# Une histoire de la logique mathématique au XXe siècle:
## de la philosophie à l'informatique

Christian Retoré
10 mars 2019 cours de logique (fin)
Professeur, Université de Montpellier & LIRMM-CNRS

## La logique est elle sulfureuse?

- **"Forse tu non pensavi ch'io loico fossi. [Lucifero]"** Dante Alighieri (1265-1321) Comedia, Inferno XXVII
  Une traduction pourrait être: "*sans doute ne savais tu pas que j'étais aussi bon logicien"*.
- *En tout cas, en France la logique est peu enseignée:*
  - *Philosophie?*
  - *Mathématiques?*
  - *Informatique?*
  - *Linguistique (sémantique et philosophie du langage)?*



## Avant la logique moderne

- Aristote
- Antiquité
- Scolastique
- XIXe anglo-américain



## La logique

- Art de raisonner correctement
- Avec la rigueur des raisonnements mathématiques
- Dériver correctement des énoncés
  …mais à partir de quels axiomes?
- Etude de la vérité dans une situation particulière, mais cela est plus récent.

## AUPARAVANT: ARISTOTE (III AV JC) L'ANTIQUITÉ & LA SCOLASTIQUE (MOYEN ÂGE)

- Certains types d'énoncés:
  - A Tout A est B
  - E Certains A sont B
  - I Aucun A est B
  - O Tous les A ne sont pas B.
    (ou Certains A ne sont pas B,
    mais le thème est différent, on parle des exceptions,
    pas de l'ensemble des A)
- Les fameux syllogismes (règles de déduction)
  - Barbara : *tout M est P, or tout S est M, donc tout S est P;*
  - Baroco : *tout P est M, or quelque S n'est pas M, donc quelque S n'est pas P*



## PLACE DE LA LOGIQUE EN PHILOSOPHIE

- A étudier en premier pour raisonner correctement (Organon Catégories,..)
- *« Celui qui souhaite atteindre la perfection humaine doit d'abord étudier la logique, puis les diverses branches des mathématiques dans l'ordre qui convient, puis la physique et enfin la métaphysique. »* (Maimonides, XIIe)



## PRINCIPES ARISTOTE & LES STOÏQUES

- Aristotle
  - Identité: Tout A est A
  - Non contradiction NON (A et NON A)
  - *"Tout personne niant le principe de non contradiction devrait être battue et brulée jusqu'à ce qu'elle admette qu'être battu n'est pas la même chose que ne pas être battu, et qu'être brulé n'est pas la même chose que ne pas être brûlé"* Avicenne (980-1037)
  - Tiers exclus: A ou NON A **(tertium non datur)**
- Stoïciens (calcul propositionnel)
  - Modus ponens *Si A alors B Or A donc B.*
  - Modus tollens *Si A alors B. Or NON B. Donc A.*
  - **Ex falso quodlibet sequitur**



## XIXE: LOGIQUE ALGÉBRIQUE ANGLO-AMERICAINE BOOLE, DE MORGAN, PIERCE

- Lois et calculs
- Calcul propositionnel: tables de vérités
  - X -> VRAI: VRAI
  - FAUX -> X: VRAI
  - VRAI->FAUX: FAUX
    (si Paris est en France alors Rome est en Chine)
- Pour les prédicats des règles parfois fausses
  $\forall x[I(x)->(F(x) v M(x))]$
  $\Leftrightarrow$
  $\forall x(I(x)->F(x))$ ou $\forall x(I(x)->M(x))$
  pensez à F= femme M=homme …

# XXe Siècle

La crise des fondements des mathématiques
La logique du permier ordre



## Gottlob Frege (1948-1925)

- Calcul des prédicats (formules quantifiées)
  - Tout x est P: (x) P(x) ou $\forall x\ P(x)$
  - Certains x sont P: Ex P(x) ou $\exists x\ P(x)$
- Tout A est B:
  Pour tout X, SI X est A ALORS X est B
  $\forall X\ (A(X) \to B(X))$
- Certains A sont B:
  Il existe X, tel que X est A ET ce même X est B.
  $\exists\ X\ (A(X)$ ET $B(X))$
- Système déductif avec des règles comme:
  - SI on a établit P(x) ou x est une variable dont on ne suppose aucune propriété ALORS on a $\forall x\ P(x)$



## Fondements des mathématiques Georg Cantor (1845-1918)

- Infinis, plusieurs sortes de nombres:
  - Ordinaux
  - Cardinaux
- Il ya plusieurs sortes d'infinis
- Il y a plus de nombres réels que d'entiers: Si nous avions une liste des nombres réels entre 0 et 1 (développements décimaux infinis, sans 999999…)
  1. 0,5786086346780965431346789098764666…
  2. 0,8861453781936528901766189918714428…
  3. 0,8623503894729474548494646849947452…
  4. 0,5618903412252820282534176444510186…
- On fabrique un nombre qui est différent du premier sur la première décimale, du second sur la deuxième, du troisième sur la troisième:
  - 0,6939….

1. Il n'est pas dans la liste!!!



## Giuseppe Peano (1848-1932) Axiomatisation de l'arithmétique

- Esperanto mathématique …
- Symboles: 0 (zéro) +, x …
  fonction S: successeur (l'entier suivant)
- $\forall n$ $S(n) \neq 0$
- $\forall n$ si $n \neq 0$ alors il existe p tel que $S(p)=n$
- $\forall n\ \forall p$ si $S(n)=S(p)$ alors $n=p$
- $\forall n$ $n + 0 = n$
- $\forall n\ \forall p$ $n+S(p) = S(n+p)$
- $\forall n$ $n \times 0 = 0$
- $\forall n\ \forall p$ $n \times S(p) = (n \times p) + n$
- Récurrence: pour établir qu'une propriété P(…) est vraie de tout entier il suffit de:
  1. Montrer qu'elle est vraie de 0
  2. Montrer qu'elle passe au successeur:
     si P(n) alors P(S(n))
  3. Un axiome **pour chaque formule F à une variable libre**:
     $(F(0)\ \&\ \forall p(F[p] \Rightarrow F[p+1])) \Rightarrow \forall n\ F[n]$

## David Hilbert (1862-1943) et le programme éponyme

- En 1900 à Paris 23 problèmes majeurs
- Programme de fondements des mathématiques
- Méthode axiomatique
- Montrer en raisonnant sur les preuves
  (de manière finitaire)
  que les axiomes ne conduisent jamais
  à une proposition fausse (par ex. 0=1)
  par les règles de déduction formelle



## Bertrand Russell (1872-1970)

- Paradoxe de Russell:
  - Schéma de compréhension sans restriction:
    $\{X / P(X)\}$ est un ensemble
  - $U = \{X / X \notin X\}$
  - $U \in U$ ?
  - $U \notin U$ ?
  - Schéma de compréhension restreint:
    $\{X \in V / P(X)\}$ est un ensemble
- Principia mathematica (1910-1913)
  formalisation/ axiomatisation purement logique
  des mathématiques



## Théorie axiomatique des ensembles Zermelo (1871-1953) puis Fraenkel (1891-1965)

- 1 seul symbole: $X \in Y$ (X appartient à Y) (et =)
- Deux ensembles sont égaux s'ils ont les mêmes éléments
- Il existe un ensemble sans élément (le vide)
  Formellement: $\exists y \forall x\ x \notin y$
- Paire, union, ensemble des parties
- Schéma de compréhension restreint:
  - $\{X \in Y \mid P(X)\}$ est un ensemble (exit le paradoxe de Russell)
- Axiome de l'infini: il existe N qui contient $\varnothing$ et
  si $x \in N$ alors $(x \cup \{x\}) \in N$
  au minimum N contient:
  $\varnothing$, $\{\varnothing\}$, $\{\varnothing,\{\varnothing\}\}$ $\{\varnothing,\{\varnothing\},\{\varnothing,\{\varnothing\}\}\}$,
  0, 1, 2, 3
- On peut faire TOUTES les mathématiques dans ZF



## Modèle, vérité: calcul propositionnel

- Une interprétation:
  - On fixe la valeur, vrai ou faux de chaque proposition élémentaire
  - On en déduit la valeur dans cette interprétation des propositions complexes par les tables de vérités
- Validité:
  - Une proposition dérivable (par exemple p->p) vaut vrai dans toute interprétation
- Complétude:
  - Si une proposition vaut vrai dans toute interprétation, alors elle est dérivable (Bernays, 1926)

## Modèle, vérité: calcul des prédicats ½

- La même chose, en plus compliqué:
  - Ensemble (domaine) par exemple les gens
  - Interprétation des constantes, des relations, …
    - Dort: ensemble de personnes
    - Connaît: ensemble de couples
  - On peut vérifier dans un modèle donné que, par exemple:
    - Pour tout x il existe y, x connaît y et y dort;
- Il y a des formules vraies dans TOUT modèle:
  - SI il existe X tel que pour tout Y
    X soit en relation R avec Y
    ALORS pour tout Y il existe un X
    tel que X soit dans la relation R avec Y
  - C'est-à-dire $\exists x\ \forall y\ P(x,y) \Rightarrow \forall y\ \exists x\ P(x,y)$
-



## Modèle, vérité: calcul des prédicats ½

- Validité: toute formule dérivable est vraie dans tout modèle,..
- **Complétude (Gödel, 1929) : Toute formule vraie dans tout modèle est dérivable**



## Luitzen Egbertus Jan Brouwer (1881-1966) et l'intuitionnisme

- Rejet du tiers exclus:
  - A ou non A
  - Différent de non (A et non A)
  - « ou » intuitionniste: plus exigeant
- Même chose pour le « il existe »
  - Existe x P(x) = P(0) ou P(1) ou P(2) …
  - Ne se déduit pas de NON pour tout X non P(x)
- Raisonnement qui construit la solution
- Ex. théorème des valeurs intermédiaires
- Paradoxe du buveur:
  - Il existe X tel que si X boit alors tout le monde boit.



## Kurt Gödel (1906-1978)

- Complétude de la logique du premier ordre
- Compatibilité de l'axiome du choix avec les axiomes de la théorie des ensembles
- Compatibilité de l'hypothèse du continu avec les axiomes de la théorie des ensembles
- Incomplétude de l'arithmétique
- Indécidabilité

- + *Une solution originale aux équation de la relativité générale dite « les univers tournants »*

## KURT GÖDEL (1906-1978)
## INCOMPLÉTUDE DE L'ARITHMÉTIQUE

- A contre courant (chute du programme de Hilbert)
  fascinant / mal compris
- Codage:
  - Formule -> entier
  - Preuve -> entier
  - Preuve(d,f) :
    - d est le code d'une démonstration D,
    - f est le code d'une formule F
    - et D est une preuve de F



## KURT GÖDEL (1906-1978)
## INCOMPLÉTUDE DE L'ARITHMÉTIQUE

- 1er théorème d'incomplétude
  Certains énoncés de l'arithmétique de sont ni démontrés ni réfutés par l'arithmétique.
  - Exemple concret:
    convergence des suites de Goodstein ( avec $\varepsilon_0$)
  - Théorème de Fermat ?
- 2e théorème d'incomplétude
  La cohérence l'arithmétique PA|-/- 0=1 fait partie des énoncés indémontrables.
- Gentzen 1934: cohérence de l'arithmétique avec induction sur $\varepsilon_0$



## ALSONSO CHURCH 1936
## INDÉCIDABILITÉ DU CALCUL DES PRÉDICATS

- Découle de l'incomplétude:
  - Q arithmétique sans le schéma de récurrence
    7 axiomes.
  - Toute extension de Q est indécidable
    (en utilisant l'incomplétude)
  - Q n'a qu'un nombre fini d'axiomes A1… A7
  - Donc il n'y a pas de procédure pour décider de A1,…,A7 => B qui est une formule du calcul des prédicats.
  - Le calcul des prédicats est indécidable.

  - Fragments décidables utiles:
    par ex. logique de description
    utilisées en IA et en représentation des connaissances.



## EN RÉSUMÉ: FONDAMENTAUX

- Preuves
- Modèles
- Théorie des ensembles, ordinaux, cardinaux
- *Lowenheim Skolem:*
  *une théorie qui admet un modèle infini*
  *en admet de toute cardinalité infinie*
- Complétude de la logique du premier ordre
- *Compacité:*
  *étant donné une famille de formules dont chaque partie finie admet un modèle, toute la famille admet un modèle*
- Incomplétude de toute théorie contenant l'arithmétique de Peano.
- Indécidabilité du calcul des prédicats

# INFORMATIQUE

Et calculabilité





## Pourquoi la Science informatique est elle aussi liée à la logique

- Informatique? Science? technologie ?
  - Données (informatique)
  - Calcul (computer science)
- Mécanisation du raisonnement (et du calcul) dans le programme de Hilbert
- Formalisation, codage,…(cf. Gödel)

- Dès les débuts analyse du langage naturel, test de Turing pour l'intelligence artificielle: berner 30% des humains dans un dialogue.





## Machine abstraite 1936 Alan Turing (1912-1954)

- Ruban avec des 0 et des 1 (qui peuvent « tout » représenter via divers codages)
- Tête qui parcours le ruban
- Etats: 1 initial plusieurs finaux
- Actions
  - lire, écrire (0, 1, rien),
  - déplacer (gauche, droite, rien),
  - Changement d'état
- Instructions:
  si dans tel état si on lit tel caractère
  alors on fait telle écriture,
  on déplace la tête et on va dans tel état.





## Machines de Turing: termine? Ou pas?

- La machine s'arrête quand on atteint un des états finaux
- Le problème de l'arrêt des machines de Turing est indécidable. Il n'y a pas de machine de Turing qui étant donnés (sur un ruban):
  - Un entier m qui code une machine de Turing M
    (un texte finit décrit la machine,
    et il se représente par un nombre entier)
  - Et un entier n

  Répond 1 si la machine M s'arrête sur l'entrée n
  et 0 sinon.

## Machine de John von Neuman (1903-1957)

- 1 processeur
- 1 mémoire : données + programme
- Contrôleur qui dicte la séquence des opérations

- Nos ordinateurs fonctionnent toujours ainsi, mais ils communiquent entre eux.



## Calculabilité à la Church (1903-1995) lambda calcul

- Du point de vue des ensembles, un nombre est la classe des ensemble ayant le même nombre d'éléments.
- On définit le nombre n comme:
  appliquer n fois une fonction F a quelque chose:
  S(S(S(0))) : 3
  mère(mère(mère(Jean))) : 3
  l'écriture de faire 3 fois une action sur un objet



## Calculabilité à la Church (1903-1995) lambda calcul

- En lambda calcul est fonction/action
- 3: (λfλx (f(f(f(x))))
- S: λnλfλx f(n f x)
- +: λn λp n f (p f x)
- Calculer c'est remplacer la variable par sa valeur… (λx. (x+3)) 5 -> 5+3 [substitution]
- Quelles fonctions sur les entiers peut on définir?
- Les mêmes que celle de Turing!
- Lambda calcul inventé pour formalisé la logique, pour substituer des formules dans d'autres: axiome appliqué à instance particulière :
  (λp p =>p)(A&B) donner (A&B)=>(A&B)



## Thèse de Church (1903-1995) Turing (1912-1954)

- La machine de Turing ou le lambda calcul de Church (équivalents) rendent exactement compte de tout ce qui se calcule automatiquement.
- Le lien avec la logique est assez clair: comme on peut coder par des entiers:
  - Les formules
  - Les preuves
  - On peut concevoir une machine qui vérifie par exemple si une formule est prouvable.

# PREUVES, 1ÈRE APPLICATION LOGIQUE INTUITIONNISTE ET PROGRAMMES CERTIFIÉS

Preuves, types et programmes





## 23 SIÈCLES APRÈS ARISTOTE — ENFIN! — DES RÈGLES DE DÉDUCTION SIMPLES

- Axiome A |– A
- Modus ponens:
  Si Γ|— A et Δ |— A→B
  alors Γ, Δ |— B
- Abstraction / généralisation :
  Si Δ, A |— B
  Alors Δ |— A → B
- Système suffisant pour dériver les propositions intuitionnistes valide (écrites avec →)
- Suffisant pour représenter les entiers!





## LA DÉDUCTION NATURELLE GERHARD GENTZEN (1909-1945), DAG PRAWITZ (1936-…)

- Déductions sous hypothèses
- Des règles simples:
- Bizarrement, la logique sous jacente est intuitonniste et non classique:
- ((a→b) →a) → a n'est pas démontrable mais elle est démontrable classiquement:
  - Si a vrai OK
  - Si a faux (a →b) vrai donc ((a→ b) →a) est faux OK





## LA CORRESPONDANCE DE CURRY-HOWARD (1900-1982 & 1926-…)

- Types de données = formule logique
  ex entier: (A→ (A→A) →A)
- Donnée de type A = preuve de A
- Type A → B
  fonction de A dans B c'est-à-dire
  fonction des preuves de A dans les preuves de B :
  évidemment: le modus ponens fait le boulot!
  preuve de A, preuve de A→B
  donc preuve de A→B

## L'entier 3 vu sous un angle différent

- Une preuve de : (A → (A → A) →A)?
- Il établir A sous les hypothèses (A-->A) et A
  on part de A|— A (0)
  et de A->A|— A → A (S) (axiomes)
  par modus ponens on obtient A, A→A|—A (1)
  par (1) en utilisant de nouveau (S), on obtient par modus ponens A, A→A, A→A |— A (2)
  par (2) en utilisant de nouveau (S), on obtient par modus ponens A, A→A, A→A, A→A |— A (3)
- Par abstraction: A |— (A→A) →A
- Par abstraction encore: |— A→ (A→A) →A





## L'entier 3 vu sous un angle différent

- Notre preuve de : (A→ (A→A) →A)
  consiste en 3 applications de A->A à A…
  c'est donc l'entier 3 du lambda calcul de Church
- On peut l'écrire (λx$^{A}$ λf$^{A->A}$ (f(f(f(x))))

10/04/2019 Ch. Retoré Cours Logique (fin) Logique au XXe: philo->info



## Correspondance logique / calcul

**Logique**

- Formule A
- Preuve de A
- Normalisation de la preuve (remplacement d'une hypothèse par une preuve)

**informatique**

- Type A
- Programme de type A
- Evaluation du programme (substitution d'une variable par sa valeur)

10/04/2019 Ch. Retoré Cours Logique (fin) Logique au XXe: philo->info



## Programme certifiés:
## les preuves vues comme des programmes

- Santé, aéronautique, finances, protocoles de connexion…
  **il faut des programme sûrs**
- **Deux méthodes (et demi):**
  1. Tests (pas vraiment sûrs, demie méthode)
  2. Vérification que le programme satisfait sa spécification
  3. **Preuves de la spécification → programmes**
     Alors il est sûr que le programme obtenu fait exactement ce qu'il est supposé faire.
     Programmes peu efficaces mais parfaitement sûrs.
- **Démonstration automatique (Coq)
  utile pour 1 et 2**

10/04/2019 Ch. Retoré Cours Logique (fin) Logique au XXe: philo->info

## PROGRAMMER = PROUVER (PROOFS AS PROGRAMS)

Prenons le cas d'une fonction f des entier dans les entiers.

p est un entier s'écrit logiquement ainsi

N(p): $\forall X \quad [X(0) \ \& \ (\forall y\ X(y) \rightarrow X(Sy))] \ \rightarrow X(n)$

« n satisfait toute propriété X qui est vraie de 0 et qui passe au successeur »

1. On écrit la spécification du programme: par exemple f(0)=1 f(Sx)=(S(S(S(x)))
2. On démontre formellement $\forall x\ N(x) \rightarrow N(f(x))$ en utilisant exclusivement la logique et la définition de f (Coq?)
3. On ne garde que la partie propositionnelle
4. On construit le lambda terme f
5. Lambda terme f appliqué à un entier p du lambda calcul, se réduit en l'entier 3p+1 lambda calcul



## PREUVES, 2E APPLICATION: CALCUL DU SENS D'UNE PHRASE

**Approche logique et compositionnelle de la grammaire et du sens**



## PREUVES

- Initialement: cohérence des mathématiques
- Contenu calculatoire (en logique intuitionniste)
- Programmes fonctionnels typés
  - Temps de calcul élevé
  - Mais totalement sûrs
- Vérification d'un protocole de communication, de programmes embarqués,
- Fondements des maths / informatique dès 1970
  - Jean-Yves Girard (1947-…) *mon directeur de thèse*
  - Per Martin-Löf (1942-…)
  - John Reynolds (1935-…)
  - Gérard Huet (1947-…)
  - Jean-Louis Krivine (1939-…)



## ANALYSE GRAMMATICALE: PREUVE JOACHIM LAMBEK (1922-2014)

- Logique sous structurelle
  - aux ressources (mots catégories grammaticales)
  - à l'ordre desdites ressources
- Analyse de S = preuve que S est une phrase
- Un lexique associe à chaque mot des formules (son comportement grammatical, ses manières de se combiner avec d'autres mots)

## Calcul de Lambek (1958)
## (logique linéaire non commutative)
## 4 règles universelles

$$\frac{A/B \quad B}{A}\,[/E] \qquad\qquad \begin{array}{c}\ldots [B]^n \\ \vdots \\ A \\ \hline A/B\end{array}\,[/I]^n$$

$$\frac{B \quad B\backslash A}{A}\,[\backslash E] \qquad\qquad \begin{array}{c}[B]^n \ldots \\ \vdots \\ A \\ \hline B\backslash A\end{array}\,[\backslash I]^n$$



## Exemple 1/5
## Lexique syntaxique

- Les enfants prendront une pizza.

- Enfant(s), pizza: n
- Prendront: (np\S)/np
- (tous) les, un : S/(np\S)/n ou ((S/np)\S)/n

- Parallèle (connu depuis l'ntiquité):
  - n:  nom commun: prédicat unaire
  - np\S Verbe intransitif : prédicat unaire
  - (np\S)/np Verbe transitif: prédicat binaire
  - np Groupe nominal: individu
  - np Nom propre: individu
  - n\n  Adjectif: prédicat ou prédicat de prédicat



## Exemple 2/5
## Lexique syntaxique et sémantique

| mot | ***catégorie syntaxique*** $u$<br>***type sémantique*** $u^*$<br>***sémantique :*** $\lambda$***-term of type*** $u^*$<br>$x^v$ ***signifie*** $x$ ***(variable, constante) de type*** $v$ |
|---|---|
| les | $(S/(np\backslash S))/n$ (subject)<br>$((S/np)\backslash S)/n$ (object)<br>$(e \to t) \to ((e \to t) \to t)$<br>$\lambda P^{e\to t}\,\lambda Q^{e\to t}\,(\forall^{(e\to t)\to t}\,(\lambda x^e(\Rightarrow^{t\to(t\to t)}\,(P\,x)(Q\,x))))$ |
| une | $((S/np)\backslash S)/n$ (object)<br>$(S/(np\backslash S))/n$ (subject)<br>$(e \to t) \to ((e \to t) \to t)$<br>$\lambda P^{e\to t}\,\lambda Q^{e\to t}\,(\exists^{(e\to t)\to t}\,(\lambda x^e(\wedge^{t\to(t\to t)}(P\,x)(Q\,x))))$ |
| enfant(s) | $n$<br>$e \to t$<br>$\lambda x^e(\mathtt{enfant}^{e\to t}\,x)$ |
| pizza | $n$<br>$e \to t$<br>$\lambda x^e(\mathtt{pizza}^{e\to t}\,x)$ |
| prendront | $(np\backslash S)/np$<br>$e \to (e \to t)$<br>$\lambda y^e\,\lambda x^e\,((\mathtt{prendront}^{e\to(e\to t)}\,x)y)$ |



## Exemple 3/5
## Analyse syntaxique
## = preuve du calcul de Lambek

$$\dfrac{\dfrac{\dfrac{\dfrac{(S/(np\backslash S))/n \quad n}{(S/(np\backslash S))}/_e \quad \dfrac{(np\backslash S)/np \quad [np]^1}{(np\backslash S)}/_e}{S}/_e}{S/np}/_i(1) \qquad \dfrac{((S/np)\backslash S)/n \quad n}{(S/np)\backslash S}\backslash_e}{S}\backslash_e$$

Le $\lambda$-terme correspondant est :

$\exists\forall = (une\ pizza)(\lambda o^e(les\ enfants)(prendront\ o))$

# Exemple 4/5
## Reduction → formule logique

$$
\begin{aligned}
&(\mathit{une\ pizza})(\lambda o\ (\mathit{les\ enfants})(\mathit{prendront}\ o))\\
&= (\lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\mathtt{pizza}^{e\to t}\ x)))(Q\ x))))\\
&\qquad (\lambda o \forall^{(e\to t)\to t}\ (\lambda w^{e}(\Rightarrow^{t\to(t\to t)}\ (\mathtt{enfant}^{e\to t}\ w)(((\mathtt{prendront}^{e\to(e\to t)}\ w)\ o\\
&= (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\mathtt{pizza}^{e\to t}\ x)))\\
&\qquad ((\lambda o \forall^{(e\to t)\to t}\ (\lambda w^{e}(\Rightarrow^{t\to(t\to t)}\ (\mathtt{enfant}^{e\to t}\ w)(((\mathtt{prendront}^{e\to(e\to t)}\ w)\ o\\
&= (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\mathtt{pizza}^{e\to t}\ x)))\\
&\qquad (\forall^{(e\to t)\to t}\ (\lambda w^{e}(\Rightarrow^{t\to(t\to t)}\ (\mathtt{enfant}^{e\to t}\ w)((\mathtt{prendront}^{e\to(e\to t)}\ w)\ x))))
\end{aligned}
$$

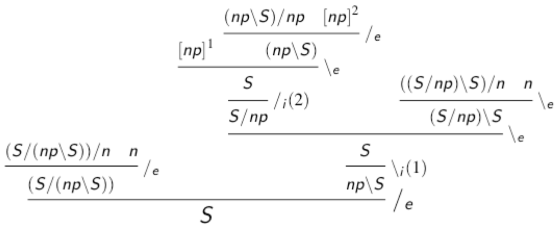
ce qui s'écrit communément :

$$\exists x.\ \mathit{pizza}(x) \wedge \forall w.\ (\mathit{enfant}(w) \Rightarrow \mathit{prendront}(w,x))$$



# Exemple 5/5
## Sans calcul, une 2e analyse

∀∃




On trouve l'autre interprétation :

$$\forall u.\ \mathit{enfants}(u) \Rightarrow \exists .x\ \mathit{pizza}(x) \wedge \mathit{prendront}(u,x)$$



# Commentaires

- Analyse syntaxique = **preuve**
- Lambda terme sémantique d'un mot / syntagme
  = **preuve** incomplète
- Lambda terme sémantique après réduction
  = formule logique
  = **preuve** de la correction de la formule logique
- Processus calculable (algorithme)
  correction assurée de la sémantique
  (si les informations lexicales soient correctes)
- Compositionnalité du sens:
  réduction de lambda termes typés = preuves



# Commentaires

- Ça marche: analyseur à large échelle du français
  GRAIL (Richard Moot CNRS LIRMM UM)
- Sémantique = formule logique ≠ connotations
- Intégration du sens lexical?
  **J'ai fini mon livre.**
  Evénement: lire, écrire, …?

## "Contrariwise, if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's logic." Lewis Carroll (1832-1898)



**Merci de votre attention.**
**Des questions?**



## Quelques références grand public

- **BD** LOGICOMIX de Apostolos Doxiadis, Christos H. Papadimitriou, Alecos Papadatos, and Annie di Donna Vuibert 2010 *Les principaux logiciens du début du XXe sont mis en scène. Pas technique, mais des annexes permettent d'aller plus loin.*
- **Roman** La déesse des petites victoires de Yannick Grannec Editions Anne Carrière. 2012 Prix des libraires 2013. *Gödel vu par sa femme. Très bonne reconstitution de la vie scientifique à Vienne puis à Princeton, avec des personnages comme Einstein (l'unique ami de Gödel), von Neuman, Morgenstern,…Très bon roman* per se.
- **BD** Les rêveurs lunaires, quatre génies qui ont changé l'histoire de Cédric Villani et Edmond Baudoin ( Gallimard/ Grasset) 2015 *Science et éthique pendant la 2$^{nde}$ guerre mondiale, avec Turing, les physiciens Heisenberg et Szilard, un général anglais, Hugh Dowding.*
- **Film** de Morten Tyldum avec Benedict Cumberbatch Imitation games 2014. *Film sur Turing, aux nombreux prix.*





## Deux ouvrages de vulgarisation

- **Essai** Les démonstrations et les algorithmes: introduction à la logique et à la calculabilité de Gilles Dowek 2010
  *Très bon ouvrage de vulgarisation.*
- **Essai** Pierre Cassou-Noguès Les démons de Gödel : Logique et folie Seuil 2015 *Sur Gödel comme le roman de Yannick Grannec, en plus précis sur les travaux de Gödel.*