Cours MALG & MOVEX

MALG

Théorie du Point-Fixe et ses Applications

Dominique Méry
Telecom Nancy, Université de Lorraine
(31 mars 2025 at 9:06 A.M.)

Année universitaire 2024-2025

Summary

Summary

Transition system

A transition system $\mathcal{ST}$ is given by a set of states Σ, a set of initial states $Init$ and a binary relation $\mathcal{R}$ on Σ.

- The set of terminal states $Term$ defines specific states, identifying particular states associated with a termination state and this set can be empty, in which case the transition system does not terminate.

event

A transformation is caused by an event that updates a temperature from a sensor, or a computer updating a computer variable, or an actuator sending a signal to a controlled entity.

Observation of a system

An observation of a system S is based on the following points :

- a state $s \in \Sigma$ allows you to observe elements and reports on these elements, such as the number of people in the meeting room or the capacity of the room : $s(np)$ and $s(cap)$ are two positive integers.
- a relationship between two states s and s' observes a transformation of the state s into a state s' and we will note $s \xrightarrow{R} s'$ which expresses the observation of a relationship R :
$R = s(np) \in 0..s(cap){-}1 \wedge s'(np) = s(np){+}1 \wedge s'(cap) = s(cap)$ is an expression of R observing that one more person has entered the room.
- a trace $s_0 \xrightarrow{R_0} s_1 \xrightarrow{R_1} s_2 \xrightarrow{R_2} s_3 \xrightarrow{R} \ldots \xrightarrow{R_{i-1}} s_i \xrightarrow{R_i} \ldots$ is a trace generated by the different observations $R_0, \ldots R_p, \ldots$

Observation of a system

- observing changes of state that correspond either to physical or biological phenomena or to artefactual structures such as a program, a service or a platform.
- An observation generally leads to the identification of a few possible transformations of the observed state, and the closed-model hypothesis follows naturally.
- One consequence is that there are visible transformations and invisible transformations.
- These invisible transformations of the state are expressed by an identity relation called event skip (or stuttering [**?**].
- A modelling produces a closed model with a skip event modelling what is not visible in the observed state.

assertions

- a language of assertions $\mathcal{L}$ (or a language of formulae) is supposed to be given : $\mathcal{P}(\Sigma)$ (the set of parts of Σ)
- $\varphi(s)$ (or $s \in \hat{\varphi}$) means that φ is true in s.
- Properties of a system S which interest us are the state properties expressing that *nothing bad can happen*.
- Examples : *the number of people in the meeting room is always smaller than the maximum allowed by law* or *the computer variable storing the number of wheel revolutions is sufficient and no overflow will happen*.
- Safety properties : the partial correctness (PC) of an algorithm A with respect to its pre/post specifications (PC), the absence of errors at runtime (RTE) ...
- Properties are expressed in the language $\mathcal{L}$ whose elements are combined by logical connectors or by instantiations of variable values in the computer sense called flexible.

- hypothesis : a system S is modelled by a set of states Σ, and $\Sigma \stackrel{def}{=} \mathtt{Var} \longrightarrow \mathrm{D}$ where Var is the variable (or list of variables) of the system S and D is the domain of possible values of variables.
- The interpretation of a formula P in a state $s \in \Sigma$ is denoted $[\![P]\!](s)$ or sometimes $s \in \hat{P}$.
- A distinction is made between flexible variable symbols $\mathtt{x}$ and logical variable symbols v, and constant symbols c are used.

1. $[\![\mathrm{x}]\!](s) = s(\mathrm{x}) = x$: x is the value of the variable x in s.
2. $[\![\mathrm{x}]\!](s') = s'(\mathrm{x}) = x'$: x' is the value of the variable x in s'.
3. $[\![c]\!](s)$ is the value of c in s, in other words the value of the constant c in s.
4. $[\![\varphi(x) \wedge \psi(x)]\!](s) = [\![\varphi(x)]\!](s)$ et $[\![\psi(x)]\!](s)$ where *and* is the classical interpretation of symbol $\wedge$ according to the truth table.
5. $[\![\mathrm{x} = 6 \wedge y = \mathrm{x}{+}8]\!](s) \stackrel{def}{=} [\![\mathrm{x}]\!](s) = [\![6]\!](s)$ **and** $[\![y]\!](s) = [\![x]\!](s){+}[\![8]\!](s) = (x = 6$ **and** $y = x{+}8$ where y is a logical variable distinct of x and where $[\![\mathrm{x}]\!](s) = s(\mathrm{x}) = x$.

Notations

- $[\![\mathrm{x}]\!](s)$ is the value of `x` in s and its value will be distinguished by the font used : `x` is the `tt` font of LaTeX and x is the math font of LaTeX.
- Using the name of the variable `x` as its current value, i.e. x and $[\![\mathrm{x}]\!](s')$ is the value of `x` in s' and will be noted x'.
- The transition relation as a relation linking the state of the variables in s and the state of the variables in s' using the prime notation as defined by L. Lamport for TLA.
- Types of variable depending on whether we are talking about the computer variable, its value or whether we are defining constants such as np, the number of processes, or π, which designates the constant π.
- a current observation refers to a current state for both endurant and perdurant information data in the sense of the Dines Bjørner.

Notations

flexible variable

A flexible variable `x` is a name related to a perdurant information according to a state of the (current observed) system :

- x is the current value of `x` in other words the value at the observation time of `x`.
- x' is the next value of `x` in other words the value at the next observation time of `x`.
- x_0 is the initial value of `x` in other words the value at the initial observation time of `x`.

A logical variable x is a name related to an endurant entity designated by this name.

state property of a system

Let be a system S whose flexible variables x are the elements of $\mathcal{V}ar(S)$. A property $P(x)$ of S is a logical expression involving ,freely the flexible variables x and whose interpretation is the set of values of the domain of x : $P(x)$ is true in x, if the value x satisfies $P(x)$.
For each property $P(x)$, we can associate a subset of D denoted $\hat{\mathrm{P}}$ and, in this case, $P(x)$ is true in x. is equivalent to $x \in \hat{\mathrm{P}}$.

Examples of property

- $P_1(x) \stackrel{def}{=} x \in 18..22$: x is a value between 18 and 22 and $\hat{P}_1 = \{18, 19, 20, 21, 22\}$.
- $P_2(p) \stackrel{def}{=} p \subset PEOPLE \wedge card(p) \leq n$: p is a set of persons and that set has at most n elements and $\hat{P}_2 = \{p_1 \ldots p_n\}$. In this example, we use a logical variable n and a name for a constant $PEOPLE$.

basic set of a system S

The list of symbols $s_1, s_2, \ldots, s_p$ corresponds to the list of basic set symbols in the D domain of S and $s_1 \cup \ldots \cup s_p \subseteq \mathrm{D}$.

constants of system S

The list of symbols $c_1, c_2, \ldots, c_q$ corresponds to the list of symbols for the constants of S.

Examples of constant and set

- $fred$ is a constant and is linked to the set $PEOPLE$ using the expression $fred \in PEOPLE$ which means that $fred$ is a person from $PEOPLE$.
- aut is a constant which is used to express the table of authorisations associated with the use of vehicles. the expression $aut \subseteq PEOPLE \times CARS$ where $CARS$ denotes a set of cars.

axiom of system S

An axiom ax(s,c) of S is a logical expression describing a constant or constants of S and can be defined as an expression depending on symbols of constants expressing a set-theoretical expression using symbols of sets and symbols of constants already defined.

Examples of axiom

- $ax1(fred \in PEOPLE)$: *fred is a person from the set PEOPLE*
- $ax2(suc \in \mathbb{N} \rightarrow \mathbb{N} \wedge (!i.i \in \mathbb{N} \Rightarrow suc(i) = i{+}1))$: *The function suc is the total function which associates any natural i with its successor. successor*
- $ax3(\forall A.A \subseteq \mathbb{N} \wedge 0 \in \mathbb{N} \wedge suc[A] \subseteq A \Rightarrow \mathbb{N} \rightarrow \subseteq A)$: This axiom states the induction property for natural numbers. It is an instantiation of the fixed-point theorem.
- $ax4(\forall x.x = 2 \Rightarrow x{+}2 = 1)$: This axiom poses an obvious problem of consistency and care should be taken not to use this kind of statement as axiom.

Axiomatics for S

. . . .

⊠ Definition(axiomatics for S)

The list of axioms of S is called the axiomatics of S and is denoted $AX(S, s, c)$ where s denotes the basic sets and c denotes the constants of S.

. . . .

. . . .

⊠ Definition(theorem for S)

A property $P(s, c)$ is a theorem for S, if $AX(S, s, c) \vdash P(s, c)$ is a valid sequent.

Theorems for S are denoted by $TH(S, s, c)$.

. . . .

Let s, s' be two states of S ($s, s' \in \mathcal{V}ar(S) \longrightarrow \textsc{Vals}$). $s \underset{R}{\longrightarrow} s'$ will be written as a relation $\mathrm{R}(x, x')$ where x and x' designate values of $\mathtt{x}$ before and after the observation of R.

. . . .

⊠ Definition(event)

Let $\mathcal{V}ar(S)$ be the set of flexible variables of S. Let s be the basis sets and c the constants of S. An event e for S is a relational expression of the form $R(s, c, x, x')$ denoted $BA(e)(s, c, x, x')$.

☒ Definition(event-based model of a system)

Let $\mathcal{V}ar(S)$ be the set of flexible variables of S denoted x. Let s be the list of basis sets of the system S. Let c be the list of constants of the system S. Let D be a domain containing sets s. An event-based model for a system S is defined by

$$(AX(s,c), x, \textsc{Vals}, Init(x), \{e_0, \ldots, e_n\})$$

where

- $AX(s,c)$ is an axiomatic theory defining the sets, constants and static properties of these elements.
- $Init(x)$ defines the possible initial values of x.
- $\{e_0, \ldots, e_n\}$ is a finite set of events of S and e_0 is a particular event present in each event-based model defined by $BA(e_0)(x,x') = (x' = x)$.

The event-based model is denoted $EM(s,c,x,\textsc{Vals}, Init(x)\{e_0, \ldots, e_n\}) = (AX(s,c), x, \textsc{Vals}, Init(x), \{e_0, \ldots, e_n\})$.

- $Next(x, x')$ or
  $Next(s, c, x, x') \stackrel{def}{=} BA(e_0)(s, c, x, x') \vee \ldots \vee BA(e_n)(s, c, x, x')$.
- the transitive reflexive closure of the relation $Next^{\star}(s, c, x_0, x) \stackrel{def}{=}$
  $$\left\{ \begin{array}{l} \vee\ x = x_0 \\ \vee\ Next(s, c, x_0, x) \\ \vee\ \exists\ xi \in \text{VALS}.Next^{\star}(s, c, x_0, xi) \wedge Next(s, c, xi, x) \end{array} \right.$$

. . . .

⊠ Definition(safety property)
A property $P(x)$ is a safety property for the system S, if

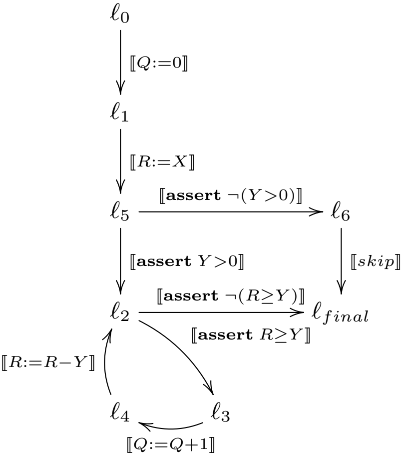
$$\forall x_0, x \in \text{VALS}.Init(x_0) \wedge Next^{\star}(s, c, x_0, x) \Rightarrow P(x).$$

. . . .

Summary

Programme as flowchart

$\ell_0[Q := 0];$
$\ell_1[R := X];$
IF $\ell_5[Y > 0]$
  WHILE $\ell_2[R \geq Y]$
    $\ell_3[Q := Q+1];$
    $\ell_4[R := R-Y]$
  ENDWHILE
ELSE
  $\ell_6[skip]$
ENDIF



Safety Properties and Invariance Properties for a Relational Model

. . . .

⊠ Definition(assertion)

Let $(Th(s,c), \mathsf{X}, \textsc{Vals}, \textsc{Init}(x), \{r_0, \ldots, r_n\})$ a relational model M of a system $\mathcal{S}$. A property A is an assertional safety property for the system $\mathcal{S}$, if

$$\forall x_0, x \in \textsc{Vals}.Init(x_0) \wedge \textsc{Next}^\star(x_0, x) \Rightarrow A(x).$$

. . . .

. . . .

⊠ Definition(relation)

Let $(Th(s,c), \mathsf{X}, \textsc{Vals}, \textsc{Init}(x), \{r_0, \ldots, r_n\})$ a relational model M of a system $\mathcal{S}$. A property R is a relational safety property for the system $\mathcal{S}$, if

$$\forall x_0, x \in \textsc{Vals}.Init(x_0) \wedge \textsc{Next}^\star(x_0, x) \Rightarrow R(x_0, x).$$

. . . .

Summary

Invariant for checking a safety property

- The event-based model for a system $\mathcal{S}$: is defined as $EM(s, c, x, \text{VALS}, Init(x)\{e_0, \ldots, e_n\}) = (AX(s, c), x, \text{VALS}, Init(x), \{e_0, \ldots, e_n\})$.
- $\text{REACHABLE}(\mathcal{S}) = \{u | u \in \text{VALS} \wedge \exists u0 \in \text{VALS}.\, (Init(u0) \wedge \text{NEXT}^{\star}(u0, u))\}$ is the set of values reachable for the system $\mathcal{S}$.

Let consider A a safety property for a system $\mathcal{S}$:

- $\forall x_0, x \in \text{VALS}. Init(x_0) \wedge \text{NEXT}^{\star}(x_0, x) \Rightarrow A(x)$.

  if, and only if,
- $\text{REACHABLE}(\mathcal{S}) \subseteq \{u | u \in \text{VALS} \wedge A(u)\}$

$\text{REACHABLE}(\mathcal{S})$ satisfies the following properties

- $\{u | u \in \text{VALS} \wedge Init(u)\} \subseteq \text{REACHABLE}(\mathcal{S})$
- $\text{NEXT}[\text{REACHABLE}(\mathcal{S})] \subseteq \text{REACHABLE}(\mathcal{S})$
- $\forall U. \left(\begin{array}{c} U \subseteq \text{VALS} \\ \{u | u \in \text{VALS} \wedge Init(u)\} \subseteq U \\ \wedge \\ \text{NEXT}[U] \subseteq U \end{array} \right) \Rightarrow \text{REACHABLE}(\mathcal{S}) \subseteq U$

$\text{NEXT}[U] = \{v | v \in \Sigma \wedge \exists u.(u \in U \wedge Next(u, v))\}$

Principle for exhaustive checking

Let a property A pour $\mathcal{S}$

- $\forall x_0, x \in \textsc{Vals}.Init(x_0) \wedge \textsc{Next}^\star(x_0, x) \Rightarrow A(x)$.

  if, and only if,

- $\textsc{reachable}(\mathcal{S}) \subseteq \{u | u \in \textsc{Vals} \wedge A(u)\}$

- $\textsc{reachable}(\mathcal{S}) = \{u | u \in \textsc{Vals} \wedge Init(u)\} \cup \textsc{Next}[\textsc{reachable}(\mathcal{S})]$
- $F \in \mathcal{P}(\textsc{Vals}) \longrightarrow \mathcal{P}(\textsc{Vals})$
- $F(U) = \{u | u \in \textsc{Vals} \wedge Init(u)\} \cup \textsc{Next}[U]$
- $F(\textsc{reachable}(\mathcal{S})) = \textsc{reachable}(\mathcal{S})$

Pfroblems related to fixex-point theory

- Solving equations as $X = \mathcal{F}(X)$
- Finding and even better computing solutions when they exist.

Theorem of Picard

Let (E, d) a complete metric space and $f : E \longrightarrow E$ a contracting function, ie there exists $k \in [0, 1[$ such that for any $(x, y) \in E$, $d(f(x), f(y)) \leq k{\cdot}d(x, y)$. Then f belongs a unique fixed-point ℓ. Moreover, every sequence defined by $u_0 \in I$, $u_{n+1} = f(u_n)$, converges to the unique fixed-point with the following estimates :

- $d(u_n, \ell) \leq k^n \times d(u_0, l)$
- $d(u_n, \ell) \leq k/(1-k) \times d(u_n, u_{n-1})$

Fixed-point theorem 2 : inductive structure

```
struct liste {
        int val;
        struct liste *next
        }

int longueur(struct liste *l)
        {
        if (l == NIL) {return(0);}
        else {return(1 + longueur(l -> next));}
```

Fixed-point theorem 3 : recursive definition

```c
#include <stdio.h>
int  f (int x, int y)
{  int i;
   if (x==y)
     {i = y+1;return(i);}
   else
     {i=f(x,f(x-1,y+1));return(i);}
}
int main ()
{
   int a =  2;
   int b = 2;
   printf("Valeur:%d\n",f(a,b));
}
```

```
fun A(0,y) = y + 1
| A(x,0) = A(x-1,1)
| A(x,y) = A(x-1,A(x,y-1));
val A = fn : int * int -> int
```

Property to prove

- $\forall (x,y) \in nat \times nat : (x,y) \in Domaine(A)$

ou

- $\forall (x,y) \in nat \times nat : P(x,y)$ avec $P(x,y) \stackrel{def}{=} ((x,y) \in domaine(A))$.

Méthode

1. Choose a well founded structure nat : $(nat \times nat, <_{lex})$
2. Use the principle of induction for the well founded structure
   1. $\forall y \in nat : P(0, y)$
   2. $\forall x \in nat : x \neq 0 : P(x, 0)$
   3. $\forall (x, y) \in nat \times nat : x \neq 0 : y \neq 0 : P(x, y)$

- Question : what is the meaning of the next function ?
  $f(x,y) = if\ x = y\ then\ y{+}1\ else\ f(x, f(x{-}1, y{+}1))\ fi$

Summary

Une structure partiellement ordonnée $(X, \sqsubseteq)$ est définie par un ensemble X et une relation d'ordre partiel $\sqsubseteq$.

Relation d'ordre partiel $\sqsubseteq$

- $\sqsubseteq \subseteq X \times X$
- $\{(a,a) | a \in X\} \subseteq \sqsubseteq$ (réflexivité)
- $\forall (a,b).((a \in X \land b \in X \land (a \sqsubseteq b) \land (b \sqsubseteq a)) \Rightarrow (a = b))$ (anti-symétrie)
- $(\sqsubseteq ; \sqsubseteq) \subseteq \sqsubseteq$ (transitivité)

Soit une structure partiellement ordonnée $(X, \sqsubseteq)$. Soit une fonction totale F sur X à valeurs dans X. On dit que $fp \in X$ est un point-fixe de F, si $F(fp) = fp$.

Point-fixe fp de F

- $fp \in X$
- $F \in X \longrightarrow X \quad F(fp) = fp$

Structure partiellement ordonnée inductive (ou CPO)

Une structure partiellement ordonnée inductive (ou un CPO Complete Partially Ordered Set) est une structure partiellement ordonnée $(D, \sqsubseteq)$ telle que :

1. D admet un plus petit élément noté $\perp_D$: $\forall d \in D. \perp_D \sqsubseteq d$.
2. tout ensemble dirigé X de D (X est dirigé, si X est non-vide et si $\forall x, y \in X. \exists z \in X. [x \sqsubseteq z \; et \; y \sqsubseteq z]$) admet une borme supérieure dans D.

Chaine (ou structure totalement ordonnée)

Une structure partiellement ordonnée inductive est une chaîne, si :

$$\forall d, d' \in D, d \sqsubseteq d' oud' \sqsubseteq d.$$

Propriété de réduction aux chaines (théorème)

Une structure partiellement ordonnée inductive (ou un CPO Complete Partially Ordered Set, ou une structure partiellement ordonnée complète (inductive)) est une structure partiellement ordonnée $(D, \sqsubseteq)$ telle que :

1. D admet un plus petit élément noté $\perp_D : \forall d \in D. \perp_D \sqsubseteq d$.
2. toute chaine C de D admet une borme supérieure dans D.

- Montrer qu'une structure est une structure partiellement ordonnée inductive revient à ne montrer le propriété que pour les chaines de la structure.
- La démonstration utilise l'axiome du choix qui permet de construire une chaine à partir d'un ensemble dirigé.

✓Exemple 1 de CPO

Flat CPO

Soit D un ensemble et $\perp$ un élément qui n'est pas élément de D. On définit l'ordre partiel sur $D \cup \{\perp\}$ $\sqsubseteq$ comme suit :

$$\forall d \in D \cup \{\perp\}.d \sqsubseteq d$$
$$\forall d \in D \cup \{\perp\}.\perp \sqsubseteq d$$

On notera $D^{\perp} = D \cup \{\perp\}$. $(D^{\perp}, \sqsubseteq)$ est un CPO.

✓Exemple 2 de CPO

Ensemble des fonctions partielles $A \nrightarrow B$

Soit A un ensemble et B un autre ensemble. On notera $\Im(A,B)$ ou $A \nrightarrow B$ l'ensemble des fonctions de A dans B.
$f \sqsubseteq g$ si, et seulement si, $(dom(f) \subseteq dom(g))$ et $(\forall x \in dom(f).f(x) = g(x))$.

Propriété de $(A \nrightarrow B, \sqsubseteq)$

$(A \nrightarrow B, \sqsubseteq)$ est une structure partiellement ordonnée inductive où $\perp_{A \nrightarrow B}$ est la fonction définie nulle part.

✓Exemple 3 de CPO

Ensemble des parties d'un ensemble

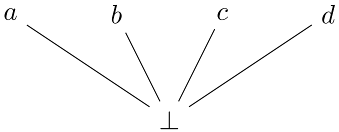
Soit E un ensemble et $\mathcal{P}(E)$, l'ensemble des parties de E. $(\mathcal{P}(E), \subseteq)$ est une structure partiellement ordonnée inductive.

✓Notations

Soit $(A, \sqsubseteq)$ une structure partiellement ordonnée et $D \subseteq A$.

- x est un majorant de D, si $\forall d \in D : d \sqsubseteq x$
- x est un minorant de D, si $\forall d \in D : x \sqsubseteq d$
- La borne supérieure d'une partie de D est le plus petit des majorants.
- La borne inférieure d'une partie de D est le plus grand des minorants.
- $x \sqcup y$ est la borne supérieure de $\{x, y\}$, lorsqu'elle existe.
- $x \sqcap y$ est la borne inférieure de $\{x, y\}$, lorsqu'elle existe.
- $Sup(D) = \sqcup D = \vee D$: borne supérieure de D quand elle existe.
- $Min(D) = \sqcap D = \wedge D$: borne inférieure de D quand elle existe.

✓Illustration des notations



- $D = \{a, b, c, d\}$
- $D^{\perp} = \{a, b, c, d\}$

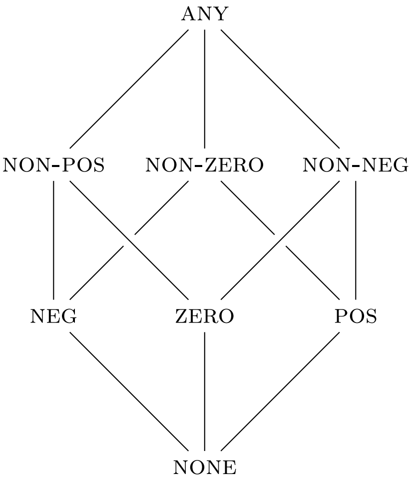
✓Illustration ds notations

$$
\begin{tikzcd}[column sep=small]
 & a \sqcup b \arrow[dl, no head] \arrow[dr, no head] & \\
a \arrow[dr, no head] & & b \arrow[dl, no head] \\
 & a \sqcap b &
\end{tikzcd}
$$

✓Illustration ds notations

✓Illustration ds notations

✓Illustration ds notations



- ANY désigne l'ensemble des entiers
- NON-POS désigne l'ensemble des entiers négatifs ou nuls
- POS désigne l'ensemble des entiers positifs non nuls.
- ZERO est l'ensemble contenant uniquement 0.

- Une fonction f de $(D_1, \sqsubseteq)$, dans $(D_2, \sqsubseteq)$ est monotone croissante, si :

$$\forall d, d' \in D_1, d \sqsubseteq_1 d' \Rightarrow f(d) \sqsubseteq_2 f(d')$$

- Un ouvert O de D est une partie de D satisfaisant :
  1. Si $x \in O$ et $x \sqsubseteq y$, alors $y \in O$.
  2. Si X est une partie dirigée de D telle que $\sqcup X \in O$, alors $X \cap O \neq \varnothing$.

- Une fonction f est continue pour une topologie donnée, si l'image réciproque de tout ouvert O est un ouvert : si O est un ouvert, $f^{-1}(O)$ est un ouvert.
- Théorème : Soit f une fonction définie de $(D, \sqsubseteq)$ dans $(D', \sqsubseteq')$. f est continue pour la topologie de Scott si, et seulement si, pour tout ensemble dirigé X de D, $f(\sqcup X) = \sqcup f(X)$.
- Une première conséquence du choix de la topologie de Scott est que toute fonction continue est monotone.

Summary

Point-fixe pour une fonction continue sur un CPO

Théorème de Kleene

Soit f une fonction continue sur $(D, \sqsubseteq)$ à valeurs dans $(D, \sqsubseteq)$ où $(D, \sqsubseteq)$ est une structure partiellement ordonnée inductive.
Alors il existe un élément x de D tel que

$$\begin{cases} f(x) = x \\ \forall y \in D : (f(y) = y) \Rightarrow (x \sqsubseteq y) \end{cases}$$

et on le notera μf.

Preuve du théorème

Preuve du théorème

- $x \stackrel{def}{=} \bigvee_{i \geq 0} f^i$ où $\forall i \in nat^\star : f^{i+1} = f(f^i)$ et $f^0 = \perp_D$:
  x existe car $(D, \sqsubseteq)$ est une structure inductive.

Preuve du théorème

- $x \stackrel{def}{=} \bigvee_{i \geq 0} f^i$ où $\forall i \in nat^\star : f^{i+1} = f(f^i)$ et $f^0 = \perp_D$:
  x existe car $(D, \sqsubseteq)$ est une structure inductive.
- x est un point-fixe de f :
  $f(x) \stackrel{\textit{definition}}{=} f(\bigvee_{i \geq 0} f^i) \stackrel{\textit{continuite}}{=} \bigvee_{i \geq 0} f(f^i)$
  $\bigvee_{i \geq 0} f(f^i) \stackrel{\textit{definition de la suite}}{=} \bigvee_{i \geq 0} f^{i+1} \stackrel{\textit{renommage}}{=}$
  $\bigvee_{j > 0} f^j \stackrel{\textit{propriete de } \perp_D}{=} \perp_D \sqcup \bigvee_{j > 0} f^j = x$

Preuve du théorème

- $x \stackrel{def}{=} \bigvee_{i \geq 0} f^i$ où $\forall i \in nat^\star : f^{i+1} = f(f^i)$ et $f^0 = \bot_D$:
  x existe car $(D, \sqsubseteq)$ est une structure inductive.
- x est un point-fixe de f :
  $f(x) \stackrel{\textit{definition}}{=} f(\bigvee_{i \geq 0} f^i) \stackrel{\textit{continuite}}{=} \bigvee_{i \geq 0} f(f^i)$
  $\bigvee_{i \geq 0} f(f^i) \stackrel{\textit{definition de la suite}}{=} \bigvee_{i \geq 0} f^{i+1} \stackrel{\textit{renommage}}{=}$
  $\bigvee_{j > 0} f^j \stackrel{\textit{propriete de } \bot_D}{=} \bot_D \sqcup \bigvee_{j>0} f^j = x$
- x est le plus petit point-fixe de F.
  Soit y un autre point-fixe de f :
  - $\bot_D \sqsubseteq y$
  - Si $\forall j \in \{0, \ldots, i-1\} : f^j \sqsubseteq y$, alors $f^i \sqsubseteq y$.
  - $\forall j \in nat : f^j \sqsubseteq y$

  D'où $x \sqsubseteq y$.
  □

Un exemple de fonction

```
int f5(int x)
{if (x==0)
    { return(0);}
  else
    { if  (x > 0)
 {return(2-f5(1-x));}
      else
 {/* x <0 */
   return(f5(-x));}
    }
}
```

```
int f5(int x)
{if (x==0)
    { return(0);}
  else
    { if  (x > 0)
 {return(2-f5(1-x));}
        else
 {/* x <0 */
   return(f5(-x));}
    }
}
```

- Déterminer l'espace du problème : $\mathbb{Z} \nrightarrow \mathbb{Z}$
- Définir la fonctionnelle définissant l'équation du problème : $\mathcal{F} \in (\mathbb{Z} \nrightarrow \mathbb{Z}) \longrightarrow (\mathbb{Z} \nrightarrow \mathbb{Z})$

Un exemple de fonction

```
int f5(int x)
{if (x==0)
    { return(0);}
  else
    { if  (x > 0)
 {return(2-f5(1-x));}
      else
 {/* x <0 */
   return(f5(-x));}
    }
}
```

- Déterminer l'espace du problème : $\mathbb{Z} \nrightarrow \mathbb{Z}$
- Définir la fonctionnelle définissant l'équation du problème : $\mathcal{F} \in (\mathbb{Z} \nrightarrow \mathbb{Z}) \longrightarrow (\mathbb{Z} \nrightarrow \mathbb{Z})$
- $(\mathbb{Z} \nrightarrow \mathbb{Z}, \sqsubseteq)$ est un CPO pour l'ordre *moins défini que*.

Un exemple de fonction

```
int f5(int x)
{if  (x==0)
    { return(0);}
  else
    { if   (x > 0)
 {return(2-f5(1-x));}
        else
 {/* x <0 */
   return(f5(-x));}
    }
}
```

- Déterminer l'espace du problème : $\mathbb{Z} \nrightarrow \mathbb{Z}$
- Définir la fonctionnelle définissant l'équation du problème : $\mathcal{F} \in (\mathbb{Z} \nrightarrow \mathbb{Z}) \longrightarrow (\mathbb{Z} \nrightarrow \mathbb{Z})$
- $(\mathbb{Z} \nrightarrow \mathbb{Z}, \sqsubseteq)$ est un CPO pour l'ordre *moins défini que*.
- $\mathcal{F}$ est continue pour la topologie de Scott

Un exemple de fonction

```
int f5(int x)
{if (x==0)
    { return(0);}
  else
    { if  (x > 0)
 {return(2-f5(1-x));}
      else
 {/* x <0 */
   return(f5(-x));}
    }
}
```

- Déterminer l'espace du problème : $\mathbb{Z} \nrightarrow \mathbb{Z}$
- Définir la fonctionnelle définissant l'équation du problème : $\mathcal{F} \in (\mathbb{Z} \nrightarrow \mathbb{Z}) \longrightarrow (\mathbb{Z} \nrightarrow \mathbb{Z})$
- $(\mathbb{Z} \nrightarrow \mathbb{Z}, \sqsubseteq)$ est un CPO pour l'ordre *moins défini que*.
- $\mathcal{F}$ est continue pour la topologie de Scott
- $f5$ satisfait l'équation $\mathcal{F}(f5) = f5$

Un exemple de fonction

```
int f5(int x)
{if  (x==0)
    { return(0);}
  else
    { if   (x > 0)
 {return(2-f5(1-x));}
        else
 {/* x <0 */
   return(f5(-x));}
    }
}
```

- Déterminer l'espace du problème : $\mathbb{Z} \not\to \mathbb{Z}$
- Définir la fonctionnelle définissant l'équation du problème : $\mathcal{F} \in (\mathbb{Z} \not\to \mathbb{Z}) \longrightarrow (\mathbb{Z} \not\to \mathbb{Z})$
- $(\mathbb{Z} \not\to \mathbb{Z}, \sqsubseteq)$ est un CPO pour l'ordre *moins défini que*.
- $\mathcal{F}$ est continue pour la topologie de Scott
- $f5$ satisfait l'équation $\mathcal{F}(f5) = f5$

$$\boxed{\mathcal{F}(f)(x) = \textbf{if } (x == 0) \textbf{ then } 0 \textbf{ elseif } (x > 0) \textbf{ then } 2 - f(1-x)) \textbf{ else } f(-x) \textbf{ fi}}$$

Développer la solution par itération

Développer la solution par itération

- $\mathcal{F}^0 = \{\}$

Développer la solution par itération

- $\mathcal{F}^0 = \{\}$
- $\mathcal{F}^1 = \mathcal{F}(\mathcal{F}^0) = \{(0,0)\}$

Développer la solution par itération

- $\mathcal{F}^0 = \{\}$
- $\mathcal{F}^1 = \mathcal{F}(\mathcal{F}^0) = \{(0,0)\}$
- $\mathcal{F}^2 = \mathcal{F}(\mathcal{F}^1) = \{(0,0),(1,2)\}$

Développer la solution par itération

- $\mathcal{F}^0 = \{\}$
- $\mathcal{F}^1 = \mathcal{F}(\mathcal{F}^0) = \{(0,0)\}$
- $\mathcal{F}^2 = \mathcal{F}(\mathcal{F}^1) = \{(0,0),(1,2)\}$
- $\mathcal{F}^3 = \mathcal{F}(\mathcal{F}^2) = \{(-1,2),(0,0),(1,2)\}$

Développer la solution par itération

- $\mathcal{F}^0 = \{\}$
- $\mathcal{F}^1 = \mathcal{F}(\mathcal{F}^0) = \{(0,0)\}$
- $\mathcal{F}^2 = \mathcal{F}(\mathcal{F}^1) = \{(0,0),(1,2)\}$
- $\mathcal{F}^3 = \mathcal{F}(\mathcal{F}^2) = \{(-1,2),(0,0),(1,2)\}$
- $\mathcal{F}^4 = \mathcal{F}(\mathcal{F}^3) = \{(-1,2),(0,0),(1,2),(2,0)\}$

Développer la solution par itération

- $\mathcal{F}^0 = \{\}$
- $\mathcal{F}^1 = \mathcal{F}(\mathcal{F}^0) = \{(0,0)\}$
- $\mathcal{F}^2 = \mathcal{F}(\mathcal{F}^1) = \{(0,0),(1,2)\}$
- $\mathcal{F}^3 = \mathcal{F}(\mathcal{F}^2) = \{(-1,2),(0,0),(1,2)\}$
- $\mathcal{F}^4 = \mathcal{F}(\mathcal{F}^3) = \{(-1,2),(0,0),(1,2),(2,0)\}$
- $\mathcal{F}^5 = \mathcal{F}(\mathcal{F}^4) = \{(-2,0),(-1,2),(0,0),(1,2),(2,0)\}$

Développer la solution par itération

- $\mathcal{F}^0 = \{\}$
- $\mathcal{F}^1 = \mathcal{F}(\mathcal{F}^0) = \{(0,0)\}$
- $\mathcal{F}^2 = \mathcal{F}(\mathcal{F}^1) = \{(0,0),(1,2)\}$
- $\mathcal{F}^3 = \mathcal{F}(\mathcal{F}^2) = \{(-1,2),(0,0),(1,2)\}$
- $\mathcal{F}^4 = \mathcal{F}(\mathcal{F}^3) = \{(-1,2),(0,0),(1,2),(2,0)\}$
- $\mathcal{F}^5 = \mathcal{F}(\mathcal{F}^4) = \{(-2,0),(-1,2),(0,0),(1,2),(2,0)\}$
- $\mathcal{F}^6 = \mathcal{F}(\mathcal{F}^4) = \{(-2,0),(-1,2),(0,0),(1,2),(2,0),(3,2)\}$

Développer la solution par itération

- $\mathcal{F}^0 = \{\}$
- $\mathcal{F}^1 = \mathcal{F}(\mathcal{F}^0) = \{(0,0)\}$
- $\mathcal{F}^2 = \mathcal{F}(\mathcal{F}^1) = \{(0,0),(1,2)\}$
- $\mathcal{F}^3 = \mathcal{F}(\mathcal{F}^2) = \{(-1,2),(0,0),(1,2)\}$
- $\mathcal{F}^4 = \mathcal{F}(\mathcal{F}^3) = \{(-1,2),(0,0),(1,2),(2,0)\}$
- $\mathcal{F}^5 = \mathcal{F}(\mathcal{F}^4) = \{(-2,0),(-1,2),(0,0),(1,2),(2,0)\}$
- $\mathcal{F}^6 = \mathcal{F}(\mathcal{F}^4) = \{(-2,0),(-1,2),(0,0),(1,2),(2,0),(3,2)\}$
- $\mathcal{F}^7 = \mathcal{F}(\mathcal{F}^4) = \{(-3,0),(-2,0),(-1,2),(0,0),(1,2),(2,0),(3,2)\}$

Développer la solution par itération

- $\mathcal{F}^0 = \{\}$
- $\mathcal{F}^1 = \mathcal{F}(\mathcal{F}^0) = \{(0,0)\}$
- $\mathcal{F}^2 = \mathcal{F}(\mathcal{F}^1) = \{(0,0),(1,2)\}$
- $\mathcal{F}^3 = \mathcal{F}(\mathcal{F}^2) = \{(-1,2),(0,0),(1,2)\}$
- $\mathcal{F}^4 = \mathcal{F}(\mathcal{F}^3) = \{(-1,2),(0,0),(1,2),(2,0)\}$
- $\mathcal{F}^5 = \mathcal{F}(\mathcal{F}^4) = \{(-2,0),(-1,2),(0,0),(1,2),(2,0)\}$
- $\mathcal{F}^6 = \mathcal{F}(\mathcal{F}^4) = \{(-2,0),(-1,2),(0,0),(1,2),(2,0),(3,2)\}$
- $\mathcal{F}^7 = \mathcal{F}(\mathcal{F}^4) = \{(-3,0),(-2,0),(-1,2),(0,0),(1,2),(2,0),(3,2)\}$
- $\ldots$

Développer la solution par itération

- $\mathcal{F}^0 = \{\}$
- $\mathcal{F}^1 = \mathcal{F}(\mathcal{F}^0) = \{(0,0)\}$
- $\mathcal{F}^2 = \mathcal{F}(\mathcal{F}^1) = \{(0,0),(1,2)\}$
- $\mathcal{F}^3 = \mathcal{F}(\mathcal{F}^2) = \{(-1,2),(0,0),(1,2)\}$
- $\mathcal{F}^4 = \mathcal{F}(\mathcal{F}^3) = \{(-1,2),(0,0),(1,2),(2,0)\}$
- $\mathcal{F}^5 = \mathcal{F}(\mathcal{F}^4) = \{(-2,0),(-1,2),(0,0),(1,2),(2,0)\}$
- $\mathcal{F}^6 = \mathcal{F}(\mathcal{F}^4) = \{(-2,0),(-1,2),(0,0),(1,2),(2,0),(3,2)\}$
- $\mathcal{F}^7 = \mathcal{F}(\mathcal{F}^4) = \{(-3,0),(-2,0),(-1,2),(0,0),(1,2),(2,0),(3,2)\}$
- $\ldots$
- $g(x) = \textbf{if } odd(x) \textbf{ then } 2 \textbf{ else } 0 \textbf{ fi}$ est solution de cette équation et $\mu\mathcal{F} \sqsubseteq g$ mais on doit montrer que $\mu\mathcal{F} = g$

- Un prédicat P sur un ensemble inductif $(D, \sqsubseteq)$ (CPO) est admissible, si, pour tout ensemble dirigé A de D, si, pour tout a de A $P(a)$, alors $P(\bigvee A)$.
- Principe d'induction du point fixe :
  Soit $(D, \sqsubseteq)$ une structure inductive, P un prédicat admissible sur D, F une fonction continue de D dans D.
  Si
  1. $P(\perp_D)$
  2. Soit $d \in D$ tel que $P(d)$:

$\vdots$

$P(F(d))$.

Alors $P(\mu F)$.

- $P(x)$ défini par $x = a$ est admissible.
- $P(x)$ défini par $x \sqsubseteq a$ est admissible.
- Si f_i et g_i sont deux suites de fonctions continues sur un CPO $(D, \sqsubseteq)$, alors $\bigwedge_i f_i(x) \sqsubseteq g_i(x)$ est un prédicat admissible.

Fonction 91 de MacCarthy

- $F(f)(x) = si\ 100 < x\ alors\ x{-}10\ sinon\ f(f(x{+}11))\ fi$ et $g(x) = si\ 100 < x\ then\ x{-}10\ sinon\ 91\ fi$. Montrez que $\mu F \sqsubseteq g$.

- $F(f)(x) = si\ x > 100\ alors\ x{-}10\ sinon\ f(f(x{+}11))\ fi$
- et $g(x) = si\ x > 100\ then\ x{-}10\ sinon\ 91\ fi$.
- On peut montrer que $\mu f \sqsubseteq g$.
- Puis on peut en déduire que $\mu f = g$, en montrant que la fonction μf est totale.

Fonction f5

$$\mathcal{F}(f)(x) = \textbf{if } (x == 0) \textbf{ then } 0 \textbf{ elseif } (x > 0) \textbf{ then } 2 - f(1-x)) \textbf{ else } f(-x) \textbf{ fi}$$

- $g(x) = \textbf{if } odd(x) \textbf{ then } 2 \textbf{ else } 0 \textbf{ fi}$ est solution de cette équation
- Utilisation de $P(f) \stackrel{def}{=} f \sqsubseteq g$
- Montrer que $dom(\mu f) = \mathbb{N}$:
  - $\mathcal{F}^{2n} = \{(p, v_p) | 0 \leq p < n \wedge v_p = g(p)\} \cup \{(p, v_p) | 0 > p \geq -(n-1) \wedge v_p = g(p)\} \cup \{(n, g(n))\}$
  - $\mathcal{F}^{2n+1} = \{(p, v_p) | 0 \leq p < n \wedge v_p = g(p)\} \cup \{(p, v_p) | 0 > p \geq -(n-1) \wedge v_p = g(p)\} \cup \{(n, g(n)), (-n, g(-n))\}$
  - $\mu\mathcal{F} = \mathcal{F}^0 \cup \mathcal{F}^1 \cup \mathcal{F}^2 \cup \mathcal{F}^3 \cup \ldots \cup \mathcal{F}^{2n} \cup \mathcal{F}^{2n+1} \ldots$
  - Pour tout $p \in \mathbb{N}$, $p \in \mathcal{F}^{2p} \cup \mathcal{F}^{2p+1}$ par construction de cette suite.

Summary

Treillis complete

☒ Definition

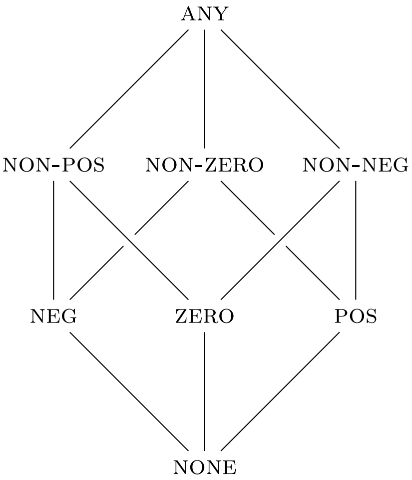
Un treillis complet $(L, \sqsubseteq)$ est un treillis satisfaisant les propriétés suivantes :

1. Pour toute partie A de L, il existe une borne supérieure notée $\sqcup A$.
2. Pour toute partie A de L, il existe une borne inférieure notée $\sqcap A$.

☼ Exemple

1. Un exemple simple est la structure $(\mathcal{P}(E), \subseteq, \varnothing, \cup, E, \cap)$ associée à l'ensemble des parties d'un ensemble E.
2. Treillis des signes est un treillis complet
   $\mathcal{S}igns =$
   $\{\text{NONE}, \text{ZERO}, \text{NON-ZERO}, \text{ANY}, \text{POS}, \text{NEG}, \text{NON-NEG}, \text{NON-POS}\}$
   $(\mathcal{S}igns, \sqsubseteq)$
3. Treillis des intervalles
   - $\mathbb{I}(\mathbb{Z}) = \{\bot\} \cup \{[l, u] | l \in \mathbb{Z} \cup \{-\infty\}, u \in \mathbb{Z} \cup \{\infty\}, l \leq u\}$
   - $[l_1, u_1] \sqsubseteq [l_2, u_2]$ si, et seulement si, $l2 \leq l1$ et $u_1 \leq u_2$.
   - $(\mathbb{I}(\mathbb{Z}), \sqsubseteq)$ est un treillis complet

Treillis des signes



- ANY désigne l'ensemble des entiers
- NON-POS désigne l'ensemble des entiers négatifs ou nuls
- POS désigne l'ensemble des entiers positifs non nuls.
- ZERO est l'ensemble contenant uniquement 0.

Treillis des intervalles

- $\mathbb{I}(\mathbb{Z}) = \{\bot\} \cup \{[l,u] | l \in \mathbb{Z} \cup \{-\infty\}, u \in \mathbb{Z} \cup \{\infty\}, l \leq u\}$
- $[l_1,u_1] \sqsubseteq [l_2,u_2]$ si, et seulement si, $l2 \leq l1$ et $u_1 \leq u_2$.
- $(\mathbb{I}(\mathbb{Z}), \sqsubseteq)$ est une structure partiellement ordonnée.
-
  1. $[l_1,u_1] \sqcup [l_2,u_2] = [min(l_1,l_2), max(u_1,u_2)]$
  2. $[l_1,u_1] \sqcap [l_2,u_2] = \begin{cases} [max(l_1,l_2), min(u_1,u_2)] \\ \bot, si\ max(l_1,l_2) > min(u_1,u_2) \end{cases}$
- $(\mathbb{I}(\mathbb{Z}), \sqcup)$ est un treillis complet.

. . . .

☒ Definition

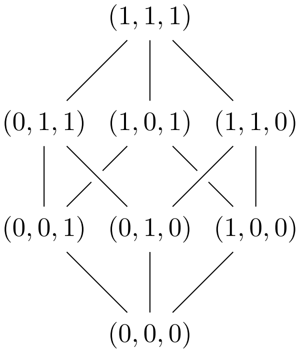
Un treillis complet $(L, \sqsubseteq, \bot, \sqcup, \top, \sqcap)$ est un treillis satisfaisant les propriétés suivantes :

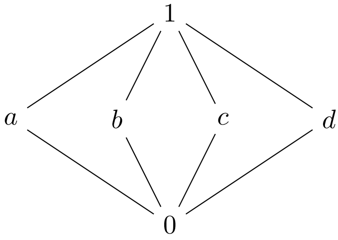
1. Pour toute partie A de L, il existe une borne supérieure notée $\sqcup A$.
2. Pour toute partie A de L, il existe une borne inférieure notée $\sqcap A$.

. . . .

- Un treillis complet $(L, \sqsubseteq,)$ peut être défini par les éléments suivants $(L, \sqsubseteq, \bot, \sqcup, \top, \sqcap)$
- Un treillis est une structure munie d'un ordre partiel et telle que deux éléments ont une borne supérieure et une inférieure dans le treillis : $(L, \sqsubseteq, \sqcup, \sqcap)$
  - $a \sqcup b$ existe et est la borne supérieure des deux éléments a et b.
  - $a \sqcap b$ existe et est la borne inférieure des deux éléments a et b.

Représentation par des diagrammes de Hasse





Summary

Pre/Post Points-fixes

Pour une fonction f définie sur un treillis $(L, \sqsubseteq, \perp, \sqcup, \top, \sqcap)$ et à valeurs dans $(L, \sqsubseteq, \perp, \sqcup, \top, \sqcap)$.

Définition

- on appelle pré-point-fixe de f, tout élément x de L satisfaisant la propriété $f(x) \sqsubseteq x$
- on appelle post-point-fixe de f, tout élément x de L satisfaisant $x \sqsubseteq f(x)$.

- $PostFIX(f) = \{x|x \in L \wedge x \sqsubseteq f(x)\}$: l'ensemble des post-points-fixes de f.
- $PreFIX(f) = \{x|x \in L \wedge f(x) \sqsubseteq x\}$: l'ensemble des pré-points-fixes de f.
- $FIX(f) = \{x|x \in L \wedge f(x) = x\}$: l'ensemble des points-fixes de f.

Théorème de Knaster-Tarski (I)

Soit f une fonction monotone croissante sur un treillis complet $(T, \bot, \top, \bigvee, \bigwedge)$. Alors il existe un plus petit point fixe et un plus grand point fixe pour f.

- μf désigne le plus petit point fixe de f.
- νf désigne le plus grand point fixe de f.
- μf vérifie les propriétés suivantes : $f(\mu f) = \mu f$ et $\forall x \in T. f(x) \sqsubseteq x \Rightarrow \mu f \sqsubseteq x$ (μf est la borne inférieure des prépoints fixes de f ou $\bigwedge(Pre(f))$).
- νf vérifie les propriétés suivantes : $f(\nu f) = \nu f$ et $\forall x \in T. x \sqsubseteq f(x) \Rightarrow x \sqsubseteq \nu f$ (νf est la borne supérieure des postpoints fixes de f ou $\bigvee(Post(f))$).

Preuve

Posons $y = \bigwedge\{x|f(x) \sqsubseteq x\}$ et montrons que y est un point fixe de f et que y est le plus petit point fixe de f.

1. $f(y) \sqsubseteq y$
   - Pour tout x de $\{x|f(x) \sqsubseteq x\}$, $y \sqsubseteq x$
   - $f(y) \sqsubseteq f(x)$ (par monotonie de f).
   - $f(x) \sqsubseteq x$ (par définition de x).
   - $f(y) \sqsubseteq x$ (par déduction).
   - $f(y) \sqsubseteq \bigwedge\{x|f(x) \sqsubseteq x\}$ (par définition de la borne inférieure, $f(y)$ est un minorant).
   - $f(y) \sqsubseteq y$
2. $y \sqsubseteq f(y)$
   - $f(y) \sqsubseteq y$ (par le cas 1)
   - $f(f(y)) \sqsubseteq f(y)$ (par monotonie de f)
   - $f(y) \in \{x|f(x) \sqsubseteq x\}$
   - $y \sqsubseteq f(y)$ (par définition de la borne inférieure)
3. Conclusion : $f(y) = y$ ou $y \in FIX(f)$.

Preuve

- $f(y) = y$ et z tel que $f(z) = z$
  - $f(z) = z$ (par hypothèse sur z)
  - $f(z) \sqsubseteq z$ (par affaiblissement de l'égalité)
  - $z \in \{x|f(x) \sqsubseteq x\}$ (par définition de cet ensemble)
  - $y \sqsubseteq z$ (par construction)
- y est le plus petit point fixe de f.

lfp(f) et gfp(f)

- $\mu f = lfp(f) = \bigwedge\{x|f(x) \sqsubseteq x\}$
- $\nu f = gfp(f) = \bigvee\{x|x \sqsubseteq f(x)\}$

- $lfp(f)$ signifie *least fixed-point*
- $gfp(f)$ signifie *greatest fixed-point*

Positionnement des éléments

$$
\begin{tikzcd}
\top \arrow[d, dash] \\
x \in \{x|f(x) \sqsubseteq x\} \arrow[d, dash] \\
\nu f \arrow[d, dash] \\
x \in \{x|x = f(x)\} \arrow[d, dash] \\
\mu f \arrow[d, dash] \\
x \in \{x|x \sqsubseteq f(x)\} \arrow[d, dash] \\
\bot
\end{tikzcd}
$$

- $\top = \sqcup\{x|f(x) \sqsubseteq x\}$
- $gfp(f) = \nu f = \sqcup\{x|x \sqsubseteq f(x)\}$
- $lfp(f) = \mu f = \sqcap\{x|f(x) \sqsubseteq x\}$
- $\bot = \sqcap\{x|x \sqsubseteq f(x)\}$

Preuve algébrique selon Hoare

- $x \in pre(f)$ **si, et seulement si,** $f(x) \leq x$ (définition)

Preuve algébrique selon Hoare

- $x \in pre(f)$ **si, et seulement si,** $f(x) \leq x$ (définition)
- $\forall a \in pre(f).\mu f \leq a$ (application de la définition de la borne inférieure d'un ensemble)

Preuve algébrique selon Hoare

- $x \in pre(f)$ **si, et seulement si,** $f(x) \leq x$ (définition)
- $\forall a \in pre(f).\mu f \leq a$ (application de la définition de la borne inférieure d'un ensemble)
- $\forall a \in pre(f).f(\mu f) \leq f(a) \leq a$ (croissance de f et propriété de a)

Preuve algébrique selon Hoare

- $x \in pre(f)$ **si, et seulement si**, $f(x) \leq x$ (définition)
- $\forall a \in pre(f).\mu f \leq a$ (application de la définition de la borne inférieure d'un ensemble)
- $\forall a \in pre(f).f(\mu f) \leq f(a) \leq a$ (croissance de f et propriété de a)
- $f(\mu f) \leq \mu f$ (application pour $a = \mu f$))

Preuve algébrique selon Hoare

- $x \in pre(f)$ **si, et seulement si,** $f(x) \leq x$ (définition)
- $\forall a \in pre(f).\mu f \leq a$ (application de la définition de la borne inférieure d'un ensemble)
- $\forall a \in pre(f).f(\mu f) \leq f(a) \leq a$ (croissance de f et propriété de a)
- $f(\mu f) \leq \mu f$ (application pour $a = \mu f$))
- $f(f(\mu f)) \leq f(\mu f)$ (croissance de f)

Preuve algébrique selon Hoare

- $x \in pre(f)$ **si, et seulement si,** $f(x) \leq x$ (définition)
- $\forall a \in pre(f).\mu f \leq a$ (application de la définition de la borne inférieure d'un ensemble)
- $\forall a \in pre(f).f(\mu f) \leq f(a) \leq a$ (croissance de f et propriété de a)
- $f(\mu f) \leq \mu f$ (application pour $a = \mu f$))
- $f(f(\mu f)) \leq f(\mu f)$ (croissance de f)
- $f(\mu f) \in pre(f)$ (définition des pré-points-fixes)

Preuve algébrique selon Hoare

- $x \in pre(f)$ **si, et seulement si,** $f(x) \leq x$ (définition)
- $\forall a \in pre(f).\mu f \leq a$ (application de la définition de la borne inférieure d'un ensemble)
- $\forall a \in pre(f).f(\mu f) \leq f(a) \leq a$ (croissance de f et propriété de a)
- $f(\mu f) \leq \mu f$ (application pour $a = \mu f$))
- $f(f(\mu f)) \leq f(\mu f)$ (croissance de f)
- $f(\mu f) \in pre(f)$ (définition des pré-points-fixes)
- $\mu f \leq f(\mu f)$ (définition de μf)
- $\mu f = f(\mu f)$ (définition de μf et propriété précédente)

Version constructive du théorème de Knaster-Tarski

Soit f une fonction monotone croissante sur un treillis complet $(T, \perp, \top, \bigvee, \bigwedge)$. Alors

1. La structure formée des points fixes de f sur T, $(fp(f), \sqsubseteq)$ est un treillis complet non-vide.
   $(fp(f) = \{x \in T : f(x) = x\}$
2. $lfp(f) \stackrel{def}{=} \bigvee_{\alpha} f^{\alpha}$ est le plus petit point fixe de f où :
$$\begin{cases} & f^0 & \stackrel{def}{=} & \perp \\ \alpha \text{ \textit{ordinal successeur}} & f^{\alpha+1} & \stackrel{def}{=} & f(f^{\alpha}) \\ \alpha \text{ \textit{ordinal limite}} & f^{\alpha} & \stackrel{def}{=} & \bigvee_{\beta<\alpha} f^{\beta} \end{cases}$$
3. $gfp(f) \stackrel{def}{=} \bigwedge_{\alpha} f_{\alpha}$ est le plus grand point fixe de f où
$$\begin{cases} & f_0 & \stackrel{def}{=} & \top \\ \alpha \text{ \textit{ordinal successeur}} & f_{\alpha+1} & \stackrel{def}{=} & f(f_{\alpha}) \\ \alpha \text{ \textit{ordinal limite}} & f_{\alpha} & \stackrel{def}{=} & \bigwedge_{\beta<\alpha} f_{\beta} \end{cases}$$

Summary

Summary

Lemme de Arden

- Soit l'ensemble des parties de $\Sigma^\star$ munies des opérations classiques des ensembles et l'opération de concaténation.
- La fonction $\mathcal{F}$ définie sur cet ensemble à valeur dans le même ensemble associe à tout ensemble un ensemble obtenu par application des opérations ensemblistes et de la concaténation : $\mathcal{F}(X) = A.X \cup B$
- $(\mathcal{P}(\Sigma^\star), \varnothing, \Sigma^\star, \cup, \cap)$ est un treillis complet.
- $\mathcal{F}$ admet un plus petit point fixe qui est le langage régulier caractérisé par l'expression régulière $a^\star b$

Summary

Fonctions induites par les grammaires

- Soit une grammaire algébrique $G=(N,T,P,S)$ définissant un langage sur T.
- On définit pour cette grammaire un opérateur sur $T^\star$ l'ensemble desmots finis sur T, noté $\mathcal{F}_G$ comme suit :
  - $N=\{X_1,\ldots,X_n\}$
  - $S=X_1$ et $X=(X_1,\ldots,X_n)$.
  - $\mathcal{F}_G(X)$ est construit par application des règles de P pour chaque non terminal X_i
  - Si $X_i \longrightarrow v_1+\ldots+v_p$, on obtient l'ensemble par remplaceme,nt des X_k dans les v_p.
- Exemple :
  - $N=\{X,Y,Z\}$
  - $T=\{a,b\}$
  - $P=\{X \longrightarrow aYZb, Y \longrightarrow aY, Z \longrightarrow Zb, Y \longrightarrow a, Z \longrightarrow b\}$
  - $F(X,Y,Z)=\left(\begin{array}{c} aYZb \\ aY\cup\{a\} \\ Zb\cup\{b\} \end{array}\right)$

- F est monotone croisssante sur $(\{X, Y, Z, a, b\}^\star)^3$.
- $\left(\begin{array}{l} X = aYZb \\ Y = aY\cup\{a\} \\ Z = Zb\cup\{b\} \end{array}\right)$
- μF est le plus petit point fixe.
- $L(G) = \pi_1^3(\mu F)$

Summary

- L'ensemble $\mathcal{EVEN}$ des nombres naturels pairs est le plus petit ensemble stable par application des règles suivantes :
  1. $0 \in \mathcal{EVEN}$
  2. Si $n \in \mathcal{EVEN}$, alors $n+2 \in \mathcal{EVEN}$
- Opérateur induit : $\mathcal{F}(X) = \{0\} \cup \{n+2 | n \in X\}$
- $\mathcal{F} \in \mathbb{P}(\mathbb{N}) \longrightarrow \mathbb{P}(\mathbb{N})$
- $\mathcal{F}(\mathcal{EVEN}) = \mathcal{EVEN}$ et $\mathcal{EVEN} = \mu\mathcal{F}$.

- Une définition inductive $\mathcal{I}$ est une structure $(U, \mathcal{F}, \perp, \sqcup)$ où
  - U est un ensemble appelé l'univers
  - $\mathcal{R}$ est un ensemble de règles de la forme $\mathbf{Si}\ P, \mathbf{alors}\ C$ où $P \subseteq U$ et $C \in U$.
  - $\perp \in U$
  - $\sqcup \in \mathbb{P}(\mathbb{P}(U)) \longrightarrow \mathbb{P}(U)$
- Pour une définition inductive, on peut dériver un opérateur induit $\mathcal{F}$: $\mathcal{F}(X) = \{c \in U | \exists P \subseteq X : \mathbf{Si}\ P, \mathbf{alors}\ C \in \mathcal{R}\}$
- $\mathcal{F}_1(X) = \{0\} \cup \{n+3 \in \mathbb{N} | n \in X\}$
- $\mathcal{F}_2(X) = Init \cup \{u \in U | \exists s. s \xrightarrow{\star} u \wedge s \in X\}$

Principe d'induction sur les naturels

$\forall A.A \subseteq \mathbb{N} \wedge 0 \in A \wedge suc[A] \subseteq A \Rightarrow \mathbb{N} \subseteq A)$

- $0 \in A$
- Pas d'induction
  - Hypothèse $n \in A$
  - Conclusion $suc(n) \in A$
- Conclusion $\forall n.n \in \mathbb{N} \Rightarrow n \in A$

Soit la propriété suivante à démontrer $\forall n.n \in \mathbb{N} \Rightarrow n \in P(n)$:

1. $A \stackrel{def}{=} \{n|n \in \mathbb{N} \wedge P(n)\}$
2.
   - $0 \in A$
   - Pas d'induction
     - Hypothèse $n \in A$
     - Conclusion $suc(n) \in A$
   - Conclusion $\forall n.n \in \mathbb{N} \Rightarrow n \in A$