



Cours MALG & MOVEX

MALG **Théorie du Point-Fixe et ses Applications**

Dominique Méry Telecom Nancy, Université de Lorraine (21 avril 2025 at 10:51 P.M.)

Année universitaire 2024-2025

- Transition Systems
 Overview of Transition Systems as Modelling Tool
 Expression of transition systems
- 2 Introduction of fixed-points
- 3 Structures Partiellement Ordonnées Inductives (CPO)
- 4 Point-fixe pour une fonction continue au sens de Scott
- 5 Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- Applications
 - Lemme de Arden Grammaires algébriques Définition inductive

- Transition Systems
 Overview of Transition Systems as Modelling Tool
 Expression of transition systems
- 2 Introduction of fixed-points
- 3 Structures Partiellement Ordonnées Inductives (CPO)
- 4 Point-fixe pour une fonction continue au sens de Scott
- Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- 7 Applications

- 1 Transition Systems
 Overview of Transition Systems as Modelling Tool
 Expression of transition systems
- 2 Introduction of fixed-points
- 3 Structures Partiellement Ordonnées Inductives (CPO)
- 4 Point-fixe pour une fonction continue au sens de Scott
- Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- Applications
 - Lemme de Arden Grammaires algébriques Définition inductive

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: $R = s(np) \in 0...s(cap) 1 \land s'(np) = s(np) + 1 \land s'(cap) = s(cap) \text{ is an expression of } R \text{ observing that one more person has entered the room.}$
- ▶ a trace $s_0 \xrightarrow{R_0} s_1 \xrightarrow{R_1} s_2 \xrightarrow{R_2} s_3 \xrightarrow{R} \dots \xrightarrow{R_{i-1}} s_i \xrightarrow{R_i} \dots$ is a trace generated by the different observations $R_0, \dots R_p, \dots$

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.

- ▶ 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)
- $ightharpoonup \varphi(s)$ (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 £ 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 $\Sigma \stackrel{def}{=} \operatorname{Var} \longrightarrow D$ 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 $s \in \hat{P}$.
- ightharpoonup A distinction is made between flexible variable symbols x and logical variable symbols v, and constant symbols c are used.

Interpretation of assertions

- ① [x](s) = s(x) = x : x is the value of the variable x in s.
- [x](s') = s'(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.
- $\P[\varphi(x) \wedge \psi(x)](s) = \llbracket \varphi(x) \rrbracket(s) \text{ et } \llbracket \psi(x) \rrbracket(s) \text{ where } \textit{and} \text{ is the classical interpretation of symbol } \wedge \text{ according to the truth table}.$



- ▶ [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 $\triangle T_F X$ and x is the math font of $\triangle T_F X$.
- Using the name of the variable x as its current value, i.e. x and [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.

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.
- x₀ 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{P} and, in this case, P(x) is true in x. is equivalent to $x \in \hat{P}$.

Examples of property

- ▶ $P_1(x) \stackrel{def}{=} x \in 18..22 : x$ is a value between 18 and 22 and $\hat{P}_1 = \{18, 19, 20, 21, 22\}.$
- ▶ $P_2(p) \stackrel{def}{=} p \subset PEOPLE \land card(p) \leq n : p$ is a set of persons and that set has at most n elements and $\hat{P}_2 = \{p_1 \dots p_n\}$. 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 $s_1 \cup \ldots \cup s_p \subseteq D$.

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

- $ightharpoonup ax1(fred \in PEOPLE)$: fred is a person from the set PEOPLE
- $\blacktriangleright ax2(suc \in \mathbb{N} \to \mathbb{N} \land (!i.i \in \mathbb{N} \Rightarrow suc(i) = i+1))$: The function suc is the total function which associates any natural i with its SUccessor, Successor
- $\blacktriangleright ax3(\forall A.A \subseteq \mathbb{N} \land 0 \in \mathbb{N} \land suc[A] \subseteq A \Rightarrow \mathbb{N} \rightarrow \subseteq A)$: 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 $(s,s'\in \mathcal{V}ar(S)\longrightarrow \mathrm{VALS})$. $s\longrightarrow_R s'$ will be written as a relation $\mathrm{R}(x,x')$ where x and x' designate values of 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

.....

☑ 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

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

where

- lack AX(s,c) is an axiomatic theory defining the sets, constants and static properties of these elements.
- ightharpoonup 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

$$EM(s, c, x, \text{VALS}, Init(x)\{e_0, \dots, e_n\}) = (AX(s, c), x, \text{VALS}, Init(x), \{e_0, \dots, e_n\}).$$

Safety property

- Next(x, x') or $Next(s, c, x, x') \stackrel{def}{=} BA(e_0)(s, c, x, x') \lor \ldots \lor BA(e_n)(s, c, x, x').$
- $\begin{array}{l} \blacktriangleright \text{ the transitive reflexive closure of the relation } Next^{\star}(s,c,x_{0},x) \stackrel{def}{=} \\ \left\{ \begin{array}{l} \lor x = x_{0} \\ \lor Next(s,c,x_{0},x) \\ \lor \exists \ xi \in \text{VALs.} Next^{\star}(s,c,x_{0},xi) \land Next(s,c,xi,x) \end{array} \right. \end{array}$

□ Definition(safety property)

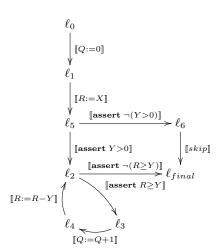
A property P(x) is a safety property for the system S, if

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

.....

- 1 Transition Systems
 Overview of Transition Systems as Modelling Too
 Expression of transition systems
- 2 Introduction of fixed-points
- 3 Structures Partiellement Ordonnées Inductives (CPO)
- 4 Point-fixe pour une fonction continue au sens de Scott
- 5 Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- 7 Applications
 - Lemme de Arden
 Grammaires algébriques
 Définition inductive

Programme as flowchart



Safety Properties and Invariance Properties for a Relational Model

- D C :: / .: .)

□ Definition(assertion)

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

$$\forall x_0, x \in \text{Vals.} Init(x_0) \land \text{Next}^*(x_0, x) \Rightarrow A(x).$$

.....

□ Definition(relation)

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

$$\forall x_0, x \in \text{Vals.} Init(x_0) \land \text{Next}^*(x_0, x) \Rightarrow R(x_0, x).$$

.....

- Transition Systems
- 2 Introduction of fixed-points
- 3 Structures Partiellement Ordonnées Inductives (CPO)
- Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- Applications

Invariant for checking a safety property

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

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

- ▶ $\forall x_0, x \in \text{VALS}.Init(x_0) \land \text{NEXT}^*(x_0, x) \Rightarrow A(x).$ if, and only if,
- ▶ REACHABLE(S) $\subseteq \{u|u \in \text{VALS} \land A(u)\}$

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

- ▶ $\{u|u \in VALS \land Init(u)\} \subseteq REACHABLE(S)$
- ▶ Next[reachable(S)] \subseteq reachable(S)

$$\forall U. \left(\begin{array}{c} U \subseteq \text{Vals} \\ \{u | u \in \text{Vals} \land Init(u)\} \subseteq U \\ \land \\ \text{Next}[U] \subseteq U \end{array} \right) \Rightarrow \text{Reachable}(\mathcal{S}) \subseteq U$$

Principle for exhaustive checking

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

- ▶ $\forall x_0, x \in \text{VALS}.Init(x_0) \land \text{NEXT}^*(x_0, x) \Rightarrow A(x).$ if, and only if,
- ▶ REACHABLE(S) $\subseteq \{u|u \in \text{VALS} \land A(u)\}$
- ▶ REACHABLE(S) = { $u|u \in Vals \land Init(u)$ } $\cup Next[Reachable(S)]$
- $ightharpoonup F \in \mathcal{P}(VALS) \longrightarrow \mathcal{P}(VALS)$
- $F(U) = \{u | u \in VALS \land Init(u)\} \cup NEXT[U]$
- ► F(REACHABLE(S)) = REACHABLE(S)

Pfroblems related to fixex-point theory

▶ Solving equations as $X = \mathcal{F}(X)$

Théorie du Point-Fixe et ses Applications (21 avril 2025) (Dominique Méry)

Finding and even better computing solutions when they exist.

Fixed-point theorem 1

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 $\ell.$ 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 :

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

Fixed-point theorem 2: inductive structure

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

```
#include <stdio.h>
int f (int x, int y)
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));
```

Fixed-point theorem 4 Definition of a function

```
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

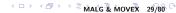
 $\forall (x,y) \in nat \times nat : (x,y) \in Domaine(A)$

ou

$$\forall (x,y) \in nat \times nat : P(x,y) \text{ avec}$$

 $P(x,y) \stackrel{def}{=} ((x,y) \in domaine(A)).$





Fixed-point theorem 4 Definition of a function

Méthode

- **1** Choose a well founded structure $nat : (nat \times nat, <_{lex})$
- 2 Use the principle of induction for the well founded structure

 - **2** $\forall x \in nat : x \neq 0 : P(x, 0)$
 - $\exists \ \forall (x,y) \in nat \times nat : x \neq 0 : y \neq 0 : P(x,y)$

Fixed-point theorem 5 Definition of a function

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

- Transition Systems
- 2 Introduction of fixed-points
- 3 Structures Partiellement Ordonnées Inductives (CPO)
- Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- Applications

✓Structure partiellement ordonnée

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

Relation d'ordre partiel □

- ightharpoonup $\sqsubset \subset X \times X$
- \blacktriangleright $\{(a,a)|a\in X\}\subseteq\sqsubseteq$ (réflexivité)
- $\forall (a,b).((a\in X\wedge b\in X\wedge (a\sqsubseteq b)\wedge (b\sqsubseteq a))\Rightarrow (a=b))$ (anti-symétrie)
- \blacktriangleright (\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

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

Structure partiellement ordonnée inductive (ou CPO)

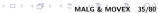
Une structure partiellement ordonnée inductive (ou un CPO Complete Partially Ordered Set) est une structure partiellement ordonnée (D,\sqsubseteq) telle que :

- **1** D admet un plus petit élément noté $\bot_D : \forall d \in D. \bot_D \sqsubseteq d$.
- 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 :

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



Caractérisation d'une structure partiellement ordonnée inductive

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 $\bot_D : \forall d \in D. \bot_D \sqsubseteq d$.
- 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 \bot un élément qui n'est pas élément de D. On définit l'ordre partiel sur $D \cup \{\bot\} \sqsubseteq$ comme suit :

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

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

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

Ensemble des fonctions partielles $A \rightarrow B$

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

 $(A \to B,\sqsubseteq)$ est une structure partiellement ordonnée inductive où $\bot_{A \to B}$ 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.

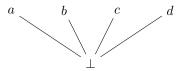
✓ Notations

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

- \blacktriangleright x est un majorant de D, si $\forall d \in D : d \sqsubseteq x$
- \blacktriangleright 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.
- $ightharpoonup x \sqcup y$ est la borne supérieure de $\{x,y\}$, lorsqu'elle existe.
- $ightharpoonup x \sqcap y$ est la borne inférieure de $\{x,y\}$, lorsqu'elle existe.
- \triangleright $Sup(D) = \sqcup D = \vee D$: borne supérieure de D quand elle existe.
- $ightharpoonup Min(D) = \Box D = \wedge D$: borne inférieure de D quand elle existe.

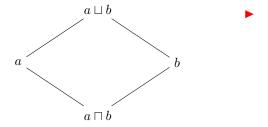


✓Illustration des notations

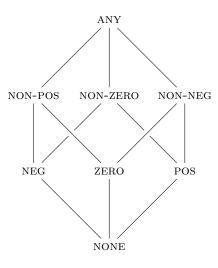


$$D = \{a, b, c, d\}$$

$$\begin{array}{c} \blacktriangleright \ D = \{a,b,c,d\} \\ \blacktriangleright \ D^\perp = \{a,b,c,d\} \end{array}$$



✓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.

Croissance d'une fonction

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

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

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



Structures mathématiques (continuité d'une fonction)

- ▶ 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

- Transition Systems
- 2 Introduction of fixed-points
- 3 Structures Partiellement Ordonnées Inductives (CPO)
- 4 Point-fixe pour une fonction continue au sens de Scott
- Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- Applications

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 .

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

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

- $x \stackrel{def}{=} \bigvee_{i \geq 0} f^i$ où $\forall i \in nat^* : f^{i+1} = f(f^i)$ et $f^0 = \bot_D : x$ existe car (D, \sqsubseteq) est une structure inductive.
- ightharpoonup x est un point-fixe de f :

$$\begin{array}{l} f(x) \stackrel{\textit{definition}}{=} f(\bigvee_{i \geq 0} f^i) \stackrel{\textit{continuite}}{=} \bigvee_{i \geq 0} f(f^i) \\ \bigvee_{i \geq 0} f(f^i) \stackrel{\textit{definition}}{=} \stackrel{\textit{de la suite}}{=} \bigvee_{i \geq 0} f^{i+1} \stackrel{\textit{renommage}}{=} \\ \bigvee_{j > 0} f^j \stackrel{\textit{propriete}}{=} \stackrel{\textit{de } \bot_D}{=} \bot_D \ \sqcup \ \bigvee_{j > 0} f^j = x \end{array}$$

- x est le plus petit point-fixe de F. Soit y un autre point-fixe de f :
 - $\perp_D \sqsubseteq y$
 - Si $\forall j \in \{0, \dots, i-1\} : f^j \sqsubseteq y$, alors $f^i \sqsubseteq y$.
 - $\forall j \in nat : f^j \sqsubseteq y$

D'où
$$x \sqsubseteq y$$
.

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

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

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

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

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

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

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

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

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



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

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

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

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

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

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

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

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

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

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



- ▶ 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

 - **2** Soit $d \in D$ tel que P(d):

Alors $P(\mu F)$.

Prédicats admissibles

- ightharpoonup P(x) défini par x=a est admissible.
- ightharpoonup 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

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



Fonction 91 de MacCarthy

- $F(f)(x) = si \ x > 100 \ alors \ x-10 \ sinon \ f(f(x+11)) \ fi$
- 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.



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

- $ightharpoonup g(x) = \mathbf{if} \ odd(x) \ \mathbf{then} \ 2 \ \mathbf{else} \ 0 \ \mathbf{fi} \ \mathbf{est} \ \mathbf{solution} \ \mathbf{de} \ \mathbf{cette} \ \mathbf{\acute{e}quation}$
- ▶ Utilisation de $P(f) \stackrel{def}{=} f \sqsubseteq g$
- ▶ Montrer que $dom(\mu f) = \mathbb{N}$:
 - $\mathcal{F}^{2n} = \{(p, v_p) | 0 \le p < n \land v_p = g(p)\} \cup \{(p, v_p) | 0 > p \ge -(n-1) \land v_p = g(p)\} \cup \{(n, g(n))\}$
 - $\mathcal{F}^{2n+1} = \{(p, v_p) | 0 \le p < n \land v_p = g(p)\} \cup \{(p, v_p) | 0 > p \ge -(n-1) \land v_p = g(p)\} \cup \{(n, g(n)), (-n, g(-n))\}$
 - $\mu \mathcal{F} = \mathcal{F}^0 \cup \mathcal{F}^1 \cup \mathcal{F}^2 \cup \mathcal{F}^3 \cup \dots \cup \mathcal{F}^{2n} \cup \mathcal{F}^{2n+1} \dots$
 - Pour tout $p \in \mathbb{N}$, $p \in \mathcal{F}^{2p} \cup \mathcal{F}^{2p+1}$ par construction de cette suite.



Summary

- Transition Systems
- 3 Structures Partiellement Ordonnées Inductives (CPO)
- 6 Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- Applications

Treillis complet

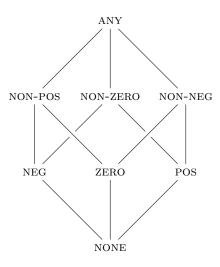
□ Definition

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 $\Box A$.

- **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 Sians ={NONE, ZERO, NON-ZERO, ANY, POS, NEG, NON-NEG, NON-POS} $(Signs, \Box)$
- Treillis des intervalles
 - $\mathbb{I}(\mathbb{Z}) = \{\bot\} \cup \{[l, u] | l \in \mathbb{Z} \cup \{-\infty\}, u \in \mathbb{Z} \cup \{\infty\}, l < u\}$
 - $[l_1, u_1] \sqsubset [l_2, u_2]$ si, et seulement si, $l_2 < l_1$ et $u_1 < u_2$.
 - $(\mathbb{I}(\mathbb{Z}), \sqsubseteq)$ 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

- $\blacktriangleright \ \mathbb{I}(\mathbb{Z}) = \{\bot\} \cup \{[l,u] | l \in \mathbb{Z} \cup \{-\infty\}, u \in \mathbb{Z} \cup \{\infty\}, l \leq u\}$
- $ightharpoonup [l_1,u_1] \sqsubseteq [l_2,u_2]$ si, et seulement si, $l2 \le l1$ et $u_1 \le u_2$.
- $ightharpoonup (\mathbb{I}(\mathbb{Z}), \sqsubseteq)$ est une structure partiellement ordonnée.
- $\begin{array}{c} \bullet & \bullet & [l_1,u_1] \sqcup [l_2,u_2] = [min(l_1,l_2),max(u_1,u_2)] \\ \bullet & [l_1,u_1] \sqcap [l_2,u_2] = \left\{ \begin{array}{c} [max(l_1,l_2),min(u_1,u_2)] \\ \perp, si \; max(l_1,l_2) > min(u_1,u_2) \end{array} \right. \end{array}$
- $ightharpoonup (\mathbb{I}(\mathbb{Z}), \sqcup)$ est un treillis complet.

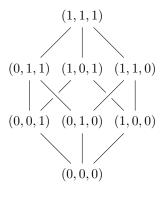
Treillis complet

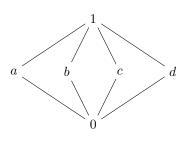
□ Definition

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 $\Box A$.
- ▶ 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

- Transition Systems
- 2 Introduction of fixed-points
- 3 Structures Partiellement Ordonnées Inductives (CPO)
- Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- Applications

Pre/Post Points-fixes

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

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

Théorème du point fixe pour les treillis complets

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.

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

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.

- - 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).
 - $f(y) \sqsubseteq y$
- $y \sqsubseteq f(y)$
 - $f(y) \sqsubseteq y$ (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

- ightharpoonup 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)
- ightharpoonup y est le plus petit point fixe de f.

Ifp(f) et gfp(f)

- $\blacktriangleright \mu f = lfp(f) = \bigwedge \{x | f(x) \sqsubseteq x\}$
- $\triangleright \ \nu f = gfp(f) = \bigvee \{x | x \sqsubseteq f(x)\}$
- ightharpoonup lfp(f) signifie least fixed-point
- ightharpoonup gfp(f) signifie greatest fixed-point



Positionnement des éléments

$$x \in \{x | f(x) \sqsubseteq x\}$$

$$|$$

$$\nu f$$

$$|$$

$$x \in \{x | x = f(x)\}$$

$$|$$

$$\mu f$$

$$|$$

$$x \in \{x | x \sqsubseteq f(x)\}$$

$$|$$

- $\vdash \top = \sqcup \{x | f(x) \sqsubseteq x\}$
- $ightharpoonup qfp(f) = \nu f = \sqcup \{x | x \sqsubseteq f(x)\}$
- $ightharpoonup \perp = \sqcap \{x | x \sqsubseteq f(x)\}$

 $ightharpoonup x \in pre(f)$ si, et seulement si, $f(x) \le x$ (définition)

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

- $ightharpoonup x \in pre(f)$ 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)
- $ightharpoonup orall a \in pre(f).f(\mu f) \leq f(a) \leq a$ (croissance de f et propriété de a)

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

- $x \in pre(f)$ si, et seulement si, $f(x) \le x$ (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)
- ▶ $f(\mu f) \le \mu f$ (application pour $a = \mu f$))
- $ightharpoonup f(f(\mu f)) \le f(\mu f)$ (croissance de f)

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



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



Version constructive du théorème de Knaster-Tarski

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

 $\textbf{0} \ \ \text{La structure formée des points fixes de } f \ \text{sur } T, \ (fp(f),\sqsubseteq) \ \text{est un treillis complet non-vide}.$

$$(fp(f) = \{x \in T : f(x) = x\}$$

2 $lfp(f)\stackrel{def}{=}\bigvee_{\alpha}f^{\alpha}$ est le plus petit point fixe de f où :

$$\begin{cases} f^0 & \stackrel{def}{=} \ \bot \\ \alpha \text{ ordinal successeur } f^{\alpha+1} & \stackrel{def}{=} \ f(f^{\alpha}) \\ \alpha \text{ ordinal limite } f^{\alpha} & \stackrel{def}{=} \ \bigvee_{\beta < \alpha} f^{\beta} \end{cases}$$

3 $gfp(f)\stackrel{def}{=} \bigwedge_{\alpha} f_{\alpha}$ est le plus grand point fixe de f où

$$\begin{cases} f_0 & \stackrel{def}{=} & \top \\ \alpha \text{ ordinal successeur } f_{\alpha+1} & \stackrel{def}{=} & f(f_{\alpha}) \\ \alpha \text{ ordinal limite } f_{\alpha} & \stackrel{def}{=} & \bigwedge_{\beta < \alpha} f_{\beta} \end{cases}$$

Summary

- Transition Systems
- 2 Introduction of fixed-points
- 3 Structures Partiellement Ordonnées Inductives (CPO)

- 6 Théorèmes du point-fixe de Knaster-Tarski
- Applications Lemme de Arden
 - Grammaires algébriques
 - Définition inductive

Summary

- Transition Systems

- 6 Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- 7 Applications

Lemme de Arden

Lemme de Arden

- ▶ Soit l'ensemble des parties de Σ^* 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 : $\mathcal F(X) = A.X \cup B$
- \blacktriangleright $(\mathcal{P}(\Sigma^*), \varnothing, \Sigma^*, \cup, \cap)$ est un treillis complet.
- $ightharpoonup \mathcal{F}$ admet un plus petit point fixe qui est le langage régulier caractérisé par l'expression régulière $a^{\star}b$



Summary

- Transition Systems

- 6 Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- Applications

Grammaires algébriques

Fonctions induites par les grammaires

- ightharpoonup Soit une grammaire algébrique G=(N,T,P,S) définissant un langage sur T.
- On définit pour cette grammaire un opérateur sur T^* l'ensemble desmots finis sur T, noté \mathcal{F}_G comme suit :
 - $N = \{X_1, \dots, X_n\}$
 - $S = X_1$ et $X = (X_1, \dots, X_n)$.
 - \(\mathcal{F}_G(X) \) 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 :
 - $N = \{X, Y, Z\}$
 - $T = \{a, b\}$
 - $\bullet \ P = \{X \longrightarrow aYZb, Y \longrightarrow aY, Z \longrightarrow Zb, Y \longrightarrow a, Z \longrightarrow b\}$
 - $F(X, Y, Z) = \begin{pmatrix} aYZb \\ aY \cup \{a\} \\ Zb \cup \{b\} \end{pmatrix}$



Fonctions induites par les grammaires

 $\blacktriangleright \ F \ {\rm est \ monotone \ croisssante \ sur \ } (\{X,Y,Z,a,b\}^{\star})^3.$

$$\qquad \qquad \left(\begin{array}{c} X = aYZb \\ Y = aY \cup \{a\} \\ Z = Zb \cup \{b\} \end{array} \right)$$

- ightharpoonup μF est le plus petit point fixe.
- $L(G) = \pi_1^3(\mu F)$

Summary

- Transition Systems

- 6 Treillis
- 6 Théorèmes du point-fixe de Knaster-Tarski
- Applications

Définition inductive

Définition inductive

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

Définition inductive

- ▶ Une définition inductive \mathcal{I} est une structure $(U, \mathcal{F}, \bot, \sqcup)$ 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 $C\in U.$
 - $\bot \in U$
 - $\sqcup \in \mathbb{P}(\mathbb{P}(U)) \longrightarrow \mathbb{P}(U)$
- Pour une définition inductive, on peut dériver un opérateur induit \mathcal{F} : $\mathcal{F}(X) = \{c \in U | \exists P \subseteq X : \mathbf{Si} \ P, \mathbf{alors} \ C \in \mathcal{R}\}$
- ▶ $\mathcal{F}_1(X) = \{0\} \cup \{n+3 \in \mathbb{N} | n \in X\}$
- $F_2(X) = Init \cup \{u \in U | \exists s.s \xrightarrow{\star} u \land s \in X\}$

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

- $ightharpoonup 0 \in A$
- ► 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 $\forall n.n \in \mathbb{N} \Rightarrow n \in P(n)$:

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