- from Galois connections to upper closure operators
- the lattice of abstract interpretations
- refinement operators
  - reduced product
  - disjunctive completion
  - completion by complements
  - reduced cardinal power
- logical interpretation of refinements
- Heyting's completion
From Galois Connections to Upper Closure Operators

Concrete domain \( C \subseteq \mathbb{E} \)

Abstract domain \( A \subseteq \mathbb{F} \)

- The composition of \( \sigma \) with \( \delta \) is an operator on \( C \)
  \[ \delta \circ \sigma : C \to C \]

Having the following properties:

- \( \mu(x') \) is a safe approximation of \( x \)
  \[ x \in \mu(x') \]

- \( \delta \) is monotonic

Composition of monotonic functions

- \( \delta \) is idempotent
  \[ \delta \circ \delta(x) = \delta(x) \]

- Due to properties of closure functions
  The approximation is obtained in one step

- \( \delta \) is an upper closure operator on \( C \)
Each closure operator \( g : C \to C \) is uniquely determined by the set of its fixpoints \( \text{Fix}(g) \) and its image \( g(C) \).

\( p(C) \) is a complete lattice with \( \leq \), where

\( X \subseteq C \) is the set of fixpoints \( p(C) \) of a closure operator \( g \) on \( C \) iff it is a Moore family:

- \( \overline{x} \in X \)
- \( X \) is downward closed

\( \overline{x} \) is the Moore-classe of \( x \),

the least subset of \( C \) which contains \( x \) and is a Moore family.
• every Galois insertion uniquely determines an upper closure operator

• given any upper closure operator \( f \) on \( C \)
  
  • there exist many abstract domains \( A \) such that \( f = \text{lift}_{\lambda} x \cdot \lambda x' (x)(x) \)
  
  • these abstract domains are "isomorphic" to \( f \)

  • since an abstract domain together with its abstraction and concretization functions defines an abstract interpretation

  • any upper closure operator on \( C \) defines an abstract interpretation

  • without an abstract domain, which just a representation 'implementation' of the property modeled by \( f \)

  • it makes easier reasoning on the relation among different abstract interpretations because they are all defined on the same domain

  • in order a representation (abstract domain and Galois insertion) is needed when designing (possibly optimal) abstract operations.
THE LATTICE OF ABSTRACT INTERPRETATIONS

- Any uco on $C$ is an abstract interpretation.

- The set of upper closure operators on $C$ has a natural partial order relation $\leq$.

  
  A function pointwise order based on $E$

  
  $f_1 : C \rightarrow C$

  $f_2 : C \rightarrow C$

  $f_1 \leq f_2$ iff $\forall x \in C, f_1(x) \subseteq f_2(x)$

- $(uco(C), \leq)$ is a complete lattice.

  The closure operator which maps every element of $C$ to the top element of $C$. 

  The "most precise" abstract interpretation.

  The identity function.

  $uco(C), \leq$
REFINEMENT OPERATORS

$\text{uco}(c), \leq$

- Refinement of abstract domains (upper closure operators)

\[ \bar{\text{uco}}(c) \rightarrow \text{uco}(c) \]

- Delivers a more precise abstract domain

\[ \forall a \in \text{uco}(c), \quad R\bar{a} \leq a \]

- Is monotonic

- Is idempotent

- By improvement in decision refinement is obtained all in one step

- Refinements are lower closure operators on $\text{uco}(c)$
SOME REFINEMENT OPERATORS

- **Reduced product**

  \[ A \Pi B \]

  Cartesian product of the two domains whose pairs having equivalent meaning (representing the same property) are identified (reduced).

- Given a domain \( A \in \text{uco}(c) \)

  \[ \lambda x. x \Pi A \text{ is clearly arown closure operator on } \text{uco}(c) \]

  \[ \downarrow \]

  it is a refinement operator

- The most abstract (simplest) domain which is more precise of the given domains which allow us to derive at least the same invariants.

- It is exactly the gcb in the lattice of uco's

- **Closed under intersection** which plays the role of conjunction of properties
AN EXAMPLE OF REDUCED PRODUCT

Concrete domain: $\mathbb{R}(Z)$,

\[ Z \]

\[ o^+ \]

\[ + \]

\[ Z \]

\[ o^- \]

\[ - \]

\[ A^+ \]

\[ 1 \]

\[ A^- \]

\[ 2 \]

\[ A^+ \cap A^- \]

- The two "new" points are the "intersections"
  - $\circ$ is the intersection between $o^-$ and $o^+$
  - $\phi$ is the intersection between $-$ and $+$

"Our friend Sigur!"
MORE REFINEMENT OPERATORS

- disjunctive completion $R_v$

$R_v : ucoc(c) \rightarrow ucoc(c)$

$R_v(a)$, $a \in ucoc(c)$ (abstract domain)

adds to $a$ denotations for concrete disjunctions of its values

- the most abstract domain which can represent concrete disjunctions

- to improve the precision of the domain in abstract computations with multiple branchings

- nonconfinement

- disjunction of properties (rather than tainting labels) on the original abstract domain
AN EXAMPLE OF DISJUNCTIVE COMPLETION

- Concrete domain

- Abstract domain

\[ \mathcal{P}(\mathbb{Z}) \]

\[ A = \]

\[ \mathcal{P}_w(A) = \]

- The "new" points

\[ 0^- \quad \text{disjunction of } - \quad \text{and} \quad 0 \]

\[ \pm 0 \quad \text{disjunction of } - \quad \text{and} \quad + \]

\[ 0^+ \quad \text{disjunction of } + \quad \text{and} \quad 0 \]

- \( \mathcal{P}_w(A) \) is also the disjunctive completion of Sign
More refinement operators

Completion by complements

$\mathcal{R}_{C_1} : \text{uo}(c) \to \text{uo}(c)$

$\mathcal{R}_{C_1}(a), \ a \in \text{uo}(c)$ (abstract cousin)

(when possible) upgrades $\mathcal{A}$ by adding (lattice-theoretic) complements of its elements
AN EXAMPLE OF COMPLETION BY COMPLEMENTS

- concrete domain

\[ P(\mathbb{Z}) \]

- abstract domains

\[ A_1 = \mathbb{Z} \]
\[ A_2 = \mathbb{Z} \]

- \( R_{\mathbb{A}}(A_1) = R_{\mathbb{A}}(A_2) = \)
MORE REFINEMENT OPERATORS

- Reduced cardinal power

\[ \mathcal{B}^A, \quad \mathcal{B}(\text{base}), A(\text{exponent}) \leq \text{uc} \circ c \]

the set of all monotonic functions

\[ A \xrightarrow{m} B \]  

Reduced with respect to cocontinuity

(we identify functions which represent the same property)

in the original definition (based on Galois connections)

\[ c = \mathcal{P}(x) \]

\[ d_A : A \rightarrow C \]
\[ d_B : B \rightarrow C \]
\[ f_A : C \rightarrow A \]
\[ f_B : C \rightarrow B \]

\[ \gamma = \lambda P. \lambda x. \, d_B (P \cup \gamma A(x)) \]  

for any \( P \)  

compute how \( P \) changes when put in conjunction with elements of the exponent abstract domain

- With closure operators

\[ A = \rho A(c) \]
\[ B = \rho B(c) \]

The function \( xx \in A, \rho B(d \land x) : A \rightarrow B \), \( d \in C \)
represents a dependency

- The reduced cardinal power is the set of all such dependencies

\[ \mathcal{B}^A = \{ \lambda x \in \mathcal{A}, \rho B (d \land x) | d \in \mathcal{C} \} \]
The idea is to model dependencies among properties defined by the two domains

- Relational analysis to improve the precision

- Remember properties such as

  \[ \text{"} X \text{ is ground if } Y \text{ is ground} \text{"} \]

  \[ \text{"} X \text{ and } Y \text{ uniquely do not share if } Z \text{ is free} \text{"} \]

- Difficult to see on sign-related abstractions in the present form

  We will show an "almost" equivalent formulation easier to handle
TOWARDS A LOGICAL INTERPRETATION OF REFINEMENTS

- if the concrete domain C is structured as \( P(X) \)
  - a property is modeled by
    - an abstract domain or
    - an upper closure operator

- Some refinement operators can be viewed as completions, which allow to model
  - property intersection (reduced product)
  - property disjunction (disjunctive completion)
  - negation of properties (completion by complements)

- What is the logical interpretation of reduced cardinal power?

  which, in principle, allows us to handle dependencies among properties and shared,
  therefore, be the basic component of (more accurate) relational analyses?
THE LOGICAL RECONSTRUCTION OF
A DOMAIN FOR SIGN ANALYSIS BY MEANS
OF REFINEMENTS

1. Basic properties

A₁ (the property of being "positive")
A₂ (the property of being "zero")
A₃ (the property of being "negative")

2. Refined domains

A₄ = A₁ ∩ A₃
A₅ = A₄ ∩ A₂
A₆ = P₀ₐ(A₅)
How to model dependencies in logic
(towards a logical characterization of reduced and dual power)

- The interesting logical operator is "implication"
  
  Given properties \( p \) and \( q \),
  
  \( p \rightarrow q \) tells us that whenever \( p \) holds, then \( q \) holds too.

- Enhance a given abstract domain of properties to include the space of all the above implications built from every pair of its elements.

- If \( a, b \) are elements of the abstract domain \( A \)

  The enhancement of \( A \) should contain relational objects (implications) \( a \rightarrow b \), with the following property

  \[ a \land (a \rightarrow b) \] is approximated by \( b \) (i.e., modus ponens)

  \[ a \land (a \rightarrow b) \leq b \]

  We have many choices for an element \( c \) representing the implication \( a \rightarrow b \)

  - A best choice exists if the complete lattice is a Heyting algebra,
    models of intuitionistic logic

  - The implication has to be understood as intuitionistic implication

  \[ a \rightarrow b \] means that a proof for \( a \)
  can be transformed into a proof for \( b \).
Heyting Completion

$C$: concrete domain

$a \rightarrow b = \text{lub}_c \{ d | \text{get}(a, d) = c \cdot b \}$

$A, B \subseteq C$ Moore families (set of fixpoints of two closure operations on $C$)

$A \rightarrow B = \{ a \rightarrow b | a \in A, b \in B \}$

This is not necessarily a Moore family

The Heyting completion

$A \Rightarrow B =$ the most abstract Moore family containing $A \rightarrow B$

The Moore completion of $A \rightarrow B$
An example

- = 8(Z), ≤

\[ A \xrightarrow{0} A \]

\[ 0 = + \]
\[ 0 = \phi \]

\[ (A \Rightarrow A) \Rightarrow (A \Rightarrow A) \]

In a Moore family

\[ Z \]

\[ 0+ \]

\[ 0- \]

\[ 0+ \]

\[ 0- \]

\[ Z \]

\[ 0+ \]

\[ 0- \]

\[ Z \]

\[ 0+ \]

\[ 0- \]
Heyting completion and refinements

- Reduced cardinal power

\[ B^A \cong A \xrightarrow{\Lambda} B \]

- Nice algebraic properties where the various refinements - compositions

  algebra of domain operators
Applications to Logic Programs

- Groundness

\[
\begin{align*}
DEF &= G \Rightarrow G \\
POS &= DEF \Rightarrow DEF \\
POS &= POS \Rightarrow POS
\end{align*}
\]

- (polymorphic) types

- Sharing 2 freeness

\[
\begin{align*}
NS &= \text{Simple non pair sharing} \\
F &= \text{freeness} \\
\text{a powerful and precise new domain}
\end{align*}
\]

\[
(\text{NSNF}) \Rightarrow (\text{NSNF})
\]