intervals
- Congruence mod 2
- One value or Sign

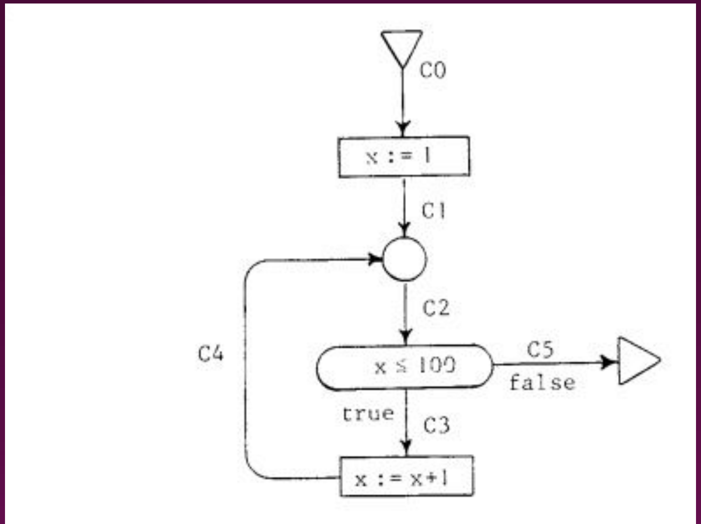

YO DAWG, I HEARD YOU LIKE LATTICES
SO I PUT YOUR LATTICES IN A LATTICE
SO YOU CAN LATTICE WHILE YOU LATTICE
imgflip.com

Interpretation

How do we actually use this?

In this case

- Flowchart language
- Context-collecting semantics (cv)
- Local Interpretation Int(r,cv)
- Global Interpretation G-Int(cv)
- cv = G-Int(cv)
- Least fixed point
- Iterate G-Int(bot) to solve



Abstract Interpretation

An abstract interpretation I of a program P is a tuple

$$I = \langle \text{A-Cont}, \circ, \leq, \top, \bot, \underline{\text{Int}} \rangle$$

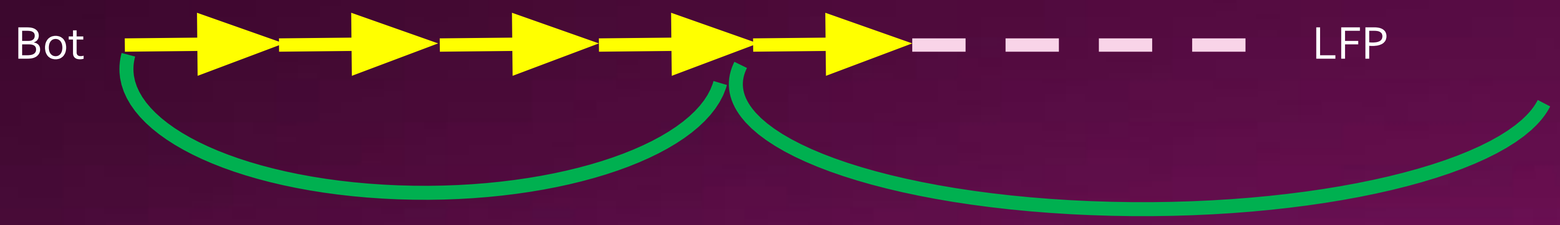
$$\{\forall (a, \bar{x}) \in \text{Arcs} \times \widetilde{\overline{\text{A-Cont}}},\ \gamma(\overline{\underline{\text{Int}}}(a, \bar{x})) \geq \underline{\text{Int}}(a, \tilde{\gamma}(\bar{x}))\}$$

and

$$\{\forall (a, x) \in \text{Arcs} \times \widetilde{\text{C-Cont}},\ \overline{\underline{\text{Int}}}(a, \tilde{\alpha}(x)) \mathrel{\bar{\geq}} \alpha(\underline{\text{Int}}(a, x))\}$$

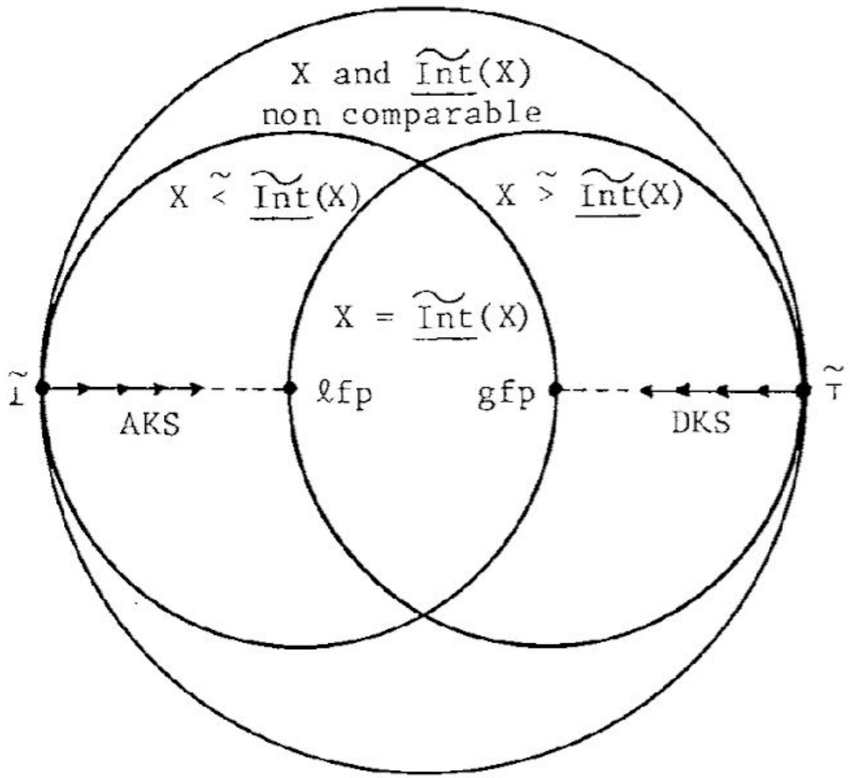
Widening

- So, we're done, right?
- No!
- We could be walking an infinite path



- Instead – jump! With over-approximations

The lattice $\widetilde{A\text{-Cont}}$ may be partitionned as follows :



Widening

Let $\widetilde{\text{A-int}} : \widetilde{\text{A-Cont}} \to \widetilde{\text{A-Cont}}$ be such that :

9.1.1.1 $\{\forall n \geq 0,\ C = \widetilde{\text{A-int}}^{n}(\tilde{\bot})$ and not$(\widetilde{\text{Int}}(C) \;\tilde{\leq}\; C)\}$
$\Longrightarrow \{C \;\tilde{\circ}\; \widetilde{\text{Int}}(C) \;\tilde{\leq}\; \widetilde{\text{A-int}}(C)\}$.

9.1.1.2 Every infinite sequence $\tilde{\bot}, \widetilde{\text{A-int}}(\tilde{\bot}), \ldots, \widetilde{\text{A-int}}^{n}(\tilde{\bot}), \ldots$ is not strictly increasing.

- The binary operation ∇ called widening defined by :

  9.1.3.1 ∇ : A-Cont × A-Cont → A-Cont

  9.1.3.2 $\forall (C, C') \in \text{A-Cont}^2,\ C \circ C' \leq C \nabla C'$

  9.1.3.3 Every infinite sequence $s_0, \ldots, s_n, \ldots$ of the form $s_0 = C_0, \ldots, s_n = s_{n-1} \nabla C_n, \ldots$ (where $C_0, \ldots, C_n, \ldots$ are arbitrary abstract contexts) is not strictly increasing.

```
A-int = λ(q, Cv) . if q ∈ W-arcs then
                      Cv(q) ∇ Int(q, Cv)
                   else
                      Int(q, Cv)
                   fi
```

Narrowing

- We might jump way too far
- Walk it back!

$$S_m \;\tilde{\geq}\; \underline{\widetilde{Int}}(S_m) \;\tilde{\geq}\; \ldots \;\tilde{\geq}\; \underline{\widetilde{Int}}^{n}(S_m) \;\tilde{\geq}\; \ldots \;\tilde{>}\; \underline{\overline{Cv}}.$$

- Again, this may be an (infinitely) long walk

Narrowing

Let $\widetilde{\underline{\text{D-int}}} : \widetilde{\text{A-Cont}} \rightarrow \widetilde{\text{A-Cont}}$ be such that :

9.3.2.1 $\{\forall C \in \widetilde{\text{A-Cont}}\}$
$\{C \tilde{\geq} \widetilde{\underline{\text{Int}}}(C)\} \Longrightarrow \{C \tilde{\geq} \widetilde{\underline{\text{D-int}}}(C) \tilde{\geq} \widetilde{\underline{\text{Int}}}(C)\}$

9.3.2.2 $\forall C \in \widetilde{\text{A-Cont}}$, every infinite sequence C, $\widetilde{\underline{\text{D-int}}}(C)$, ..., $\widetilde{\underline{\text{D-int}}}^n(C)$, ... is not strictly decreasing.

9.3.4.1 Δ : A-Cont $\times$ A-Cont $\rightarrow$ A-Cont

9.3.4.2 $\forall (C, C') \in \text{A-Cont}^2$,
$\{C \geq C'\} \Longrightarrow \{C \geq C \,\Delta\, C' \geq C'\}$

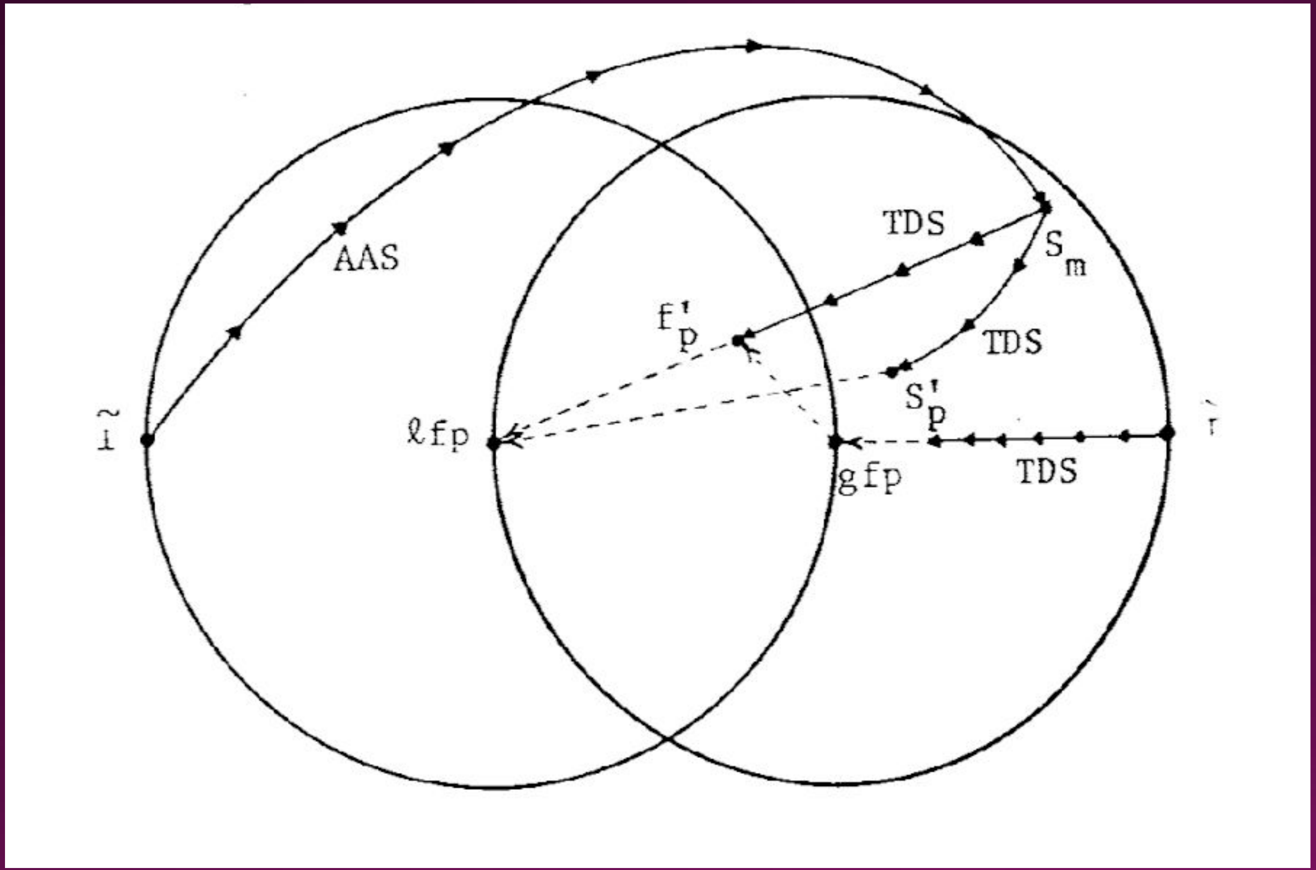
9.3.4.3 Every infinite sequence s_0, ..., s_n, ... of the form $s_0 = C_0$, $s_1 = s_0 \,\Delta\, C_1$, ..., $s_n = s_{n-1} \,\Delta\, C_n$, ... for arbitrary abstract contexts C_0, C_1, ..., C_n, ... is not strictly decreasing.

The approximated interpretation $\underline{\text{D-int}}$: $\text{Arcs}^0 \times \widetilde{\text{A-Cont}} \rightarrow$ A-Cont is defined by :

9.3.4.4 $\underline{\text{D-int}} = \lambda(q, \underline{\text{Cv}})$. $\underline{\text{if}}$ q $\in$ W-arcs $\underline{\text{then}}$
$\underline{\text{Cv}}(q) \,\Delta\, \underline{\text{Int}}(q, \underline{\text{Cv}})$
$\underline{\text{else}}$
$\underline{\text{Int}}(q, \underline{\text{Cv}})$
$\underline{\text{fi}}$

and

$\widetilde{\underline{\text{D-int}}} = \lambda \underline{\text{Cv}} . (\lambda q . \underline{\text{D-int}}(q, \underline{\text{Cv}}))$



AAS
TDS
S_m
f'_p
TDS
S'_p
$\tilde{i}$
ℓfp
gfp
TDS