Démonstration Automatique
Métathéorèmes
Connaissances déclaratives

Dominique Pastre

Journée en hommage à Jacques Pitrat
6 mars 2020

- Démonstration automatique, Raisonnement mathématique

- 1966 thèse de Jacques Pitrat, (méta)théorèmes
- 1972 séminaire + cours DEA
  équipe Pitrat + Laurière, thèses en DAT et calcul symbolique

- PROVER(Bledsoe), DATTE et Muscadet
  connaissances déclaratives ?

- objectifs

- et maintenant ?

1966

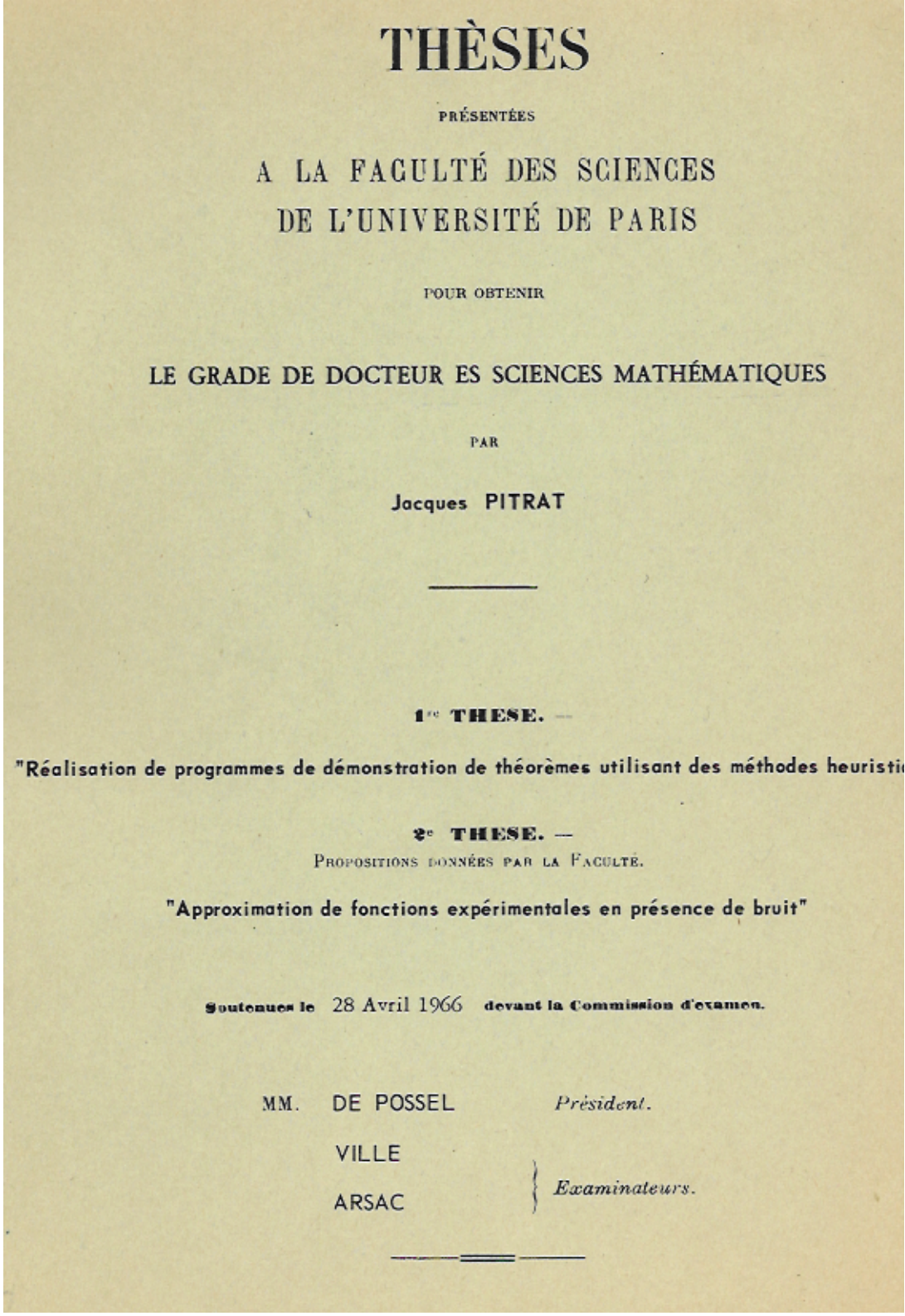
Jacques PITRAT

1ère THESE

Réalisation de programmes de démonstration de théorèmes utilisant des méthodes heuristiques

28 Avril 1966

MM DE POSSEL Président
VILLE
ARSAC Examinateurs



THÈSES

PRÉSENTÉES

A LA FACULTÉ DES SCIENCES
DE L'UNIVERSITÉ DE PARIS

POUR OBTENIR

LE GRADE DE DOCTEUR ES SCIENCES MATHÉMATIQUES

PAR

Jacques PITRAT

1re THESE. —

"Réalisation de programmes de démonstration de théorèmes utilisant des méthodes heuristi

2e THESE. —

PROPOSITIONS DONNÉES PAR LA FACULTÉ.

"Approximation de fonctions expérimentales en présence de bruit"

Soutenues le 28 Avril 1966 devant la Commission d'examen.

MM. DE POSSEL *Président.*
VILLE
ARSAC } *Examinateurs.*

Systèmes formels pour la logique des propositions

- Axiomes (théorèmes par définition)
- Règles de déduction (productions)
- on déduit de nouveaux théorèmes en appliquant les règles de déduction à des théorèmes connus

But du travail de Jacques Pitrat : écrire un programme <u>général</u> qui découvre et/ou démontre des théorèmes dans diverses axiomatiques de la logique des propositions

Exemple d'axiomatique (Lukasiewicz)

- Axiomes : $\supset\supset p \supset q\ r \supset\supset p\ q \supset p\ r$

  $\supset p \supset q\ p$

  $\supset\supset \sim p \sim q \supset q\ p$
- Règles : Productions $p, \supset p\ q \Rightarrow q$

  + Substitutions

- Axiomes : ⊃⊃ p ⊃ q r ⊃⊃ p q ⊃ p r
  ⊃ p ⊃ q p
  ⊃⊃ ~p ~q ⊃ q p

  $(p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r))$

  $p \supset (q \supset p))$

  $(\sim p \supset \sim q) \supset (q \supset p)$

- Règles : Productions p, ⊃ p q ⇒ q

  $p, p \supset q \Rightarrow q$

  + Substitutions

Découverte et preuve d'un nouveau théorème

A1 $(p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r))$ donne le théorème T $(p \supset (q \supset p)) \supset ((p \supset q) \supset (p \supset p))$

M1 $p, p \supset q \Rightarrow q$ donne le métathéorème

M2 $p \supset (q \supset p), \underline{p \supset (q \supset p)} \underline{\supset ((p \supset q) \supset (q \supset p))}$

$\Rightarrow (p \supset q) \supset (p \supset p)$

c'est-à-dire

A2, $\underline{T} \Rightarrow (p \supset q) \supset (p \supset p)$

donc **$(p \supset q) \supset (p \supset p)$ est un théorème**

Jacques Pitrat découvre, étudie et utilise un algorithme (*) qu'il appelle **algorithme de dérivation** et qui et très proche de l'**algorithme d'unification** trouvé indépendamment par Robinson, et qui donne les substitutions les plus générales.

(*) qu'il n'avait pas le droit de publier avant d'avoir soutenu sa thèse

Recherche de nouvelles productions - Métathéorèmes

On aimerait bien avoir d'autres productions, en particulier de la forme $A \Rightarrow B$ qu'on pourrait appliquer à n'importe quel théorème T pour donner d'autres théorèmes.

Exemple : Supposons que $p \supset (q \supset r)$ soit un théorème

A1 $(p1 \supset (q1 \supset r1)) \supset ((p1 \supset q1) \supset (p1 \supset r1))$

+T $(p \supset (q \supset r)$ donne T1 $(p \supset q) \supset (p \supset r)$

A2 $p2 \supset (q2 \supset p2)$

+T1 $(p \supset q) \supset (p \supset r)$ donne T2 $q2 \supset ((p \supset q) \supset (p \supset r)$

A1 $(p3 \supset (q3 \supset r3)) \supset ((p3 \supset q3) \supset (p3 \supset r3))$

+T2 $q2 \supset ((p \supset q) \supset (p \supset r))$ donne T3 $(q2 \supset (p \supset q)) \supset (q2 \supset (p \supset r))$

T3 $(q2 \supset (p \supset q)) \supset (q \supset (p \supset r))$

+A2 $p4 \supset (q4 \supset p4)$ donne T4 $q \supset (p \supset r)$

d'où le métathéorème $p \supset (q \supset r) \Rightarrow q \supset (p \supset r)$

intéressant mais pas général

Métathéorèmes

- on a vu $p \supset (q \supset r) \Rightarrow q \supset (p \supset r)$
- on a $p, p \supset q \Rightarrow q$
- on a aussi $p \supset q, q \supset r \Rightarrow p \supset r$

On peut en trouver d'autres, mais ils ne sont pas généraux.

Il faut que ce soit le programme qui les découvre avec … des métamétathéorèmes

Métamétathéorèmes

généraux (Il y en a 8)

$a \Rightarrow b \; ; \; b \Rightarrow c \rightarrow a \Rightarrow c$

$a, b \Rightarrow c \rightarrow b, a \Rightarrow c$

$a \; ; \; a \Rightarrow b \rightarrow b$

...

spécifiques à une axiomatique

$(p \supset q) \rightarrow p \Rightarrow q$

$p \Rightarrow (q \supset r) \rightarrow p, q \Rightarrow r$

$(p \supset (q \supset r)) \rightarrow ((p \supset q) \Rightarrow (p \supset r))$

Métamétamétathéorèmes

5 métamétamétathéorèmes tous généraux

$a, b \Rightarrow c \rightsquigarrow a \rightarrow b \Rightarrow c$

$a, b \Rightarrow c \rightsquigarrow p \Rightarrow a \rightarrow p, b \Rightarrow c$

...

qui génèrent des métamétathéorèmes qui …

Pour décider ou non de garder un nouveau (méta)^{n}théorème, son intérêt est évalué (sa structure, le nombre de connectives, intérêts de ce qui précède (et des conséquences précédentes), ...).

Résultats

- M1 + MMM1 donne $(p \supset q) \rightarrow p \Rightarrow q$ MM1
- M1 + MMM2 $p \Rightarrow (q \supset r) \rightarrow p, q \Rightarrow r$ MM2
- T2 + MM1 $p \Rightarrow (q \supset p)$ M3

  ...
- ... $p \supset (q \supset r) \Rightarrow q \supset (p \supset r)$ M8
- ... $p \supset p$ T9
- ... $\sim\sim p \supset p$ T13
- ... $p \supset \sim\sim p$ T14
- ... $(\sim p \supset q) \supset (\sim q \supset p)$ T17
- ... $(p \supset q) \supset (\sim q \supset \sim p)$ T24

1972

Séminaire d'Intelligence artificielle du 7 janvier

- …
- logique et démonstration de théorèmes
- Principe de Résolution + une démonstration détaillée (théorie des groupes)
- il y a 2 familles de méthodes :

  - s'inspirer de ce que fait le mathématicien vs **principe de résolution**

  - avantages du principe de résolution : généralité, programmes simples, résultats théoriques

  - inconvénients : pas constructif, formalisation lourde, trop général donc pas performant, beaucoup de papiers mais peu de programmes
- théorie des ensembles (**Bledsoe**), très proche de ce que fait le mathématicien

DEA
thèses avec J.Pitrat

Dernier cours du DEA

- heuristiques pour choisir un sujet de thèse

Les premières thèses encadrées

- 8 thèses de **démonstration automatique et/ou résolution de problèmes mathématiques** parmi les 14 premières thèses (jusqu'en 1978)
- 16 au total (jusqu'en 1994)

- 1970, trigonométrie, J-L.Delaye
- 1972, calcul de limites, J-P.Laurent
- 1973, identité, raisonnement par récurrence, M.Vivet
- 1974, équations trigonométriques, M.GrandBastien
- 1974, arithmétique, R.Dallard
- 1975, exercices d'algèbre, A.Durand
- 1975, logique des prédicats du premier ordre avec égalité, A.Hertz
- 1975, **constructions géométriques, M.Buthion**
- 1976, **théorie des ensembles, D.Pastre**
- 1976, problèmes combinatoires, J-L.Laurière (thèse d'état)
- 1978, équations d'arithmétique, D.Bourgoin
- 1978, **représentation des ensembles, Topologie, B.Mérialdo**
- 1980, théorie des groupes, M.Gillet
- 1984, **(méta)connaissances en mathématiques, D.Pastre** (thèse d'état)
- 1984, **expertise pour le calcul formel, M.Vivet** (thèse d'état)
- + 1982, manipulation formelle d'expressions, M.Baron (avec JLL)

PROVER (Bledsoe)
DATTE et Muscadet

De PROVER et DATTE à Muscadet
règles et métarègles

- 2 programmes
  - PROVER (Bledsoe), sequents, découpages, réécritures, puis Principe de Résolution
  - DATTE, représentation du théorème à démontrer,

    règles construites par un programme
- 4 versions du système à base de connaissances Muscadet
  - faits, règles et métarègles
  - bases de connaissances générales

    + domaines particuliers,
  - version tout à fait générale (TPTP)

PROVER (Bledsoe)

découpages et réécritures (SPLIT et REDUCE)

INPUT	*général*	OUTPUT
• $A \wedge B$		• deux théorèmes A et B
$\forall x\ P(x)$		$P(x)$
$p \rightarrow (A \wedge B)$		$(p \rightarrow A) \wedge (p \rightarrow B)$
$p \vee q \rightarrow A$		$(p \rightarrow A) \wedge (q \rightarrow A)$
$p \rightarrow (A \rightarrow B)$		$p \wedge A \rightarrow B$
$\forall x\ P(x)$		$P(x)$
$A \rightarrow \forall x\ P(x)$		$A \rightarrow P(y)$ (nouvelle variable)
$\exists x\ P(x) \rightarrow D$		$P(y) \rightarrow D$ "
• $\neg (A \wedge B)$		• $\neg A \vee \neg B$
...		...

PROVER suite

- *pour la théorie des ensembles*

INPUT	OUTPUT
$s \in A \cap B$	$s \in A \wedge s \in B$
$s \in A \cup B$	$s \in A \vee s \in B$
$s \in SB(A)$	$s \subset A$
$C \subset A \cap B$	$C \subset A \wedge C \subset B$
$A \cup B \subset C$	$A \subset B \wedge A \subset C$

- *programmé* (CYCLE)

$A \subset B$	$t \in A \rightarrow t \in B$
$p \rightarrow A \subset B$	$p \rightarrow (t \in A \rightarrow t \in B)$

- *dans une sous-formule*

$A \subset B$	$\forall t\ (t \in A \rightarrow t \in B)$

Exemple de démonstration par PROVER

$$SB(A) \cap SB(B) = SB(A \cap B)$$

Remplacement de l'égalité

$$SB(A) \cap SB(B) \subset SB(A \cap B) \wedge SB(A \cap B) \subset SB(A) \cap SB(B)$$

Découpage

(1) $SB(A) \cap SB(B) \subset SB(A \cap B)$

$t \in SB(A) \cap SB(B) \rightarrow t \in SB(A \cap B)$

$t \in SB(A) \wedge t \in SB(B) \rightarrow t \in SB(A \cap B)$

$t \subset A \wedge t \subset B \rightarrow t \subset A \cap B$

$t \subset A \wedge t \subset B \rightarrow t \subset A \wedge t \subset B$

true

(2) $SB(A \cap B) \subset SB(A) \cap SB(B)$

.

.

.

.

true

true

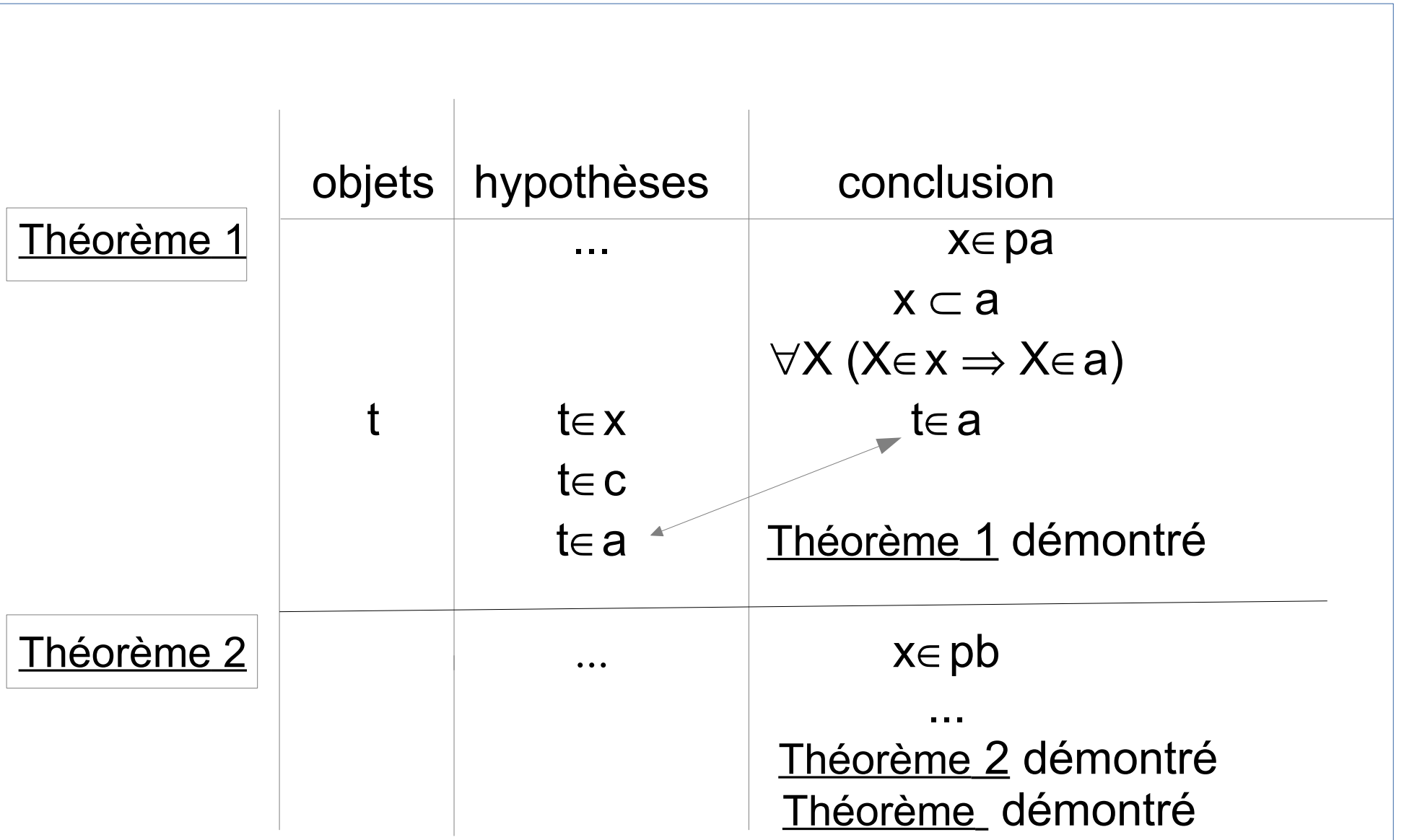
Démonstration par DATTE et Muscadet du théorème $\forall A \forall B(\mathcal{P}(A \cap B) \subset \mathcal{P}(A) \cap \mathcal{P}(B))$

objets	hypothèses	conclusion
		$\forall A \forall B(\mathcal{P}(A \cap B) \subset \mathcal{P}(A) \cap \mathcal{P}(B))$
	$\forall A \forall B \forall C \forall PA\ \forall PB \forall PC \forall PD(C{:}A \cap B \wedge PA{:}\mathcal{P}(A) \wedge PB{:}\mathcal{P}(B) \wedge PC{:}\mathcal{P}(C) \wedge PD{:}PA \cap PB \Rightarrow PC \subset PD)$	
a, b, c, pa pb, pc, pd		$c{:}a \cap b \wedge pa{:}\mathcal{P}(A) \wedge pb{:}\mathcal{P}(b) \wedge pc{:}\mathcal{P}(c) \wedge pd{:}pa \cap pb \Rightarrow pc \subset pd$
	$c{:}a \cap b$, $pc{:}\mathcal{P}(c)$ $pa{:}\mathcal{P}(a)$, $pb{:}\mathcal{P}(b)$ $pd{:}pa \cap pb$	$pc \subset pd$
		. $\forall X\ (X \in pc \Rightarrow X \in pd)$
x		$x \in pc \Rightarrow x \in pd$
	$x \in pc$	$x \in pd$
		$x \subset c$
		$x \in pa \wedge x \in pb$

découpé en Théorème 1 et Théorème 2

Démonstration par DATTE et Muscadet
du théorème $\forall A \forall B(\mathcal{P}(A \cap B) \subset \mathcal{P}(A) \cap \mathcal{P}(B))$

objets	hypothèses	conclusion
		$\forall A \forall B(\mathcal{P}(A \cap B) \subset \mathcal{P}(A) \cap \mathcal{P}(B))$
		$\forall A \forall B(!C{:}A \cap B, !PA{:}\mathcal{P}(A), !PB{:}\mathcal{P}(B), !PC{:}\mathcal{P}(C), !PD{:}PA \cap PB, PC \subset PD$
a, b, c, pa pb, pc, pd		$c{:}a \cap b \wedge pa{:}\mathcal{P}(A) \wedge pb{:}\mathcal{P}(b) \wedge pc{:}\mathcal{P}(c) \wedge pd{:}pa \cap pb \Rightarrow pc \subset pd$
	$c{:}a \cap b$, $pc{:}\mathcal{P}(c)$	
	$pa{:}\mathcal{P}(a)$, $pb{:}\mathcal{P}(b)$	
	$pd{:}pa \cap pb$	$pc \subset pd$
		. $\forall X\ (X \in pc \Rightarrow X \in pd)$
x		$x \in pc \Rightarrow x \in pd$
	$x \in pc$	$x \in pd$
		$x \subset c$
		$x \in pa \wedge x \in pb$
découpé en Théorème 1 et Théorème 2		


objets
hypothèses
conclusion
Théorème 1
...
x∈ pa
x ⊂ a
∀X (X∈ x ⇒ X∈ a)
t
t∈ x
t∈ a
t∈ c
t∈ a
Théorème 1 démontré
Théorème 2
...
x∈ pb
...
Théorème 2 démontré
Théorème_ démontré

Connaissances déclaratives

Propriétés des connaissances déclaratives

- séparées de leur mode d'emploi
- données en vrac
- indépendantes kles unes des autres
- faciles à manipuler, modifier, ajouter, enlever
- générales, utilisables dans de nombreuses applications
- lisibles
- parfois efficaces (ex ALICE, car l'ordre n'est pas figé)

- pour les utiliser une forme procédurale est en général préférable ou même nécessaire (pour l'ordinateur !), mais il est souhaitable que le système traduise lui-même le déclaratif en procédures

règles, métarègles et super-actions utilisées dans les exemples

règles données

regle => : si concl (A => B) alors ajhyp A, nouvconcl B

regle ∀ : si concl ∀X P alors créer X1 nouvconcl $P_{X\backslash X1}$

regle concl_et : si concl A B alors dem A dem B

regle def_concl : si concl C, definition C<=>D alors nouvconcl D

règles construites

regle ⊂ : si hyp A ⊂ B, hyp X ∈ A, alors ajhyp X ∈ B

regle ∩1 : si hyp C : A ∩ B, hyp X ∈ C alors ajhyp X ∈ A

regle ∩3 : si hyp C : A ∩ B, hyp X ∈ A, hyp X ∈ B alors ajhyp X ∈ C

regle parties1 : si hyp B:parties(A), hyp X ∈ B alors ajhyp X ⊂ A

regle parties2 : si hyp B:parties(A), hyp X ⊂ A alors ajhyp X ∈ B

super-actions définies par des paquets de règles

pour ajhyp H si H = A ∧ B alors ajhyp A ajhyp B
si hyp H ne rien faire
si H = ∀x P(x) alors consreg (H, ...)
…
sinon ajouter H

pour nouvconcl affecter concl C

pour consreg si def (A<=>B) … alors … consreg(…)
…

pour consreg(E, CC, AA, …)
si E=… alors … consreg(…)
... ajoucond(…)
ajoureg(...)

Evolution de l'expression de la règle =>

regle => si concl A => B alors ajhyp A nouvconcl B
REC IMP X IMP Y Z IMP ET X Y Z

regle(N, =>) :- concl(N, A => B), ajhyp(N, A), nouvconcl(N,B)

regle(N, =>) :- concl(N, A => B, Etape),
ajhyp(N, A, Etape1), nouvconcl(N, B, Etape1)

regle(N, =>) :- concl(N, A => B, Etape),
ajhyp(N, A, Etape1), nouvconcl(N,B,Etape1),
traces(N,Etape1, regle(=>), <les conditions>, <les actions>,
<les étapes des antécédents>, <message>)

traces (1) écrit des messages à chaque étape (mot à mot, français, ou rien)
(2) enregistre les paramètres dans des faits tracem(…) qui seront i
examinés à la fin pour extraire la trace utile

exemple de message :
pour démontrer H=>C, on suppose H et on doit démontrer C

Déclaratif ?

- les règles (sens de JP dans M) sont des **actions conditionnelles**, cad des règles opérationnelles (productions), on ajoute des faits, on crée des objets : **nécessaire** car on construit les étapes d'un démonstration
- **en vrac** ?
  oui : on a des types de règles gérés par des métarègles (règles générales et pas dangereuses, règles faisant les découpages (pas trop tôt), règles qui modifient la conclusion (par exemple en remplaçant un prédicat ou un symbole fonctionnel par sa définition, ou par tout autre chose, irréversible), règles gérant les hypothèses existentielles et disjonctives (expansivité possible), règles à appliquer en premier ou en dernier
  l'utilisateur peut en rajouter autant qu'il veut, c'est parfaitement déclaratif)
  non : ordre à l'intérieur des règles de même type
  - n'importe comment mais on peut tricher
  - règles exclusives, ok
  - règles si … si … alors sinon … plus commode (et préféré par les utilisateurs)

- **facile à manipuler, modifier, ajouter, enleve**r ? oui, sauf si elles sont déjà très compliquées
- **efficace** ? : parfois moins efficace que des représentations particulières de certaines propriétés
  - graphe des relations binaires de DATTE a été utilisé puis abandonné dans Muscadet
  - graphe de Mérialdo a été simulé mais l'efficacité de la représentation a disparu (sauf en cas de nécessité de donner un lemme)
  - des manipulations de formules (par exemple transformer en forme normale conjonctive) d'abord gérées par des règles ont été remplacées par un simple prédicat prolog récursif
- **traduire du déclaratif en procédures** ? non, mais traduire du déclaratif (définitions) en règles opérationnelles oui, par métarègles

Création de nouveaux objets

- élimination des symboles fonctionnels (prioritaire et plutôt procédural)
  - règle et super-action elifon (récursive et compliquée)
    P(f(x)) remplacée par !(y:f(x)) P(y)
  - règle :
    si concl (! y : f(x)) P(y) hyp(y1:f(x) alors nouvconcl P(y1)
    si concl (! y : f(x)) P(y) alors créer y1 ajhyp (y1:f(x)) nouvconcl P(y1)
  - superaction :

  pour ajhyp H
    si H= !(y:f(x))P(y), hyp (z:f(x)) alors ajhyp P(z)
    si H= !(y:f(x))P(y) alors créer z, ajhyp (z : f(x)), ajhyp P(z)ajhyp P(z)
- à partir d'une hypothèse ∃x P(x) (non prioritaire)
  - pour ajhyp H si H = ∃x P(x), hyp(P(x1) alors ne rien faire
    si H = ∃x P(x), alors créer x1, ajhyp P(x1)
  - les objets seront créés un par un et le moment de cette création est donné par des métarègles

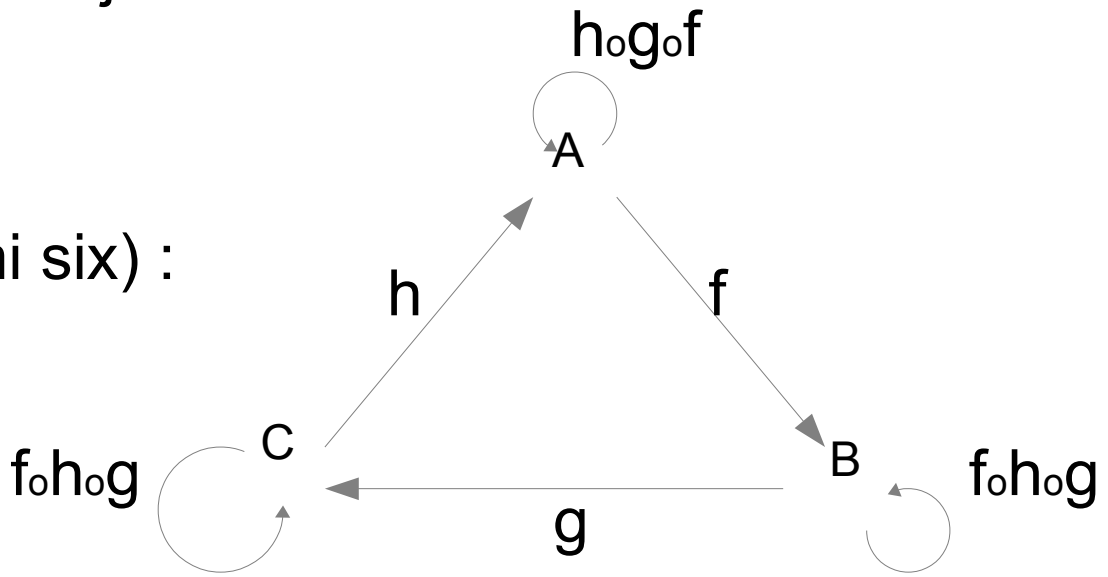
Exemple en théorie des ensembles (applications)

Théorème : Soient trois applications f, g, h de A dans B, B dans C, C dans A. Si parmi les trois applications $h \circ g \circ f$, $g \circ f \circ h$, $f \circ h \circ g$, deux sont injectives (resp. surjectives) et la troisième est surjective (resp. injective), alors f, g et h sont bijectives.

Par exemple (un cas parmi six) :

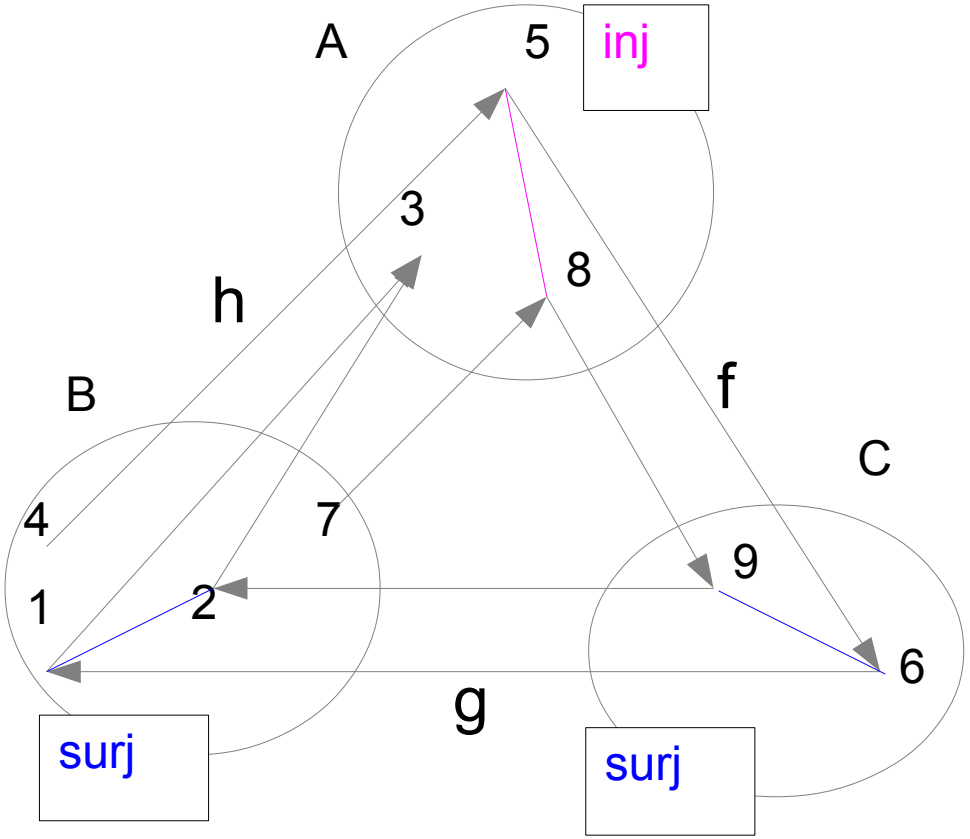
$h \circ g \circ f$ injective

$g \circ f \circ h$ et $f \circ h \circ g$ surjective

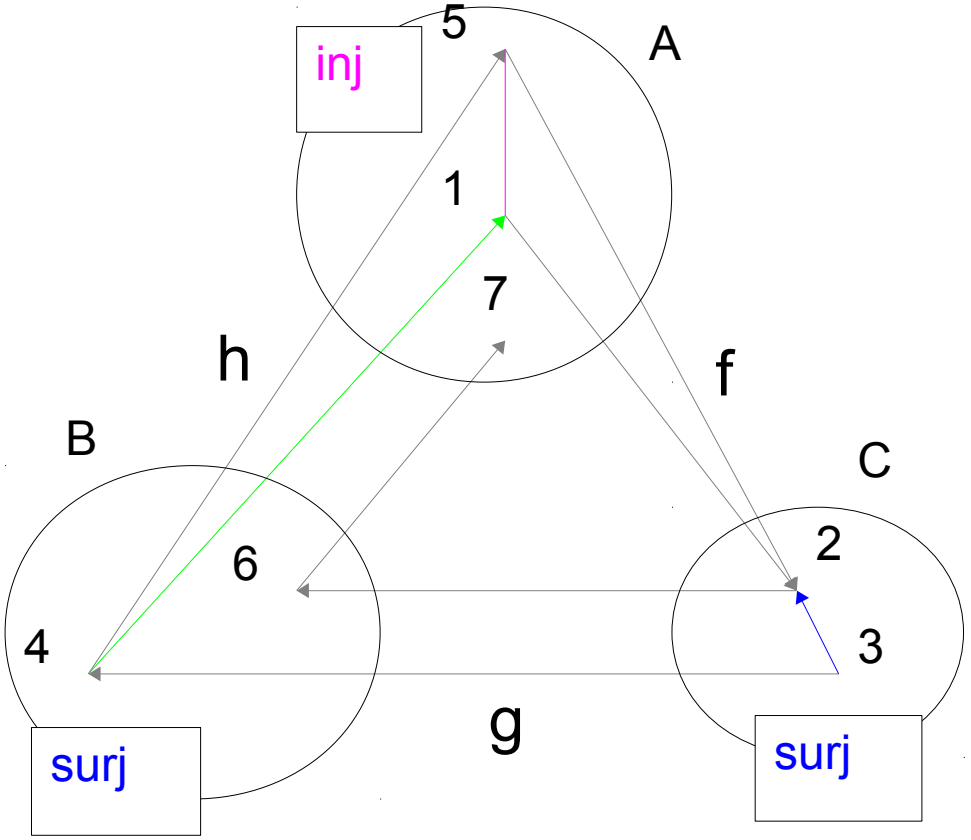


Cas $h \circ g \circ f$ injective, $g \circ f \circ h$ et $f \circ h \circ g$ surjective (un cas parmi six)

montrer que h est injective :
si 1 et 2 ont la même image 3, alors ils sont égaux



montrer que h est surjective :
4 est un antécédent de 1 car 1 est égal à son image 5



Objectifs

- à long terme
- à court terme

Et maintenant ?