Taken from Lecture II of David Schmidts

Foundations of Abstract Interpretation
1. Lattices and continuous functions
2. Galois connections, closures, and Moore families
3. Soundness and completeness of operations on abstract data
4. Soundness and completeness of execution trace computation

Outline
Datasets are complete lattices. A complete lattice is a partially ordered set with unique minimal and maximal elements, and with greatest-lower-bound and least-upper-bound operations.

\[ \begin{align*}
  \text{none} &= \{ \} \\
  \text{all} &= \{ \} \\
  \text{all} &= \{ \text{notpos, notneg, neg, zero} \} \\
  \text{notpos, notneg} &= \{ \text{zero} \} \\
  \text{notpos, notneg} &= \{ \text{zero} \} \\
  \text{pos} &= \{ \text{none, all, pos} \} \\
  \text{zero} &= \{ \text{pos, zero} \} \\
  \text{neg} &= \{ \text{notpos, neg, neg} \} \\
  \text{all} &= \{ \text{pos, zero, neg} \} \\
  \text{top} &= \{ \text{all} \} \\
  \text{bottom} &= \{ \text{none} \}
\end{align*} \]
Here is a more precise definition: A complete lattice consists of a set, D, and a partial ordering, \( \leq \), on D such that D is not empty, and for all \( a, b \in D \):

- Every lower bound \( \sqcup \) of any subset \( S \subseteq D \) is also in D.
- Every upper bound \( \sqcap \) of any subset \( S \subseteq D \) is also in D.

A least upper bound operation \( \sqcup \), defined dually to the above:

\[ a \sqcup c \text{ for all } a, c \in S, \text{ and when } a \sqsubseteq c \text{ for all } a, c \in S, \text{ we have that} \]

A greatest lower bound operation \( \sqcap \), defined dually:

\[ a \sqcap c \text{ for all } a, c \in S, \text{ we have that} \]

A greatest element, \( \top \) (such that \( a \sqsubseteq \top \) for all \( a \in D \)) and a smallest element, \( \bot \) (such that \( \bot \sqsubseteq a \) for all \( a \in D \)) and a partial ordering, \( \leq \), on D.
The first example is the complete lattice \( \langle \text{Int}, \subseteq, \cap, \cup, \emptyset, \cap \rangle \); the next two are abstractions of it:

The first example is the complete lattice, the next two are abstractions of it:
Monotonic and chain-continuous functions

Given complete lattices, so it is no restriction to use only them.

A monotonic function preserves the "precision of information" in its argument.

\[
0 \leq a_i \implies 0 \leq a_i' \implies f(a_i) = f(a_i')
\]

A function, \( f : A \rightarrow B \), is \( w \)-continuous iff

Say that we have an \( w \)-chain, \( a_0 \sqsubseteq a_1 \sqsubseteq \ldots \sqsubseteq a_{i+1} \sqsubseteq a_i \).

A monotonic function preserves the "precision of information" in its argument.

For all \( a_i', a_i \in A \), \( a_i' \sqsubseteq a_i \implies f(a_i') \sqsubseteq f(a_i) \) monotonically.

Given complete lattices, \( A \) and \( B \), we say that a function, \( f : A \rightarrow B \), is \( w \)-continuous if

**Monotonic and chain-continuous functions**
Galois connections

Given a complete lattice of "concrete" (execution) data, \( \mathcal{C} \), and a simpler complete lattice of "abstract" data, \( \mathcal{A} \), we relate the two by given a complete lattice of "concrete" (execution) data, \( \mathcal{C} \), and a simpler complete lattice of "abstract" data, \( \mathcal{A} \), we relate the two by

\[ \alpha : \mathcal{C} \rightarrow \mathcal{A}, \quad \gamma : \mathcal{A} \rightarrow \mathcal{C} \]

It will be useful that \( \alpha \) have an "inverse";

\[ \forall a \in \mathcal{A}, \exists c \in \mathcal{C} : \alpha(c) = a \]

\[ \forall c \in \mathcal{C}, \exists a \in \mathcal{A} : \gamma(a) = c \]

written \( \langle \alpha, \gamma \rangle \) for the pair, \( \langle \alpha, \gamma \rangle \) form a Galois connection.

Definition: For complete lattices \( \mathcal{C} \) and \( \mathcal{A} \), and monotonic functions \( \alpha \) and \( \gamma \), such that \( \alpha \) act like a homomorphism when we study the operations on \( \mathcal{C} \).
The maps $\alpha$ and $\gamma$ are inverse maps on each other's image:

\[ \{ (a) \, \gamma \subseteq C \mid \forall a \, (\alpha(a) \subseteq a) \} \cap C = \{ c \mid \forall c \subseteq C \alpha(c) \subseteq a \} \]

That is, for all $c \in \gamma(C)$, $c = \gamma \circ \alpha(c)$, for all $c \in C$, $\alpha = \gamma \circ \alpha(c)$.

Moreover, $\alpha$ is $\omega$-continuous (and even preserves $\nabla$ for arbitrary sets in $C$); $\gamma$ preserves $\bot$ for arbitrary sets in $A$. Each map uniquely defines the other:

The maps $\alpha$ and $\gamma$ are inverse maps on each other's image.
The previous fact suggests this alternative characterization of Galois connection:

**Proposition:** For complete lattices $\mathcal{A}$ and $\mathcal{C}$, the pair $\mathcal{A} \leftarrow \mathcal{C}$ is a Galois connection when, for all $\mathcal{C} \in \mathcal{C}$ and $\mathcal{A} \in \mathcal{A}$,

$$\alpha \in \mathcal{A}, \quad \mathcal{C} \subseteq \mathcal{A} \iff \mathcal{A} \cup \mathcal{C} \subseteq \mathcal{A}.$$

From this definition, we can prove that both $\alpha$ and $\gamma$ are monotonic.

That $\mathcal{C} \subseteq \mathcal{C} \cap \mathcal{A}(\mathcal{C})$, and that $\alpha \circ \gamma(\mathcal{A}) \subseteq \mathcal{A}$. 

$$\exists \mathcal{A} \ni \mathcal{C} \subseteq \mathcal{C} \cap \mathcal{A}(\mathcal{C}) \iff \mathcal{A} \cup \mathcal{C} \subseteq \mathcal{A}.$$
In the other direction, assume \( \mathcal{C}, \mathcal{A} \) is an adjunction. We show that \( \mathcal{C}, \mathcal{A} \) is similar.

Proof: Assume that \( \mathcal{C}, \mathcal{A} \) is a Galois connection, and let \( a \subseteq \mathcal{C} \). Since \( \mathcal{C} \) is
monotone, we have \( \mathcal{A} \) \( \subseteq \mathcal{A} \subseteq \mathcal{A} \), using \( \mathcal{C} \mathcal{A} \subseteq \mathcal{A} \mathcal{C} \subseteq \mathcal{A} \mathcal{C} \). We get \( c \subseteq \mathcal{A} \mathcal{C} \subseteq \mathcal{A} \mathcal{C} \mathcal{A} \mathcal{C} \mathcal{A} \subseteq \mathcal{A} \mathcal{C} \mathcal{A} \mathcal{C} \). Since \( \mathcal{C} \) is

is called an adjunction if every \( a \in \mathcal{A} \) and \( a \in \mathcal{A} \) satisfy an alternative definition is given by the notion of adjunction. A structure \( \mathcal{C}, \mathcal{A} \) is a Galois connection if both

Recall that a structure \( \mathcal{C}, \mathcal{A} \) is a Galois connection if both

Galois Connection
Claim 2. The maps $\map{a}{y}$ and $\map{y}{a}$ are inverse maps on each other's images. That is, $\map{a}{y} \circ \map{y}{a} = y$ and $\map{y}{a} \circ \map{a}{y} = a$.

**Proof:** We have $x = \map{a}{\map{y}{x}}$, and since $x$ is monotone we get $x \subseteq \map{a}{\map{y}{(\map{y}{x})}}$. Similarly, $\map{y}{x} \subseteq x$. Thus $\map{a}{y} \circ \map{y}{a} = a$. The proof of $\map{y}{a} \circ \map{a}{y} = y$ is similar.
is completely multiplicative.

We can characterize these properties by saying that $\alpha$ is completely additive while

\[ \{ \alpha \} \subseteq \{ a \} \Leftrightarrow (\alpha)(\lambda) \subseteq \alpha \]

The proof that $\alpha$ is analogous.

\[ \{ S : \alpha \subseteq \{ a \} \} \Leftrightarrow \{ a \} \subseteq \{ \alpha \} \]

We conclude that $\alpha$ is continuous.

\[ \{ S : \alpha \subseteq \{ a \} \} \Leftrightarrow \{ a \} \subseteq \{ \alpha \} \]

Let $S \subseteq C$. We reason as follows:

**Proof:** We rely on the fact that, if $x \notin A$, then $x = y$. The mapping $\alpha$ preserves $\sqcap$.

**Claim 3:** The mapping $\alpha$ preserves $\sqcup$ and is therefore $\sqcup$-continuous. The mapping of "Meet" and "Join"'s preservation of "Meet" and "Join"’s
Given as above, we can always complete it into a Galois connection by defining a companion additive \(a\). Note that this claim can also be interpreted as stating that, given a completely analogous:

\[ \{ (n) \subseteq c \mid \forall n \in a \} \sqsubseteq = (a \sigma) \]

The proof of

\[ \{ n \subseteq (a \sigma) c \mid \exists c \in c \} \sqsubseteq = (a \sigma) \]

which, due to the adjunction property, is equal to

\[ \{ n \subseteq c \mid \exists c \in c \} \sqsubseteq = (a \sigma) \]

We first show how \( \sqsubseteq \) is determined by \( a \). Obiviously, we first show how \( \sqsubseteq \) is determined by \( a \).

\textsc{proof:}

\[ \{ (n) \subseteq c \mid \forall n \in a \} \sqsubseteq = (a \sigma) \]

\textit{Claim 4.} Each of \( a \) and \( \sqsubseteq \) uniquely identifies its partner, as given by

\textit{Uniqueness of Inverse Maps}
Galois connections are closed under composition, product, and so on:

If

are Galois connections, then

so is

If

is a Galois connection, for all \( i \in I \), then so is

If

are Galois connections, then so is

Galois connections are closed under composition, product, and so on:
Why do we require the elaborate structure of a Galois connection?

1. If we are certain about the precise definition of \( \land \), we can synthesize \( \land \) and its adjoint \( \lor \) to synthesize abstract operations. For each \( f : C \leftarrow A \), we can use \( x \) and its adjoint \( y \) as a "homomorphism" from \( C \) to \( A \), and vice versa.

2. We obtain many mathematical properties about \( x \), expressed in terms of its adjoint, \( y \) (and vice versa).

3. Since we intend to use \( x \) and its adjoint \( y \) to synthesize abstract terms of \( y \) and \( x \) (and vice versa).

We will see that \( f \circ \# = \# \circ f \). Such that \( \alpha \) is a "homomorphism" with respect to \( f \) and \( \# \). (We can synthesize \( f : \# \rightarrow \# \) as a "homomorphism" from \( C \) to \( A \), and vice versa.)
Closure maps

Definition: A closure map $p: C \rightarrow C'$ is a closure map if it is:

(i) monotonic: for all $c \in C$, $p(c) = p \circ p = p$.

(ii) extensive: $c \subseteq p(c)$, for all $c \in C$.

(iii) idempotent: $p \circ p = p$.

A's elements are mere "tokens" that name distinguished sets in $C$. As a sublattice: for $C \triangleleft C'$, it is common that $x$ is onto. This means $A$ embeds into $C$.

为例，$\{0, 2, 4, ..., \}$
Every closure map, \( p : c \leftarrow c' \), defines the Galois connection, and every Galois connection, \( \langle c', c \rangle \), defines a closure map, \( \vee \circ c \).

\[
\begin{align*}
\{0, 2, 4, \ldots\} & = \{0, 2, 4, \ldots\} \\
\{0, 1, 3, 6, 10, \ldots\} & = \{0, 1, 3, 6, 10, \ldots\}
\end{align*}
\]

A closure map defines the embedding.
Given $C$, we can define an abstract interpretation by selecting some $M$ that is a Moore family.

For a closure map, $p : C \rightarrow C$, its image, $p[C]$, is a Moore family:

\[ \{ \{ c \} \mid c \in M \} \subseteq C. \]

We can define a closure map as $p(c) = \bigcup \{ \{ c \} \mid c \in M \}$.

**Definition:** $M \subseteq C$ is a Moore family iff for all $S \subseteq M$, $(\cup S) \in M$.

Closed under greatest-lower-bounds:

Given $C$, can we define a closure map on it by choosing some elements of $C$? The answer is yes. If the elements of $C$ we select are
Closed binary relations

Oftena Galois connection uses a powerset for its concrete domain, that is, the powerset of a set.

This format yields a simple characterization:

Given unordered set \( A \) and complete lattice \( D \), it is natural to relate the elements in \( D \) to those in \( A \) by a binary relation \( \rho \). Given unordered set \( D \) and complete lattice \( A \), it is natural to relate this format yields a simple characterization:

\[
\forall a \in \rho(D), \quad A \rightarrow \rho(a)
\]

Example: \( D = \text{Int}, \) and \( a \in \rho \) as \( \rho \) a.

\[
\forall a \in \rho \] has property \( a \) means \( a \in \rho \) or as \( a \). We write this as

\[
\text{example: } D = \text{Int}, \text{ and }\]

such that \( (d, a) \in \rho \) or \( a \in \rho \), \( a \) means \( d \) has property \( a \). We write this as

\[
\text{example: } D = \text{Int}, \text{ and }\]

Given unordered set \( D \) and complete lattice \( A \), it is natural to relate this format yields a simple characterization:

\[
\forall a \in \rho(D), \quad A \rightarrow \rho(a)
\]

This format yields a simple characterization:

Ofta a Galois connection uses a powerset for its concrete domain.
and in this sense, \( R \) "is" a Galois connection.

The second item is of course the definition of a Galois connection:

- All \( c \in C \).
- For all \( a \in A \) and \( c \in B \), \( c \subseteq (a) \) iff \( (\forall c \in C; a \sqsubseteq (\forall c \in C; a)) \).

Properties:

If \( R \) defines a Galois connection, then we have these crucial

- \( \text{G-closed: } c \ni a \iff R \).
- \( \text{U-closed: } c \ni a \iff R \).

Proposition: \( R \sqsubseteq A \times B \) defines a Galois connection between \( A \) and \( B \).

Propositionally, \( R \) directly upon \( R \):

We can check that \( \forall [A] \) defines a Moore family. But we can check for this

showing that \( \forall [A] \) defines a Moore family. By we can check for this

We can check if \( \forall \) is the upper adjoint of a Galois connection, say, by

...
A recipe for abstract-domain building

Given an unordered set $\mathcal{D}$ of concretes values, we might take this approach to abstracting its elements:

We ask, "What are the properties about $\mathcal{D}$ that I wish to calculate? Can I relate these properties to elements by means of a UG-closed binary relation?"

Let $\mathcal{A}$ be the existing machinery to define the Galois connection.

Ensure that $\mathcal{A}$ is a Moore family by adding greatest-lower-bound bounds elements to $\mathcal{A}$ as needed.

Define this partial ordering on $\mathcal{A}$: $a \sqsubseteq a'$ if there are distinct $a, a' \in \mathcal{A}$ such that $\forall (a) = \forall (a')$, then merge them.

Define $\forall : A \times \mathcal{D} \supseteq \mathcal{D}$ as $\forall (a) = \{ d | d \in \mathcal{D} \}$. (If there are distinct $a, a' \in \mathcal{A}$ such that $\forall (a) \sqsubseteq \forall (a')$, then merge.)

Use this existing machinery to define the Galois connection.
This produces a Galois connection, \( \langle \text{store}, \text{store} \rangle \) Abstore.

Example: \( \langle x : 3, y : 4 \rangle \) Abstore \( x : \text{any}, y : \text{even} \).

A concrete store is related to an abstract store:

A conjunction that abstracts the store and the state.

Say that we have the relation, \( \text{Data} \subseteq \text{Data} \times \text{AbData} \), and we have the induced Galois connection.

Example: \( \langle x : 1 \rangle \) AbData \( x : \text{Id} \), \( \forall x \in \text{Data} \).

and the concrete program state is a program state.

Store = \( \text{Data} \times \text{IdenticalData} \).

The concrete storage vector is a product, Abstore.
Hence, $\forall s \in \text{store}(\sigma) \Rightarrow \{\{p, \sigma \} | \exists s \in \text{store}(\sigma)\}$.

Finally, we can relate a concrete state to an abstract one:

$\forall s \in \text{store}(\sigma) \Rightarrow \{\{p, \sigma \} | \exists s \in \text{store}(\sigma)\}$

A program point is abstracted to itself: $p \text{pp} p$, suggesting that the abstract domain of program points might be merely $\text{Abspp} = \text{ProgramPoint}$. (A complete lattice.)

For example:

$\forall s \in \text{store}(\sigma) \Rightarrow \{\{l: x \mapsto \text{store}(\sigma) \} | \exists \sigma \in \text{store}(\sigma)\}$

where $\text{AbsStore} = \text{IdentifiableAbsData}$ and