Principles of Program Analysis:

Abstract Interpretation

Transparencies based on Chapter 4 of the book: Flemming Nielson, Hanne Riis Nielson and Chris Hankin: Principles of Program Analysis. Springer Verlag 2005.

A Mundane Approach to Semantic Correctness

Semantics:

$$p \vdash v_1 \leadsto v_2$$

where $v_1, v_2 \in V$.

Note: $\leadsto$ might be deterministic.

Program analysis:

$$p \vdash l_1 \rhd l_2$$

where $l_1, l_2 \in L$.

Note: $\rhd$ should be deterministic:

$$f_p(l_1) = l_2.$$

What is the relationship between the semantics and the analysis?

Restrict attention to analyses where properties directly describe sets of values i.e. *"first-order"" analyses* (rather than *"second-order" analyses*).

Example: Data Flow Analysis

Structural Operational Semantics:

Values: $V = \mathbf{State}$

Transitions:

$$S_\star \vdash \sigma_1 \rightsquigarrow \sigma_2$$

iff

$$\langle S_\star, \sigma_1 \rangle \to^* \sigma_2$$

Constant Propagation Analysis:

Properties: $L = \widehat{\mathbf{State}}_{\mathsf{CP}} = (\mathbf{Var}_\star \to \mathbf{Z}^\top)_\bot$

Transitions:

$$S_\star \vdash \widehat{\sigma}_1 \rhd \widehat{\sigma}_2$$

iff

$$\widehat{\sigma}_1 = \iota$$
$$\widehat{\sigma}_2 = \bigsqcup\{\mathsf{CP}_\bullet(\ell) \mid \ell \in \mathit{final}(S_\star)\}$$
$$(\mathsf{CP}_\circ, \mathsf{CP}_\bullet) \models \mathsf{CP}^=(S_\star)$$

Correctness Relations

$$R : V \times L \to \{\textit{true}, \textit{false}\}$$

Idea: $v \; R \; l$ means that the value v is described by the property l.

Correctness criterion: R is preserved under computation:

$$\begin{array}{ccccc}
p & \vdash & v_1 & \rightsquigarrow & v_2 \\
 & & \vdots & & \vdots \\
 & & R & \Rightarrow & R \\
 & & \vdots & & \vdots \\
p & \vdash & l_1 & \rhd & l_2
\end{array}$$

logical relation:

$$(p \vdash \cdot \rightsquigarrow \cdot) \; (R \twoheadrightarrow R) \; (p \vdash \cdot \rhd \cdot)$$

Admissible Correctness Relations

$$v \; R \; l_1 \;\wedge\; l_1 \sqsubseteq l_2 \;\Rightarrow\; v \; R \; l_2$$

$$(\forall l \in L' \subseteq L : v \; R \; l) \;\Rightarrow\; v \; R \; (\bigsqcap L')$$ ($\{l \mid v \; R \; l\}$ is a Moore family)

Two consequences:

$$v \; R \; \top$$

$$v \; R \; l_1 \;\wedge\; v \; R \; l_2 \;\Rightarrow\; v \; R \; (l_1 \sqcap l_2)$$

Assumption: $(L, \sqsubseteq)$ is a complete lattice.

Example: Data Flow Analysis

Correctness relation

$$R_{\mathsf{CP}} : \mathbf{State} \times \widehat{\mathbf{State}}_{\mathsf{CP}} \rightarrow \{\mathit{true}, \mathit{false}\}$$

is defined by

$$\sigma \; R_{\mathsf{CP}} \; \widehat{\sigma} \quad \text{iff} \quad \forall x \in \mathit{FV}(S_\star) : (\widehat{\sigma}(x) = \top \;\vee\; \sigma(x) = \widehat{\sigma}(x))$$

Representation Functions

$$\beta : V \rightarrow L$$

Idea: β maps a value to the *best* property describing it.

Correctness criterion:

$$\begin{array}{ccccc}
p & \vdash & v_1 & \rightsquigarrow & v_2 \\
 & \beta & \downarrow & & \downarrow & \beta \\
 & & & \Rightarrow & & \\
 & & \sqsubseteq & & \sqsubseteq \\
p & \vdash & l_1 & \rhd & l_2
\end{array}$$

Equivalence of Correctness Criteria

Given a representation function β we define a correctness relation R_β by

$$v \; R_\beta \; l \quad \text{iff} \quad \beta(v) \sqsubseteq l$$

Given a correctness relation R we define a representation function β_R by

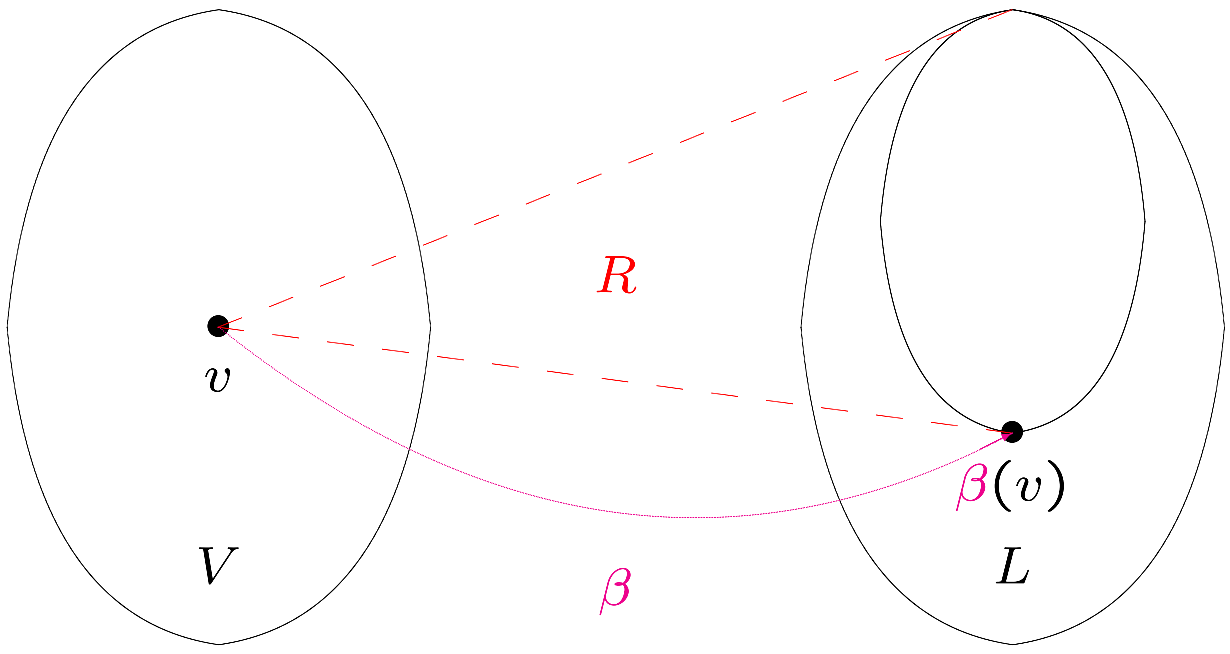
$$\beta_R(v) \;=\; \bigsqcap\{l \mid v \; R \; l\}$$

Lemma:

(i) Given $\beta : V \to L$, then the relation $R_\beta : V \times L \to \{\mathit{true}, \mathit{false}\}$ is an admissible correctness relation such that $\beta_{R_\beta} = \beta$.

(ii) Given an admissible correctness relation $R : V \times L \to \{\mathit{true}, \mathit{false}\}$, then β_R is well-defined and $R_{\beta_R} = R$.

Equivalence of Criteria: R is *generated* by β



Example: Data Flow Analysis

Representation function

$$\beta_{\mathsf{CP}} : \mathbf{State} \rightarrow \widehat{\mathbf{State}}_{\mathsf{CP}}$$

is defined by

$$\beta_{\mathsf{CP}}(\sigma) = \lambda x.\sigma(x)$$

R_{CP} is *generated by* β_{CP}:

$$\sigma \; R_{\mathsf{CP}} \; \widehat{\sigma} \quad \underline{\text{iff}} \quad \beta_{\mathsf{CP}}(\sigma) \sqsubseteq_{\mathsf{CP}} \widehat{\sigma}$$

A Modest Generalisation

Semantics:

$$p \vdash v_1 \rightsquigarrow v_2$$

where $v_1 \in V_1, v_2 \in V_2$

Program analysis:

$$p \vdash l_1 \rhd l_2$$

where $l_1 \in L_1, l_2 \in L_2$

$$\begin{array}{ccccc} p & \vdash & v_1 & \rightsquigarrow & v_2 \\ & & \vdots & & \vdots \\ & & R_1 & \Rightarrow & R_2 \\ & & \vdots & & \vdots \\ p & \vdash & l_1 & \rhd & l_2 \end{array}$$

logical relation:

$$(p \vdash \cdot \rightsquigarrow \cdot)\ (R_1 \twoheadrightarrow R_2)\ (p \vdash \cdot \rhd \cdot)$$

Approximation of Fixed Points

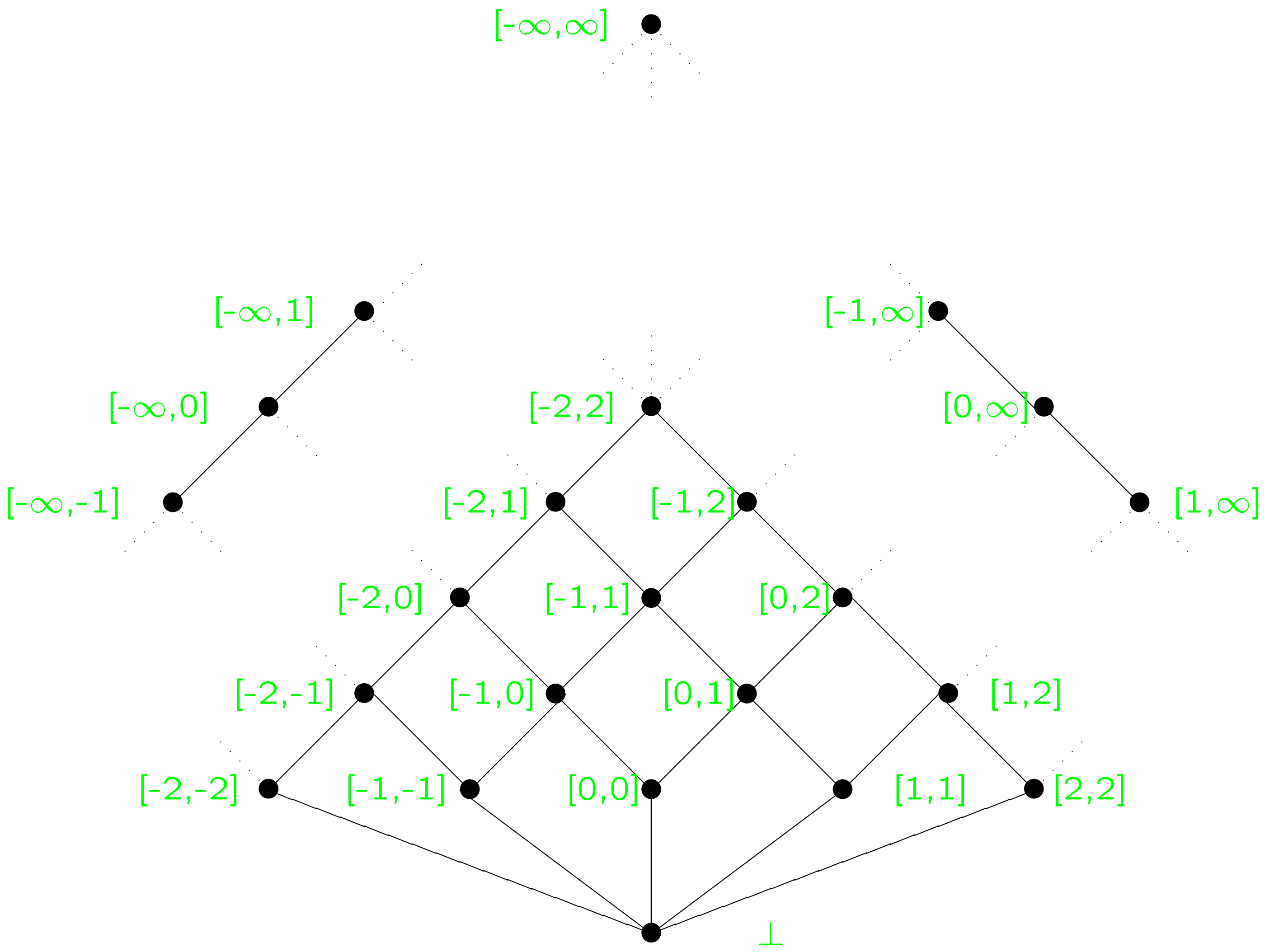
- Fixed points

- Widening

- Narrowing

Example: lattice of intervals for *Array Bound Analysis*

The complete lattice **Interval** = (**Interval**, $\sqsubseteq$)



Fixed points

Let $f : L \to L$ be a *monotone function* on a complete lattice $L = (L, \sqsubseteq, \sqcup, \sqcap, \bot, \top)$.

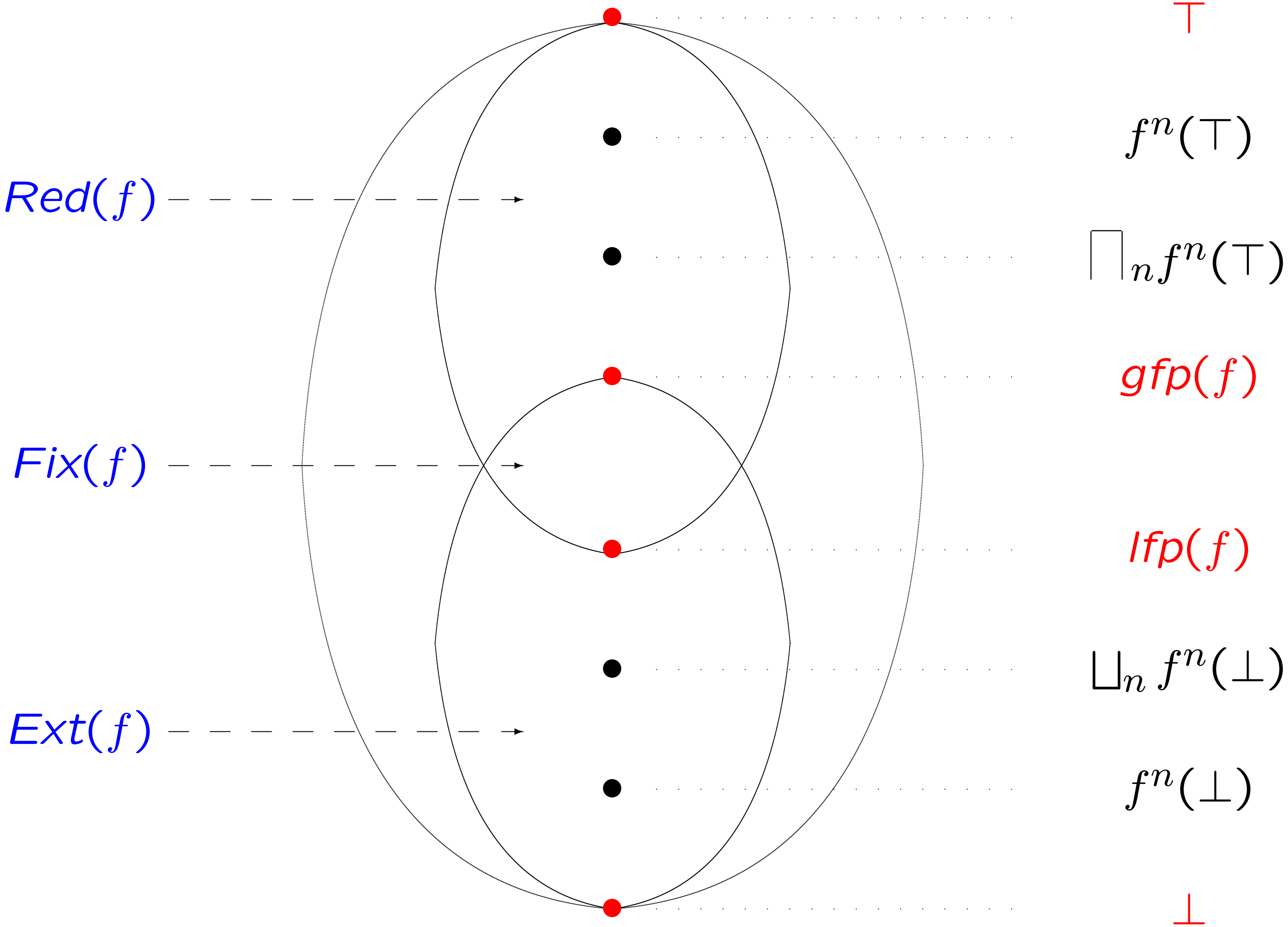
l is a *fixed point*	iff	$f(l) = l$	$Fix(f) = \{l \mid f(l) = l\}$
f is *reductive* at l	iff	$f(l) \sqsubseteq l$	$Red(f) = \{l \mid f(l) \sqsubseteq l\}$
f is *extensive* at l	iff	$f(l) \sqsupseteq l$	$Ext(f) = \{l \mid f(l) \sqsupseteq l\}$

Tarski's Theorem ensures that

$$lfp(f) = \bigsqcap Fix(f) = \bigsqcap Red(f) \in Fix(f) \subseteq Red(f)$$

$$gfp(f) = \bigsqcup Fix(f) = \bigsqcup Ext(f) \in Fix(f) \subseteq Ext(f)$$

Fixed points of f



Widening Operators

Problem: We cannot guarantee that $(f^n(\bot))_n$ eventually stabilises nor that its least upper bound necessarily equals *lfp*(f).

Idea: We replace $(f^n(\bot))_n$ by a new sequence $(f^n_\nabla)_n$ that is known to eventually stabilise and to do so with a value that is a safe (upper) approximation of the least fixed point.

The new sequence is parameterised on the widening operator ∇: an upper bound operator satisfying a finiteness condition.

Upper bound operators

$\breve{\sqcup} : L \times L \to L$ is an *upper bound operator* iff

$$l_1 \sqsubseteq l_1 \breve{\sqcup} l_2 \sqsupseteq l_2$$

for all $l_1, l_2 \in L$.

Let $(l_n)_n$ be a sequence of elements of L. Define the sequence $(l_n^{\breve{\sqcup}})_n$ by:

$$l_n^{\breve{\sqcup}} = \begin{cases} l_n & \text{if } n = 0 \\ l_{n-1}^{\breve{\sqcup}} \breve{\sqcup} l_n & \text{if } n > 0 \end{cases}$$

Fact: If $(l_n)_n$ is a sequence and $\breve{\sqcup}$ is an upper bound operator then $(l_n^{\breve{\sqcup}})_n$ is an ascending chain; furthermore $l_n^{\breve{\sqcup}} \sqsupseteq \bigsqcup\{l_0, l_1, \cdots, l_n\}$ for all n.

Example:

Let *int* be an arbitrary but fixed element of **Interval**.

An upper bound operator:

$$int_1 \,\check{\sqcup}^{int}\, int_2 = \begin{cases} int_1 \sqcup int_2 & \text{if } int_1 \sqsubseteq int \;\vee\; int_2 \sqsubseteq int_1 \\ [-\infty,\infty] & \text{otherwise} \end{cases}$$

Example: $[1,2]\check{\sqcup}^{[0,2]}[2,3] = [1,3]$ and $[2,3]\check{\sqcup}^{[0,2]}[1,2] = [-\infty,\infty]$.

Transformation of:	$[0,0],[1,1],[2,2],[3,3],$ $[4,4]$ $,[5,5],\cdots$
If $int = [0,\infty]$:	$[0,0],[0,1],[0,2],[0,3],$ $[0,4]$ $,[0,5],\cdots$
If $int = [0,2]$:	$[0,0],[0,1],[0,2],[0,3],$ $[-\infty,\infty]$ $,[-\infty,\infty],\cdots$

Widening operators

An operator $\nabla : L \times L \rightarrow L$ is a *widening operator* iff

- it is an upper bound operator, and
- for all ascending chains $(l_n)_n$ the ascending chain $(l_n^{\nabla})_n$ eventually stabilises.

Widening operators

Given a monotone function $f : L \to L$ and a widening operator ∇ define the sequence $(f_\nabla^n)_n$ by

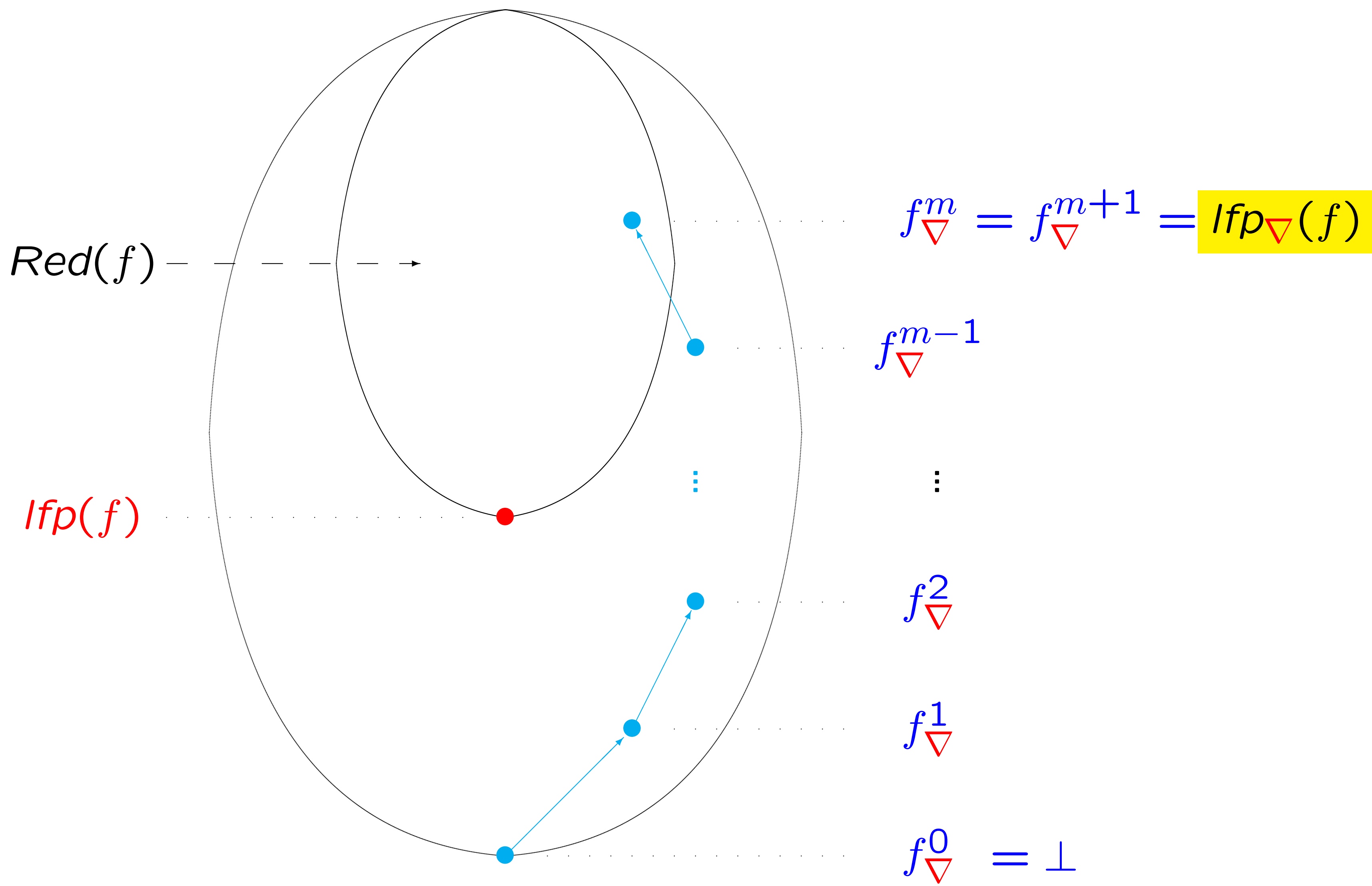
$$f_\nabla^n = \begin{cases} \bot & \text{if } n = 0 \\ f_\nabla^{n-1} & \text{if } n > 0 \ \wedge \ f(f_\nabla^{n-1}) \sqsubseteq f_\nabla^{n-1} \\ f_\nabla^{n-1} \ \nabla \ f(f_\nabla^{n-1}) & \text{otherwise} \end{cases}$$

One can show that:

- $(f_\nabla^n)_n$ is an ascending chain that eventually stabilises
- it happens when $f(f_\nabla^m) \sqsubseteq f_\nabla^m$ for some value of m
- Tarski's Theorem then gives $f_\nabla^m \sqsupseteq \mathit{lfp}(f)$

$$\mathit{lfp}_\nabla(f) = f_\nabla^m$$

The widening operator ∇ applied to f



Example:

Let K be a *finite* set of integers, e.g. the set of integers explicitly mentioned in a given program.

We shall define a widening operator ∇ based on K.

Idea: $[z_1, z_2] \;\nabla\; [z_3, z_4]$ is

$$[\ \mathsf{LB}(z_1, z_3)\ ,\ \mathsf{UB}(z_2, z_4)\ ]$$

where

- $\mathsf{LB}(z_1, z_3) \in \{z_1\} \cup K \cup \{-\infty\}$ is the best possible lower bound, and
- $\mathsf{UB}(z_2, z_4) \in \{z_2\} \cup K \cup \{\infty\}$ is the best possible upper bound.

The effect: a change in any of the bounds of the interval $[z_1, z_2]$ can only take place finitely many times – corresponding to the cardinality of K.

Example (cont.) — formalisation:

Let $z_i \in \mathbf{Z}' = \mathbf{Z} \cup \{-\infty, \infty\}$ and write:

$$\mathsf{LB}_K(z_1, z_3) = \begin{cases} z_1 & \text{if } z_1 \leq z_3 \\ k & \text{if } z_3 < z_1 \ \wedge \ k = \max\{k \in K \mid k \leq z_3\} \\ -\infty & \text{if } z_3 < z_1 \ \wedge \ \forall k \in K : z_3 < k \end{cases}$$

$$\mathsf{UB}_K(z_2, z_4) = \begin{cases} z_2 & \text{if } z_4 \leq z_2 \\ k & \text{if } z_2 < z_4 \ \wedge \ k = \min\{k \in K \mid z_4 \leq k\} \\ \infty & \text{if } z_2 < z_4 \ \wedge \ \forall k \in K : k < z_4 \end{cases}$$

$$int_1 \ \nabla \ int_2 = \begin{cases} \perp & \text{if } int_1 = int_2 = \perp \\ [\ \mathsf{LB}_K(\inf(int_1), \inf(int_2))\ ,\ \mathsf{UB}_K(\sup(int_1), \sup(int_2))\ ] & \text{otherwise} \end{cases}$$

Example (cont.):

Consider the ascending chain $(int_n)_n$

$$[0,1],[0,2],[0,3],[0,4],[0,5],[0,6],[0,7],\cdots$$

and assume that $K = \{3,5\}$.

Then $(int_n^{\nabla})_n$ is the chain

$$[0,1],[0,3],[0,3],[0,5],[0,5],[0,\infty],[0,\infty],\cdots$$

which eventually stabilises.

Narrowing Operators

Status: Widening gives us an upper approximation $\mathit{lfp}_\nabla(f)$ of the least fixed point of f.

Observation: $f(\mathit{lfp}_\nabla(f)) \sqsubseteq \mathit{lfp}_\nabla(f)$ so the approximation can be improved by considering the iterative sequence $(f^n(\mathit{lfp}_\nabla(f)))_n$.

It will satisfy $f^n(\mathit{lfp}_\nabla(f)) \sqsupseteq \mathit{lfp}(f)$ for all n so we can stop at an arbitrary point.

The notion of narrowing is *one way* of encapsulating a termination criterion for the sequence.

Narrowing

An operator $\triangle : L \times L \to L$ is a *narrowing operator* iff

- $l_2 \sqsubseteq l_1 \;\Rightarrow\; l_2 \sqsubseteq (l_1 \triangle l_2) \sqsubseteq l_1$ for all $l_1, l_2 \in L$, and
- for all descending chains $(l_n)_n$ the sequence $(l_n^{\triangle})_n$ eventually stabilises.

Recall: The sequence $(l_n^{\triangle})_n$ is defined by:

$$l_n^{\triangle} = \begin{cases} l_n & \text{if } n = 0 \\ l_{n-1}^{\triangle} \triangle l_n & \text{if } n > 0 \end{cases}$$

Narrowing

We construct the sequence $([f]^n_\triangle)_n$

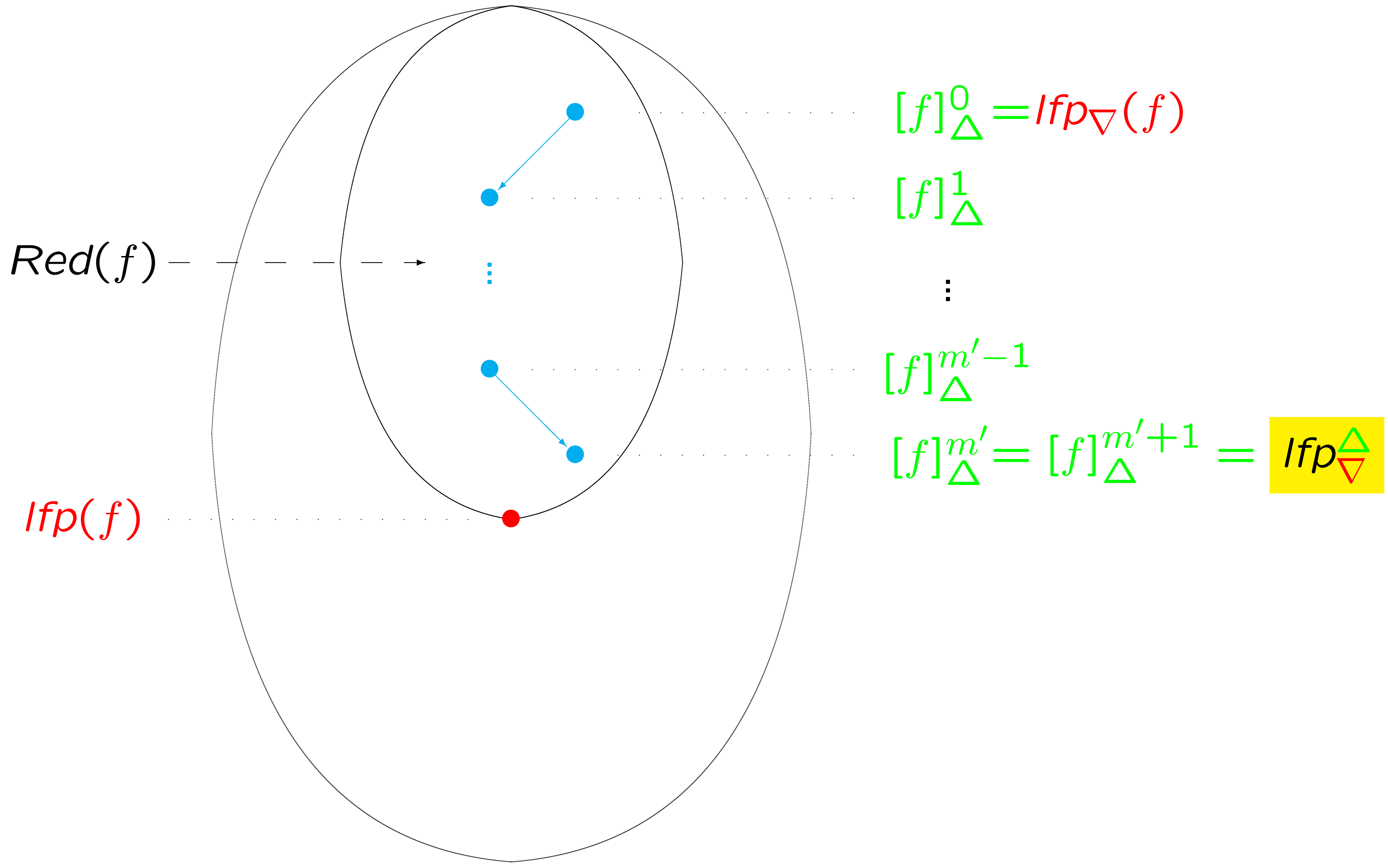
$$[f]^n_\triangle = \begin{cases} \mathit{lfp}_\nabla(f) & \text{if } n = 0 \\ [f]^{n-1}_\triangle \;\triangle\; f([f]^{n-1}_\triangle) & \text{if } n > 0 \end{cases}$$

One can show that:

- $([f]^n_\triangle)_n$ is a descending chain where all elements satisfy $\mathit{lfp}(f) \sqsubseteq [f]^n_\triangle$
- the chain eventually stabilises so $[f]^{m'}_\triangle = [f]^{m'+1}_\triangle$ for some value m'

$$\mathit{lfp}^{\triangle}_{\nabla}(f) = [f]^{m'}_\triangle$$

The narrowing operator Δ applied to f



Example:

The complete lattice (**Interval**, $\sqsubseteq$) has two kinds of infinite descending chains:

- those with elements of the form $[-\infty, z]$, $z \in \mathbf{Z}$
- those with elements of the form $[z, \infty]$, $z \in \mathbf{Z}$

Idea: Given some fixed non-negative number N
the narrowing operator $\triangle_N$ will force an infinite descending chain

$$[z_1, \infty], [z_2, \infty], [z_3, \infty], \cdots$$

(where $z_1 < z_2 < z_3 < \cdots$) to stabilise when $z_i > N$

Similarly, for a descending chain with elements of the form $[-\infty, z_i]$ the narrowing operator will force it to stabilise when $z_i < -N$

Example (cont.) — formalisation:

Define $\triangle = \triangle_N$ by

$$int_1 \triangle int_2 = \begin{cases} \bot & \text{if } int_1 = \bot \ \vee \ int_2 = \bot \\ [z_1, z_2] & \text{otherwise} \end{cases}$$

where

$$z_1 = \begin{cases} \inf(int_1) & \text{if } N < \inf(int_2) \wedge \sup(int_2) = \infty \\ \inf(int_2) & \text{otherwise} \end{cases}$$

$$z_2 = \begin{cases} \sup(int_1) & \text{if } \inf(int_2) = -\infty \wedge \sup(int_2) < -N \\ \sup(int_2) & \text{otherwise} \end{cases}$$

Example (cont.):

Consider the infinite descending chain $([n, \infty])_n$

$$[0, \infty], [1, \infty], [2, \infty], [3, \infty], [4, \infty], [5, \infty], \cdots$$

and assume that $N = 3$.

Then the narrowing operator $\triangle_N$ will give the sequence $([n, \infty]^{\triangle})_n$

$$[0, \infty], [1, \infty], [2, \infty], [3, \infty], [3, \infty], [3, \infty], \cdots$$

