Boolean Algebra

BA axiomatizes

1) logical operations $\wedge, \vee, \neg$ (Boole 1854)
2) set operations $\cap, \cup$, complement

- first instance of abstract algebra

Tautology Theorem

For every proper model $\mathcal{L}$ of BA and
for every B. equation e

$$BA \vdash e \overset{1}{\Leftrightarrow} BA \models e \overset{2}{\Leftrightarrow} \mathcal{L} \models e$$

- Equivalence 1 may be obtained from a completeness theorem for algebraic specifications (Birkhoff 1930)
- Equivalence 2 is a special property of BA
  Properties of logical operations $\hat{=}$ properties of set operations

Quiz

Find an equation e s.t.

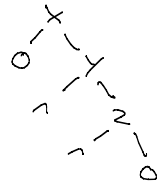
1) $\mathcal{T} \models e$
2) $BA \not\vdash e$

Prime Trees

$$\forall s \in BT \quad \exists^{\uparrow} t \in PT : \quad BA \vdash s = t$$

For all B. terms s, t
$BA \vdash s = t \Leftrightarrow \pi s = \pi t$

$PT \subset DT \subset BT$
ordered, reduced



Conditionals

$(s, t_0, t_1) \rightsquigarrow \bar{s} \cdot t_0 + s \cdot t_1$

if $s = 0$ then t_0 else t_1

Surprise: properties of 2-valued conditionals are decidable in BA

$(x, y, y) = y$

$(x, y, z) \circ u = (x, y \circ u, z \circ u)$

$(x, y, z) \circ (x', y', z') = (x, y \circ y', z \circ z')$

0-1 Theorem

$\Lambda \in BT$ closed $\Rightarrow$ $BA \vdash \Lambda = 0$ $\vee$ $BA \vdash \Lambda = 1$

Expansion Theorem

$s, x \in BT \Rightarrow \quad BA \vdash s = (x, s[x:=0], s[x:=\epsilon])$

Suffices to show

1) $BA \vdash x \cdot s = x \cdot (s[x:=1])$

2) $BA \vdash \bar{x} \cdot s = \bar{x} \cdot (s[x:=0])$

Proof by induction on s, using the following tautologies

$x \cdot x = x \cdot 1$ $\qquad$ $x \cdot (y \cdot z) = (x \cdot y) \cdot (x \cdot z)$

$x \cdot \bar{y} = x \cdot \overline{xy}$ $\qquad$ $x \cdot (y + z) = x \cdot y + x \cdot z$

Canonicity Lemma

$\mathcal{Y}$ proper model of BA

s, t different prime trees

$\Rightarrow \exists M : \mathcal{Y} \leq M \wedge \hat{M} s \neq \hat{M} t$

Proof is somewhat tricky

requires notion of

significant variables $SV_{\mathcal{Y}} s$

Significant Variables

$\mathcal{G}$ proper model of BA

$SV_{\mathcal{G}}\,\sigma := \{x \mid \tau x = \mathbb{B} \wedge \exists \mathcal{Y}: \mathcal{G} \subseteq \mathcal{Y} \wedge \exists v \in \mathcal{G}\mathbb{B}: \hat{\mathcal{Y}}\sigma \neq \hat{\mathcal{Y}}_{x,v}\sigma\}$

F0 $\boxed{BA \vdash \sigma = t \Rightarrow SV_{\mathcal{G}}\sigma = SV_{\mathcal{G}}t}$

F1 $\boxed{SV_{\mathcal{G}}\sigma \subseteq \mathcal{N}\sigma}$ Coincidence

More Facts

$\mathcal{G}$ proper model of BA, $\mathcal{G} \subseteq \mathcal{Y}$

F2 $\boxed{\begin{aligned} \mathcal{Y}x = \mathcal{G}o &\Rightarrow \mathcal{Y}(x,\sigma,t) = \mathcal{Y}\sigma \\ \mathcal{Y}x = \mathcal{G}\tau &\Rightarrow \mathcal{Y}(x,\sigma,t) = \mathcal{Y}t \end{aligned}}$

F3 $\boxed{\mathcal{Y}\sigma_0 \neq \mathcal{Y}\sigma_1 \wedge x \notin \mathcal{N}\sigma_0 \cup \mathcal{N}\sigma_1 \Rightarrow x \in SV_{\mathcal{G}}(x,\sigma_0,\sigma_1)}$ F2

F4 $\boxed{\begin{aligned} &\mathcal{Y}\sigma_0 \neq \mathcal{Y}t_0 \wedge x \notin \mathcal{N}\sigma_0 \cup \mathcal{N}t_0 \\ &\Rightarrow \exists \mathcal{Y}': \mathcal{G} \subseteq \mathcal{Y}' \wedge \hat{\mathcal{Y}}'(x,\sigma_0,\sigma_1) \neq \hat{\mathcal{Y}}'(x,t_0,t_1) \end{aligned}}$ $\mathcal{Y}' = \mathcal{Y}_{x,\mathcal{G}o}$ F2

$\mathcal{G}$ proper model of BA and $\sigma, t \in PT$. Then

1) $x \in \mathcal{N}\sigma \iff x \in SV_{\mathcal{G}}\sigma$

2) $\sigma \neq t \Rightarrow \exists \mathcal{Y}: \mathcal{G} \subseteq \mathcal{Y} \wedge \hat{\mathcal{Y}}\sigma \neq \hat{\mathcal{Y}}t$

Proof. By ind on $|\sigma| + |t|$, both claims together

1) "$\Leftarrow$" by Coincidence. "$\Rightarrow$" by case analysis.
   - Case $\sigma \in \{0,1\}$. ✓
   - Case $\sigma = (x,\sigma_0,\sigma_1)$. Follows with IH (2) and F3

2) Case $\sigma, t \in \{0,1\}$. ✓
   - Case "The root variable of σ doesn't occur in t or vice versa" Follows with (1) and F0
   - Case "σ and t have the same root variable" Follows with IH (2) and F4 □

Modelling with Boolean Equations

- Will consider an application
- Switch to ordinary math mode
- Boolean will mean 2-valued, i.e., $\mathbb{B} = \{0,1\}$

Diet Rules of an Old Gentleman

1) If you don't drink beer, always eat fish

2) If you have both beer and fish, don't eat ice cream

3) If you don't drink beer or eat ice cream, don't have fish

Model the Diet Rules

with 3 Boolean observables

B: meal includes beer

F: meal includes fish

I: meal includes ice cream

and one equation per rule

$$\overline{B} \to F = 1$$

$$B \wedge F \to \overline{I} = 1$$

$$\overline{B} \vee I \to \overline{F} = 1$$

1) If you don't drink beer, always eat fish

2) If you have both beer and fish, don't eat ice cream

3) If you don't drink beer or eat ice cream, don't have fish

What the theory pays off

What you have learned so far in this course doesn't help with the modelling but once you have the model it pays off:

1) Are there meals that satisfy the rules?
Is the equation system solvable?

2) Are there other (simpler) diet rules that license exactly the same meals?
Are there equivalent equation systems?

Definition of Solutions

$\mathfrak{A}$: structure

$\mathcal{U} \subseteq Var \cup Con$, $\mathcal{U} \cap \Sigma_{\mathfrak{A}} = \emptyset$ Unknowns

$$Sol_{\mathfrak{A},\mathcal{U}}\, E := \{\sigma \mid \exists \mathfrak{Y}: \mathfrak{A} \subseteq \mathfrak{Y} \wedge \mathfrak{Y} \models E \wedge \sigma \subseteq \mathfrak{Y} \wedge Dom\,\sigma = \mathcal{U}\}$$

Equation System → Boolean Term

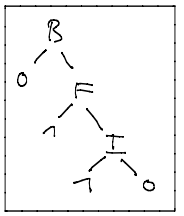
Formalize the equation system E as a B.term $\wedge$

$$(\overline{B} \to F)\ (B \wedge F \to \overline{I})\ (\overline{B} \vee I \to \overline{F})$$

- Solutions E = Solutions $\wedge = 1$
- E solvable $\iff$ $\forall \not\models \wedge = 0$
- $\forall \models \wedge = t \implies$ Solutions $\wedge = 1$ = Solutions $t = 1$
- $\pi \wedge$ can serve as solved form for E

Solving $E \doteq$ Simplifying $\wedge$

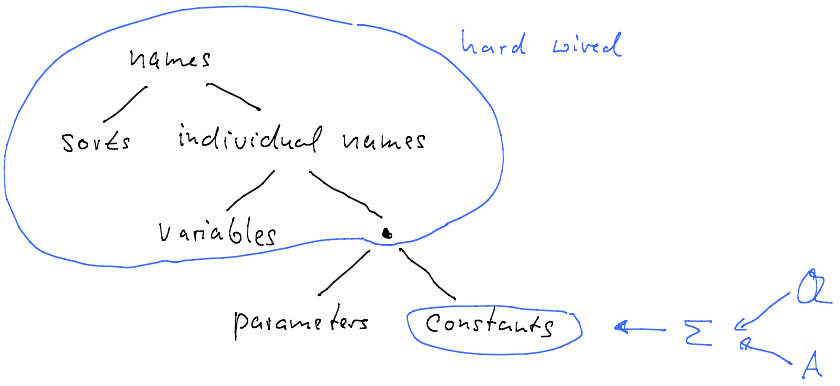
$\pi \wedge =$



$$\begin{aligned}
\wedge &= (\overline{B} \to F)\ (B \wedge F \to \overline{I})\ (\overline{B} \vee I \to \overline{F}) \\
&\equiv (\overline{\overline{B}} + F)\ (\overline{B \cdot F} + \overline{I})\ (\overline{\overline{B} + I} + \overline{F}) \\
&\equiv (B + F)\ (\overline{B} + \overline{F} + \overline{I})\ (B \cdot \overline{I} + \overline{F}) \\
&\equiv (B + F)\ (\overline{B} + \overline{F} + \overline{I})\ (B + \overline{F})\ (\overline{I} + \overline{F}) \\
&\equiv (\quad)(\quad)(\quad)(\quad)\ B \qquad \text{Resolution} \\
&\equiv (\overline{I} + \overline{F})\ B \qquad \text{Absorption}
\end{aligned}$$

Parameters

2006/6/12



Example

sorts	$\mathbb{B}$
constants	$0, 1 : \mathbb{B}$
	$- : \mathbb{B} \to \mathbb{B}$
	$+, \cdot : \mathbb{B} \to \mathbb{B} \to \mathbb{B}$
variables	$x, y, z : \mathbb{B}$
parameters	$a, b, c : \mathbb{B}$

signature

notational declaration

- parameters = individual names
  minus variables
  minus constants

- Variables and parameters act both as "unknowns"

- Variables are generative, parameters are not

$$x \cdot 1 = x \vdash a \cdot 1 = a$$

$$\vdash (x+a) \cdot 1 = x + a$$

Generativity

$$\text{ker}\ \theta \subseteq \text{Var} \Rightarrow \{\ell\} \vdash \varphi\theta\ell$$

The deduction rule behind generativity is $\exists$

Semantically, generativity amounts to universal quantification

$$\Omega \models \ell \iff \forall \mathcal{Y} : \Omega \subseteq \mathcal{Y} \Rightarrow \mathcal{Y} \models \ell$$

structures must not interpret variables

Do we need variables ??

Do specifications need variables ?

$\forall A\ \exists A' :\quad A' \text{ closed} \wedge A \dashv\vdash A'$?

$e \qquad e'$

$\{x \cdot 1 = x\} \dashv\vdash$

The unknowns of equation systems should be parameters

$\{a \cdot b = 0,\ a + b = 1\} \overset{BA}{\dashv\vdash} \{a = \bar{b}\}$

$\{x \cdot y = 0,\ x + y = 1\} \overset{BA}{\dashv\vdash} \{x = \bar{y}\}$

Deduction Rule ξ

$$\frac{r = t}{\lambda x.r = \lambda x.t}$$

- Sound if x var: $\forall \Omega:\ \Omega \vdash r = t \Rightarrow \Omega \vdash \lambda x.r = \lambda x.t$
- unsound otherwise:

$a \cdot u = a \rightsquigarrow \lambda u.\ a \cdot u = \lambda u.\ a$

$\rightsquigarrow (\lambda x.\ a \cdot x) = \lambda x.\ a$

$\top,\ u = 1$

$,\ a = 1$

Deduction without ξ seems incomplete

$\vdash \lambda x.\ \underbrace{(\lambda x.x)\,x} = \lambda x.\ x$?

Stability

Θ stable for A $\Rightarrow$

1) $A \vdash e \Rightarrow \gamma\Theta A \vdash \gamma\Theta e$

2) $A \models e \Rightarrow \gamma\Theta A \models \gamma\Theta e$

Counterexample

$x \cdot 1 = x$

$\Theta = \{1 := x\}$

$x \cdot x = x$

Θ stable for A :$\Leftrightarrow$

1) $\mathrm{Ker}\,\Theta \cap \mathrm{Var} = \emptyset$

2) $\forall e \in A \;\; \forall c \in \mathcal{N}e \;\; \forall x \in \mathcal{N}(\Theta c) : \; x \notin \mathcal{N}e$

↑ constant or parameter

Condition (2) not needed if there are no variables

(2) satisfied if A closed or Θc closed for $c \in \mathrm{Ker}\,\Theta$

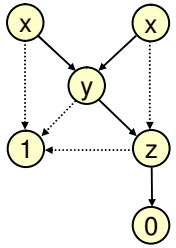
Efficient Implementation of Prime Tree Operations

[R. Bryant 1986]

- direct implementation of "and t_1 t_2" is exponential (exponential number of recursive calls)
- **Minimal graph representation** of prime trees yields constant equality test
- **Memoing*** of triples "and t_1 t_2 = t_3" yields $O(n^2)$ algorithm where n is the number of nodes reachable from t_1, t_2

* Dynamic Programming

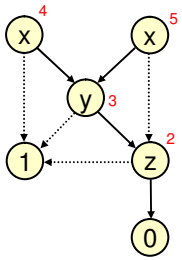
Minimal Graph Representation



- Every node describes a prime tree
- Graph describes a subtree-closed set of prime trees
- Graph minimal iff different nodes describe different trees

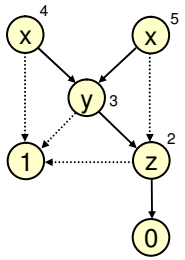
Binary decision diagrams (BDDs)

Graph → Table



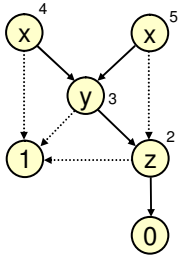
Number nodes of graph

Graph → Table



2	(z,1,0)
3	(y,1,2)
4	(x,1,3)
5	(x,2,3)

Graph → Table → Function



i	tab(i)
2	(z,1,0)
3	(y,1,2)
4	(x,1,3)
5	(x,2,3)

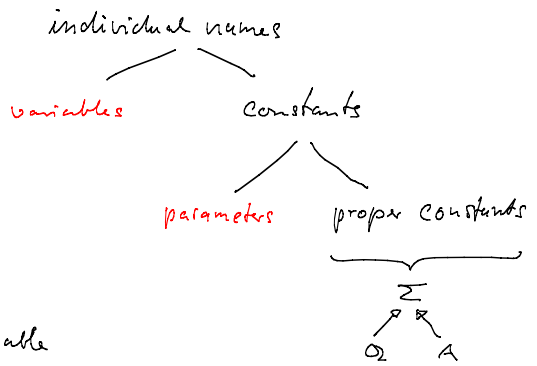
Graph minimal iff tab injective

Constant Time Realization of cond

cond(x,n,n') =
  if n=n' then n
  else if (x,n,n') ∈ Dom(tab^{-1})
    then tab^{-1} (x,n,n')
    else let n'' = least number not in Dom tab
      in tab := tab[n'':=(x,n,n')] ;
        n''

Implement tab^{-1} with hashing

2006-6-14



variable ≙ argument variable

free variable ≙ dangling argument pointer

parameter ≙ global variable

Examples

Specification: BA
Constants: $0, 1, -, +, \cdot$
Variables: x, y, z
parameters: a, b, c

$x + 0 = x$

$\lambda x.\ x+0 = \lambda x.\ x$ ← x occurs in the description but not in the term

BA, $a = b \vdash a = a \cdot b \cdot b \cdot a$

BA, $x = y \vdash 0 = 1$

BA, $a = b \nvdash 0 = 1$

Notational Conveniences

$e_1, e_2 \vdash e_3 \quad \rightsquigarrow \quad \{e_1, e_2\} \vdash \{e_3\}$

$\Theta A \quad \rightsquigarrow \quad \Psi \Theta A$

$\Theta e \quad \rightsquigarrow \quad \Psi \Theta e$

$\Theta 1 \quad \rightsquigarrow \quad \Psi \Theta 1$

$\Psi \vdash A \quad \rightsquigarrow \quad \forall e \in A:\ \Psi \vdash e$

Closed Specifications

A closed $:\Leftrightarrow \quad \mathcal{N}A \cap Var = \emptyset$

Extensionality $\quad s = t \vdash \lambda x.\ s = \lambda x.\ t$

$\Rightarrow$ variables ≙ dangling argument references

$\Rightarrow$ every specification is equivalent to a closed specification

$\Rightarrow$ open specifications are a notational convenience

Closed specs have nice properties

Let A be closed. Then:

1) $A \models e \iff \forall y : y \models A \Rightarrow y \models e$ — semantic entailment
2) $A \models e \Rightarrow \Theta A \models \Theta e$
3) $A \vdash e \Rightarrow \Theta A \vdash \Theta e$

} stability

$\Rightarrow$ Generativity is a consequence of stability

$\Rightarrow$ Can replace vars with params and vice versa in $A \models e$ and $A \vdash e$

$\Rightarrow$ Variables are not essential for $A \models e$, $A \vdash e$

$\Rightarrow$ Generativity and stability for open specs are consequences of stability for closed specs

Formal proofs

- Gödel-style proofs (derivations)
- conversion proofs

compile

Recall

$A \vdash e :\iff \exists$ Gödel-style proof of e from A

Soundness: $A \vdash e \Rightarrow A \models e$

Example of Gödel-style proof

$fx = a \vdash fy = a$ (f, a constants, x variable)

1	$fx = a$	
2	$\lambda x.\, fx = \lambda x.\, a$	ξ 1
3	$(\lambda x.\, fx)\, y = (\lambda x.\, a)\, y$	CR 2
4	$(\lambda x.\, fx)\, y = fy$	β
5	$(\lambda x.\, a)\, y = a$	β
6	$fy = (\lambda x.\, fx)\, y$	Sym 4
7	$fy = (\lambda x.\, a)\, y$	Trans 6, 3
8	$fy = a$	Trans 7, 5

Def Gödel-style proof

A Gödel-style proof of e from A is a tuple $(e_1, \dots, e_n)$ such that

1) $e_n = e$
2) $\forall i \in \{1, \dots, n\}$:
   $e_i \in A$ or
   exists instance (E, e_i) of a deduction rule such that $E \subseteq \{e_1, \dots, e_{i-1}\}$

This definition will work for every set of inference rules

Example of conversion proof

$BA \vdash x = x \cdot x$

	x			
$=$	$x \cdot 1$	Id	$x = x \cdot 1$	
$=$	$x(x + \bar{x})$	Compl	$x + \bar{x} = 1$	Sym, CR
$=$	$xx + x\bar{x}$	Dist	$x(y+z) = xy + xz$	Generativity
$=$	$xx + 0$	Compl	$x\bar{x} = 0$	CR
$=$	xx	Id	$x + 0 = x$	Generativity

Definition of conversion proof

A conversion proof of e from A is a tuple $(s_1, \ldots, s_n)$ such that

1) $e = (s_1, s_n)$
2) $\forall i \in \{1, \ldots, n-1\}$:

(s_i, s_{i+1}) is a λ-conversion step or an A-conversion step

Trans, Ref

Def of conversion steps

A λ-conversion step is an equation $s = t$ such that $s = t$ or $t = s$ is a λ-reduction step

An A-conversion step is an equation $s = t$ such that $s = t$ or $t = s$ is an A-reduction step

Sym

Definition of λ-reduction steps

- A β-reduction step is an extension of an equation $(\lambda x.s)t = s[x := t]$
- An η-reduction step is an extension of an equation $\lambda x.sx = s$ where $x \notin Vs$
- A λ-reduction step is a β- or an η-reduction step

β, η

Definition of A-reduction steps

An *A-reduction step* is
an extension of an instance of an equation in A

Extensions and instances

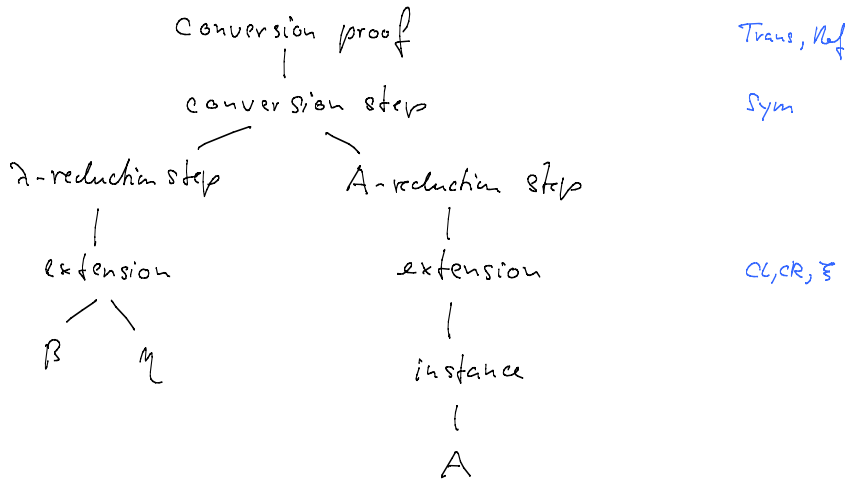
- An *extension of e* is an equation that can be derived from e with CL, CR, ξ

$x = a$
$fx = fa$
$\lambda x.fx = \lambda x.fa$
$(\lambda x.fx)a = (\lambda x.fa)a$

- An *instance of e* is an equation Θe where $Var\,\Theta \subseteq Var$

$fx = y$
$fa = y$
$(\lambda x.x)x = y$
$(\lambda x.x)a = fy$

$(\beta, \xi, CL, CR, Sym, Trans)$ Generativity



∃ Gödel-style proof of e from A
$\iff$ ∃ conversion proof of e from A

Proof

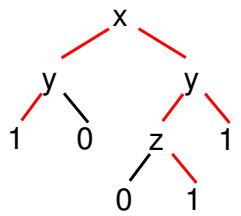
"$\Leftarrow$" easy

"$\Rightarrow$" Show for every instance (E, e') of a deduction rule:
$\forall e'' \in E$: (∃ conversion proof of e'' from A)
$\Rightarrow$ (∃ conversion proof of e' from A)

Clause Forms (DNF, CNF)

Willard V. Quine.
On Cores and Prime Implicants of Truth Functions.
American Mathematical Monthly, 1959.

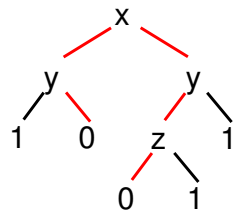
Decision Tree → DNF



$(\neg x \wedge \neg y) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y)$

yields 1 iff one of the clauses yields 1

Decision Tree → CNF



$(x \vee \neg y) \wedge (\neg x \vee y \vee z)$

yields 0 iff one of the clauses yields 0

Conjunctive Normal Forms

$CNF = 1 \mid DC_1 \wedge \cdots \wedge DC_n$ where $n \geq 1$

$DC = 0 \mid L_1 \vee \cdots \vee L_n$ disjunctive clause
where $n \geq 1$ and no variable occurs more than once

$L = x \mid \neg x$ literal

∀ B. term ∃ equiv. CNF

Disjunctive Normal Forms

$DNF = \top \mid CC_1 \vee \dots \vee CC_n$ where $n \geq 1$

$CC = 0 \mid L_1 \wedge \dots \wedge L_n$ **conjunctive clause**
where $n \geq 1$ and no variable occurs more than once

$L = x \mid \neg x$ **literal**

$\forall$ B. term $\exists$ equiv. DNF

$\wedge$ CNF equiv. to $t \iff \hat{\wedge}$ DNF equiv. to $\hat{t}$

Clause Sets

Good representations for CNFs and DNFs

Clause C: finite set of B. terms

Clause set S: finite set of clauses

Conjunctive interpretation $\bigwedge_{C \in S} \bigvee_{t \in C} t$

Disjunctive interpretation $\bigvee_{C \in S} \bigwedge_{t \in C} t$

$\bigwedge_{\in \emptyset} = \top$, $\bigvee_{\in \emptyset} = 0$

Literal Clause Sets

- clause is called **literal** if it contains only literals
- clause set is called **literal** if it contains only literal clauses

S, S' literal clause sets:
S, S' conj. equiv. $\iff$ S, S' disj. equiv

equivalent wrt conjunctive interpretation (i.e., S, S' describe same B. function)

Proof: Duality.

Normal Clause Sets

- C **trivial** if $\exists t \in C. \ \overline{t} \in C$
- C **normal** if C literal and not trivial
- S **normal** if all clauses of S are normal

Normal clause sets represent CNFs and DNFs

Explicitness

For every normal clause set S:
S conj. equiv. to $1 \iff S$ disj. equiv. to $0 \iff S = \emptyset$

CNFs and DNFs are not canonical

$$x = xy + x\bar{y}$$

$$x = (x+y)(x+\bar{y})$$

Will define conjunctive and disjunctive **prime forms** that are canonical (set representation required)

Redundant Clauses

C redundant for S if

either C trivial

or $\exists D \in S.\ D \subseteq C$

Deletion of redundant clauses is equivalence transformation wrt both interpretations

$$x + \bar{y}yz = x \qquad \text{(Compl, Id)}$$

$$x + y + yz = x + y \qquad \text{(Absorption)}$$

Resolvents

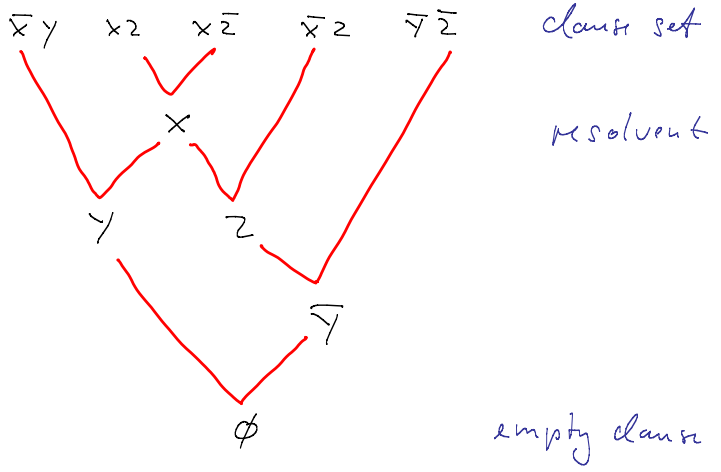
$(C - \{t\}) \cup (D - \{\bar{t}\})$ **resolvent for** S

if $C, D \in S$ and $t \in C$ and $\bar{t} \in D$

Addition of resolvents is equivalence transformation wrt both interpretations

$$xy + \bar{x}z = xy + \bar{x}z + yz \qquad \text{(Resolution)}$$

Example



1) Addition of resolvents and deletion of redundant clauses are equivalence transformations wrt both interpretations

2) Adding non-redundant resolvents and deleting redundant clauses terminates

Proof of Termination

Cla = set of all clauses containing only terms appearing in initial clause set — finite!

Red S = $\{C \in Cla \mid C \text{ redundant for } S\}$

- Idea: $Red\,S_1 \subseteq Red\,S_2 \subseteq Red\,S_3 \subseteq \dots \subseteq Cla$
- Deletion of a redundant clause doesn't change Red S
- Addition of a non-redundant resolvent makes Red S larger. □

Prime Forms

A clause set is called **prime form** if it
- is literal
- contains no redundant clauses
- has no non-redundant resolvent

Resolution Theorem

∀ prime form S ∀ literal clause C:

S equiv. to $S \cup \{C\}$ ⇔ C redundant for S

Proof in lecture notes 2004

Semantic redundancy = syntactic redundancy

Consequences of Resolution Theorem

Different prime forms denote different B. functions (wrt both interpretations) — **Canonicity**

For every prime form S:
S conj. equiv. to 0 ⇔ S disj. equiv. to 1 ⇔ $S = \{\emptyset\}$ — **Explicitness**

∀ prime form S ∀ literal clause set S':
S, S' equiv. ⇒ ∀ C ∈ S' C redundant for S

CPFs and DPFs

A prime form S is called a **CPF** [**DPF**] **for a term** t if S is equivalent to t and the conjunction [disjunction] interpretation.

- Every term has exactly one CPF and DPF
- For all B. terms s,t the following are equivalent
  (1) s and t are equivalent
  (2) s and t have the same CPF
  (3) s and t have the same DPF

Proof easy with the previous theorems

Example

$E \rightsquigarrow \{t = \top\}$ normalisation
$\rightsquigarrow$ CNF for t
$\rightsquigarrow$ CPF for t
$\rightsquigarrow$ solved form for E

$\bar{x} \to y = \top$
$xy \to \bar{z} = \top$
$(z + \bar{x}) \to \bar{y} = \top$

CNF

$\bar{x} \to y = x + y$
$xy \to \bar{z} = \overline{xy} + \bar{z} = \bar{x} + \bar{y} + \bar{z}$
$(z + \bar{x}) \to \bar{y} = \overline{z + \bar{x}} + \bar{y} = \bar{z}x + \bar{y}$
$= (\bar{z} + \bar{y})(x + \bar{y})$

CPF

$xy \quad \bar{x}\bar{y}\bar{z} \quad \bar{y}\bar{z} \quad x\bar{y}$

x

solved form

$x(\bar{y} + \bar{z}) = \top$

x : beer
y : fish
z : ice cream

Entailment Relations
Inference Systems
Confluent Relations

abstractions useful for logical systems

2006/6/19+21

Entailment Relations

$A \vdash x$

assumptions / axioms — consequence / conclusion

$A \subseteq \mathcal{X}$, $x \in \mathcal{X}$ statements

Definition

An entailment relation on a set $\mathcal{X}$ is a set $\vdash \subseteq \mathcal{P}\mathcal{X} \times \mathcal{X}$ such that for all $A, A', B \subseteq \mathcal{X}$:

1) $x \in A \Rightarrow A \vdash x$ Expansivity

2) $A \vdash x \wedge A \subseteq A' \Rightarrow A' \vdash x$ Monotonicity

3) $A \vdash B \wedge A \cup B \vdash x \Rightarrow A \vdash x$ Idempotence

where $A \vdash B :\Leftrightarrow \forall x \in B : A \vdash x$

Semantic entailment and deductive entailment are entailment relations on the set of all equations

Verify!

$A \vdash e$ defined by inference system $\Rightarrow$ $\vdash$ entailment relation

$A \models e :\Leftrightarrow \forall Q : Q \models A \Rightarrow Q \models e$

$Q \models A :\Leftrightarrow \forall e \in A : Q \models e$

$\Rightarrow$ $\models$ entailment relation

Notations

$x_1, \dots, x_n \vdash x \quad :\Leftrightarrow \quad \{x_1, \dots, x_n\} \vdash x$

$A, x \vdash x' \quad :\Leftrightarrow \quad A \cup \{x\} \vdash x'$

$A, B \vdash x \quad :\Leftrightarrow \quad A \cup B \vdash x$

$A \vdash B \quad :\Leftrightarrow \quad \forall x \in B : A \vdash x$

$B \overset{A}{\vdash} B' \quad :\Leftrightarrow \quad A, B \vdash B'$

} reflexive and transitive

$A \dashv\vdash A' \quad :\Leftrightarrow \quad A \vdash A' \wedge A' \vdash A$

$B \overset{A}{\dashv\vdash} B' \quad :\Leftrightarrow \quad A, B \dashv\vdash A, B'$

} equivalence relations

Entailment Equivalence

$$A \dashv\vdash A' \Leftrightarrow \forall x \in \mathcal{X} : A \vdash x \Leftrightarrow A' \vdash x$$

If $A \dashv\vdash A'$, then

1) $A \cup B \dashv\vdash A' \cup B$
2) $A \vdash x \Leftrightarrow A' \vdash x$
3) $A \vdash B \Leftrightarrow A' \vdash B$
4) $B \vdash A \Leftrightarrow B \vdash A'$
5) $B \overset{A}{\dashv\vdash} B' \Leftrightarrow B \overset{A'}{\dashv\vdash} B'$

An entailment relation $\vdash$ is

- compact if
  $A \vdash x \Rightarrow \exists \text{ finite } A' \subseteq A : A' \vdash x$
- effective if $\mathcal{X}$ decidable and
  A semi-decidable $\Rightarrow \{x \mid A \vdash x\}$ semi-decidable

Deductive entailment is compact and effective — proof easy

Semantic entailment is neither compact nor effective — Consider axiomatisation of $\mathbb{N}$

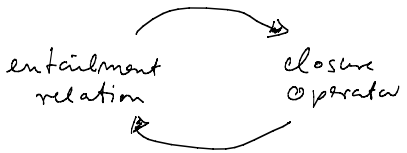
Closure Operators

[Tarski 1930]

$[A] := \{x \mid A \vdash x\}$ closure of A

Closure operator $[\cdot] \in \mathcal{P}\mathcal{X} \to \mathcal{P}\mathcal{X}$
has nice algebraic properties:

1) $A \subseteq [A]$ — expansivity
2) $A \subseteq B \Rightarrow [A] \subseteq [B]$ — monotonicity
3) $[[A]] = [A]$ — idempotence



$A \vdash x \Leftrightarrow x \in [A]$

Inference Systems

- Describe possible inferences (i.e., inference steps)

$$\frac{x_1 \quad \dots \quad x_n}{x}$$

premises / conclusion

- Yield compact entailment relations

$A \vdash x \iff x$ can be obtained from the statements in A by finitely many inferences

Definition

An inference system on $\mathcal{X}$ is a set of pairs (P, x) such that P is a finite subset of $\mathcal{X}$ and $x \in \mathcal{X}$.

Derivations

Let S be an inference system on $\mathcal{X}$

A **derivation of $x \in \mathcal{X}$ from $A \subseteq \mathcal{X}$ in S** is a tuple $(x_1, \dots, x_n)$ such that

1) $x_n = x$
2) $\forall i \in \{1, \dots, n\} : x_i \in A \vee \exists P \subseteq \{x_1, \dots, x_{i-1}\} : (P, x_i) \in S$

$A \vdash_S x :\iff \exists$ derivation of x from A in S

$\vdash_S$ is compact entailment relation

$\vdash_S$ is effective if $\mathcal{X}$ decidable and S semi-decidable

Closures

$S[A] := \{x \mid A \vdash_S x\}$ closure of A wrt S

A closed under $S :\iff \forall (P, x) \in S : P \subseteq A \Rightarrow x \in A$

A closed under $S \cong A$ invariant for S

$S[A]$ closed under S

Closure Theorem

S inference system

$A \subseteq Q$, Q closed under S $\}$ $S[A] \subseteq Q$

Proof is straightforward

Encapsulates induction on length of derivation

Cor: $S[A]$ is the least set that contains A and is closed under S

Cor: A closed under $S \iff A = S[A]$

Closure Theorem is useful for proofs

Claim: $A \vdash e \Rightarrow \exists$ conversion proof of e from A

Proof: By closure theorem it suffices to show

1) $\forall e \in A$: $\exists$ conversion proof of e from A

2) $\forall (P, e) \in BED$:

$(\forall e \in P \ \exists$ conversion proof of e from $A) \Rightarrow \exists$ conversion proof of e from A

$Q = \{ e \mid \exists$ conversion proof of e from $A \}$

Inference systems are a tool for the recursive definition of sets

$X = \mathbb{N}$

$S = \{ ((x, y), x \cdot y) \mid x, y \in \mathbb{N} \}$ $\quad \frac{x \quad y}{x \cdot y}$

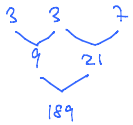
$S[A] \hat{=}$ all numbers that can be obtained from A by multiplication

$S[\emptyset] = \emptyset$

$S[\{2\}] = \{ 2^n \mid n \geq 1 \}$

$S[\{3,7\}] = \{3, 7, 9, 21, 27, 49, \ldots\}$

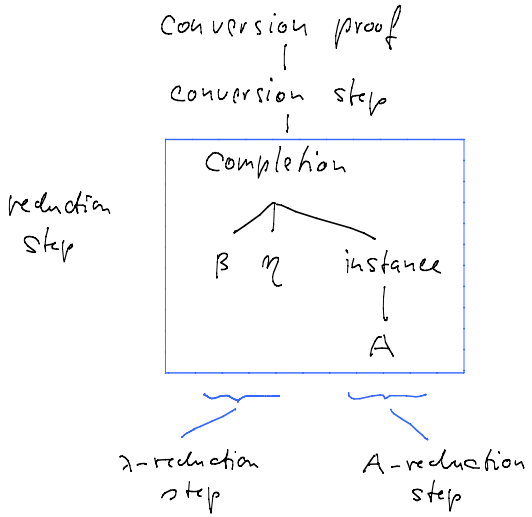
$= \{ 3^m \cdot 7^n \mid m+n \geq 1 \}$



Conversion Proofs

- alternative characterisation of deductive entailment
- based on replacement of equals with equals
- useful for hand calculations

$x = x \cdot 1$	Id	$x \cdot 1 = x$
$= x(x + \bar{x})$	Compl	$x + \bar{x} = 1$
$= xx + x\bar{x}$	Dist	$x(y+z) = xy + xz$
$= xx + 0$	Compl	$x\bar{x} = 0$
$= xx$	Id	$x + 0 = x$



Refl, Trans

Sym

CL, CR, ξ

Conversion Theorem

$$A \vdash \ell \iff \exists \text{ conversion proof of } \ell \text{ from } A$$

$(\ell_1, \ldots, \ell_n)$

$\ell = \ell_n$

inferences according to

Refl, Sym, Trans, CL, CR, ξ, β, η

inference system

$(r_1, \ldots, r_n)$

$\ell = (r_1, s_n)$

λ - conversions

A - conversions

rewrite system

Normal Form Theorem

[Church / Rosser / Tait]

$$\phi \vdash r = t \iff r, t \text{ have the same } \lambda\text{-NF}$$

similar result shown by Church/Rosser 1937

$r \xrightarrow{\lambda} t :\iff (r,t)$ is a λ-reduction step

$\xrightarrow{\lambda}$ is pronounced λ-reduction

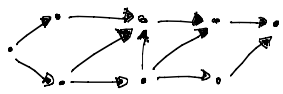
$\xrightarrow{\lambda}$ is confluent and terminating relation

termination shown by Tait 1967

Normal Form Theorem is straightforward consequence of Conversion Theorem and the confluence and termination of $\xrightarrow{\lambda}$

Confluent Relations

- Conversion proofs can be analyzed with binary relations on the set of terms replacement step $\rightsquigarrow (r, t)$
- $\longrightarrow \subseteq X^2 \times X^2$
- see $\longrightarrow$ as graph (possibly infinite)



- useful abstractions evolved over a long time summerized by Huet 1980
- suggested reading: Baader/Nipkow

Reflexive Transitive Closure $\to^*$

$\to \subseteq X \times X$

$x \to^* y \;:\Leftrightarrow\; \exists\, (x_1, \dots, x_n) \in X^*:\; x = x_1 \to \cdots \to x_n = y$

$\exists$ path from x to y, y reachable from x

$\to^*$ is reflexive, transitive and contains $\to$

$\to^0 := \{(x,x) \mid x \in X\}$

$\to^n := \to \circ \to^{n-1} \quad (n \geq 1)$

$$\to^* = \bigcup \{ \to^n \mid n \in \mathbb{N} \}$$

Our definition of $\to^*$ is intuitively pleasing but inconvenient for proofs. The characterization with $\to^n$ solves the problem.

Connection with inference systems

$\to$ reflexive $\Leftrightarrow$ $\to$ closed under $\dfrac{}{(x,x)}\ x \in X$

$\to$ transitive $\Leftrightarrow$ $\to$ closed under $\dfrac{(x,y)\quad (y,z)}{(x,z)}\ x,y,z \in X$

$\to^*$ is closure of $\to$ wrt

$\dfrac{}{(x,x)}\ x \in X$ and $\dfrac{(x,y)\quad (y,z)}{(x,z)}\ x,y,z \in X$

Characterization as inference closure

Proof

$\to^* \subseteq S[\to]$

We show $\forall n \forall x,y:\; x \to^n y \Rightarrow (x,y) \in S[\to]$

By induction on n.

Let $x \to^n y$

Case $n=0$. Then $x = y$ and hence $(x,y) \in S[\to]$

Case $n>0$. Then $x \to x' \to^{n-1} y$

$(x,x') \in S[\to]$

$(x',y) \in S[\to]$ induction

$(x,y) \in S[\to]$

$S[\to] \subseteq \to^*$

By Closure Theorem

1) $\to \subseteq \to^*$
2) $\to^*$ closed under reflexivity rule
3) $\to^*$ closed under transitivity rule

□

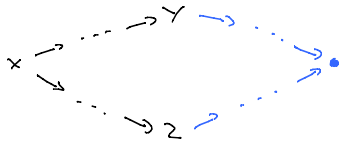
Corollary

$\to^*$ is the least reflexive and transitive relation that contains $\to$

$x \downarrow y \;:\Leftrightarrow\; \exists z:\ x \to^* z \wedge y \to^* z$

x, y joinable

$\to$ confluent $:\Leftrightarrow\; \forall x,y,z:\ x \to^* y \wedge x \to^* z \Rightarrow y \downarrow z$



$\to$ semi-confluent $:\Leftrightarrow\; \forall x,y,z:\ x \to y \wedge x \to^* z \Rightarrow y \downarrow z$

$\to$ locally confluent $:\Leftrightarrow\; \forall x,y,z:\ x \to y \wedge x \to z \Rightarrow y \downarrow z$

Conversion

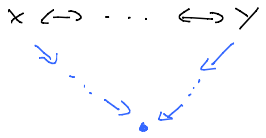
$x \leftrightarrow y \;:\Leftrightarrow\; x \to y \cup y \to x$

x, y one-step convertible

$x \leftrightarrow^* y \;\Leftrightarrow\; \exists (x_0, \dots, x_n) \in X^*:\ x = x_0 \leftrightarrow \cdots \leftrightarrow x_n = y$

x, y convertible

$\to$ Church Rosser $:\Leftrightarrow\; \forall x,y:\ x \leftrightarrow^* y \Rightarrow x \downarrow y$



$\to$ semi-confluent $\Rightarrow$ $\to$ Church Rosser

Proof Let $\to$ be semi-confluent.

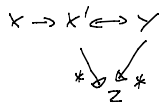
To show: $\forall n\ \forall x,y:\ x \leftrightarrow^n y \Rightarrow x \downarrow y$.

By induction on n. Let $x \leftrightarrow^n y$.

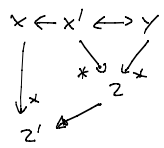
Case $n = 0$. Then $x = y$ and hence $x \downarrow y$.

Case $n > 0$. Then $x \leftrightarrow x' \leftrightarrow^{n-1} y$

Case 1



Case 2



induction
semi-confluence $\square$

Corollary

$\to$ Church-Rosser $\Leftrightarrow$ $\to$ confluent $\Leftrightarrow$ $\to$ semi-confluent

Termination

$\to$ terminating on x $:\Leftrightarrow$

$\neg \exists A \subseteq \mathcal{X} : \; x \in A \wedge \forall y \in A \; \exists z \in A : \; y \to z$

then exists no infinite path issuing from x

$\to$ terminating $:\Leftrightarrow$ $\to$ terminating for all $x \in \mathcal{X}$

Normal Forms

x NF $:\Leftrightarrow \neg \exists y : x \to y$ terminal node

x NF for y $:\Leftrightarrow$ x NF and $y \to^* x$ reachable terminal node

$\to$ terminating $\Rightarrow \forall x \in \mathcal{X}$: x has NF

$\to$ confluent $\Rightarrow \forall x \in \mathcal{X}$: x has at most one NF

If $\to$ terminating and confluent, then

1) $\forall x \in \mathcal{X}$: x has exactly one NF

2) $\forall x, y \in \mathcal{X}$: $x \leftrightarrow^* y \Leftrightarrow x, y$ have the same NF

Proof: Exercise!

Induction Theorem

Often called well-founded induction
First studied by Emmy Noether 1882 - 1935

If $\to$ terminating, then

$$\left(\forall x \in \mathcal{X} : \{y \mid x \to y\} \subseteq Q \Rightarrow x \in Q\right) \Rightarrow \mathcal{X} \subseteq Q$$

Proof By contradiction

Let $\forall x \in \mathcal{X}: \{y \mid x \to y\} \subseteq Q \Rightarrow x \in Q$ (1)

$x \in \mathcal{X}$, $x \notin Q$ (2)

Then x NF w.l.o.g.; $\to$ terminating, (1), (2)

$x \in Q$ (1)

□

Newman's Lemma

Newman 1942

$\to$ terminating and locally confluent
$\Rightarrow$ $\to$ confluent

Proof of Huet 1980 is a very nice demonstration of general induction

Proof Let $\to$ be terminating and locally confluent

Show $\forall x \; \forall y z \in \mathcal{X}: \; x \to^* y \wedge x \to^* z \Rightarrow y \downarrow z$

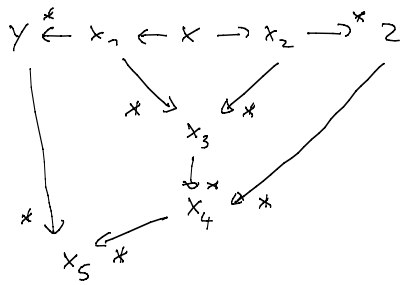
By induction on x wrt $\to$

$Q = \{x \mid \forall y z \in \mathcal{X}: \; x \to^* y \wedge x \to^* z \Rightarrow y \downarrow z\}$
show: $\{y \mid x \to y\} \subseteq Q \Rightarrow x \in Q$

Let $x \to^* y \wedge x \to^* z$

Case $x = y \vee x = z$

Case $y \;{}^{*}\!\leftarrow x_1 \leftarrow x \to x_2 \to^* z$



local confluence

induction for x_2

induction for x_1

□

Higher-Order Propositional Logic

- Categorical axiomatization of Boolean operations
- Non-algebraic deductions

2006-6-26+28, 2006-7-3

Specification PL

defined constants (can be eliminated)

$o : \mathcal{B}$ $\to : \mathcal{B} \to \mathcal{B} \to \mathcal{B}$		$\top = o \to o$	D⊤
		$\neg x = x \to o$	D¬
		$x \vee y = (x \to y) \to y$	D∨
		$x \wedge y = \neg(\neg x \vee \neg y)$	D∧
		$x \not\leftarrow y = \neg(y \to x)$	D↚
		$x \leftrightarrow y = (x \to y) \wedge (y \to x)$	D↔
		$x \not\leftrightarrow y = \neg(x \leftrightarrow y)$	D↮
$o \to x = \top$	Io		
$\top \to x = x$	I⊤		
$fo \to f\top \to fx = \top$	BCA		
$x \vee y = y \vee x$	Com		

$\mathcal{J}$: standard model of PL

PL categorical

- $\not\leftarrow$ serves as dual of $\to$
- Com semantically redundant
- only 4 essential axioms

Notation

Operator precedence

$\leftrightarrow$, $\not\leftrightarrow$

$\to$, $\not\leftarrow$

$\vee$

$\wedge$

$\doteq$, $\neq$ notational variants of $\leftrightarrow$, $\not\leftrightarrow$; highest precedence

all operators associate to the right, e.g.: $x \to y \to z$ means $x \to (y \to z)$

$\mathcal{J}$ means $\neg\neg$

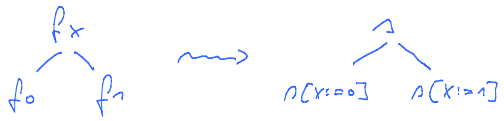
Modus Ponens

$\vdash_{PL}$

$\alpha \to \xi = \top$, $\alpha = \top$ $\quad \xi = \top$ MP

Use of BCA

$PL \vdash f_0 \to f_n \to fx = n$



$PL \vdash n[x:=0] \to n[x:=1] \to n = n$

Proof. $n[x:=0] \to n[x:=1] \to n$

$= t0 \to t1 \to tx$ $\quad t = \lambda x.n, \beta$

$= n$ $\quad$ BCA $\quad \square$

$n[x:=0] = 1,\ n[x:=1] = 1 \overset{PL}{\vdash} n[x:=t] = 1$

0-1-Theorem

A term n is pure if it has the form

$n = 0 \mid 1 \mid x \mid \neg n \mid n \circ n$

$x : \mathbb{B}$ $\qquad$ $\circ = \to \mid \leftrightarrow \mid \wedge \mid \vee \mid \leftarrow$

pure term $\hat{=}$ Boolean term

n pure and closed $\Rightarrow PL \vdash n=0 \vee PL \vdash n=1$

Proof. By induction on $|n|$.

n pure and closed $\Rightarrow (PL \models n=1 \iff PL \vdash n=1)$

Corollary to the statement above.

Completeness for pure terms

n pure $\Rightarrow (\mathcal{T} \models n=1 \iff PL \vdash n=1)$

Proof. Let n be pure.

"$\Leftarrow$" Let $PL \vdash n=1$. Then $PL \models n=1$ by Soundness and $\mathcal{T} \models n=1$ by $\mathcal{T} \models PL$.

"$\Rightarrow$" Show $\forall\, \Delta$ pure: $\mathcal{T} \models n=1 \Rightarrow PL \vdash n=1$

by induction on number n of variables in n.

Let n be pure and $\mathcal{T} \models n=1$. (1)

Case $n=0$. By Soundness $PL \not\vdash n=0$. Hence $PL \vdash n=1$ $\quad$ 0-1-Theo.

Case $n>0$. Let $x \in \mathcal{V}n$. Then

$\mathcal{T} \models n[x:=0] = 1,\ \mathcal{T} \models n[x:=1] = 1$ $\quad$ (1)

$PL \vdash n[x:=0] = 1,\ PL \vdash n[x:=1] = 1$ $\quad$ Induction

$PL \vdash n = 1$ $\quad$ BCA $\quad \square$

$n \wedge t = 1 \overset{PL}{\vdash} n=1, t=1$ $\qquad$ And

Proof "$\dashv$" $\quad$ 0-1-Theo

"$\vdash$" $\quad PL \vdash x \wedge y \to x = 1$ $\quad$ Completeness

$PL \vdash x \wedge y \to y = 1$ $\quad$ Completeness

$n \wedge t = 1 \overset{PL}{\vdash} n=1, t=1$ $\quad$ Generativity, MP $\quad \square$

$x = t \overset{PL}{\dashv\vdash} x \doteq t = 1$ **Eq**

This is the only result for which Com is needed, and Com is included to get this result

Proof $\vdash$ Follows with $PL \vdash x \doteq x = 1$

$\dashv$ $x \doteq t = 1 \overset{PL}{\dashv\vdash} (x \to t) \wedge (t \to x) = 1$ by D⇔

$\overset{PL}{\dashv\vdash} x \to t = 1,\ t \to x = 1$ by And (1)

$\overset{PL}{\dashv\vdash} x = t$ by (2)

(2) $x = 1 \to x$ I1

$= (t \to x) \to x$ (1)

$= t \vee x$ Dv

$= x \vee t$ Com

$= (x \to t) \to t$ Dv

$= 1 \to t$ (1)

$= t$ I1 □

Completeness for pure equations

e pure $\Rightarrow$ $(\mathcal{T} \models e \iff PL \vdash e)$

- $x = t$ pure if x, t pure
- pure equation = Boolean equation

Proof Compl pure terms, Eq, Soundness □

Cor $PL \vdash BA$ $\wedge = \cdot$, $\vee = +$

Can now reuse deduction results for BA

Duality

$\delta 0 = 1$	$\delta 1 = 0$
$\delta \wedge = \vee$	$\delta \vee = \wedge$
$\delta \to = \leftarrow$	$\delta \leftarrow = \to$
$\delta \leftrightarrow = \leftrightarrow$	$\delta \not\leftrightarrow = \not\leftrightarrow$

$\delta(\delta x) = x$

$PL \vdash \delta(PL)$

$PL \vdash e \iff PL \vdash \delta e$

Tautologies and Practical Advice

$Taut := \{ e \text{ pure} \mid \mathcal{T} \models e \}$

$e \in Taut \iff e$ pure $\wedge\ PL \vdash e$

To prove $e \in Taut$, BA-style reasoning works well

To prove $PL \vdash e$, a combination of
- Eq
- BCA
- Taut-conversions

may work well

Boolean Replacement

$$PL \vdash x \doteq y \to fx = x \doteq y \to fy \qquad \text{BRep}$$

- Internal formulation of "Replacement of equals with equals"
- Useful for conversion proofs
- Not pure

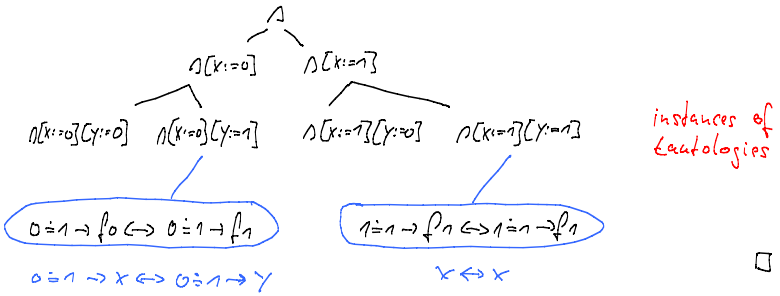
Proof of BRep

Claim $PL \vdash x \doteq y \to fx = x \doteq y \to fy$

Proof Eq, BCA, Taut

Let $\alpha = x \doteq y \to fx \leftrightarrow x \doteq y \to fy$

Show $PL \vdash \alpha = \top$ with BCA and Taut (+MP, Gen)



□

Variants of BRep

$$PL \vdash x \doteq y \wedge fx = x \doteq y \wedge fy$$

$$PL \vdash x \doteq y \to fx \to fy = \top$$

Generalized BCA (Proof Technique)

Show $PL \vdash \alpha = \top$ ($PL \vdash e$ with Eq)

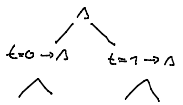
by iterated case analysis with the tautologies

$x = (y \doteq 0 \to x) \wedge (y \doteq 1 \to x)$ GBCA

$x \to y \wedge z = (x \to y) \wedge (x \to z)$

and BRep (exploiting the premises introduced by GBCA)

and Taut-Conversion steps



goal

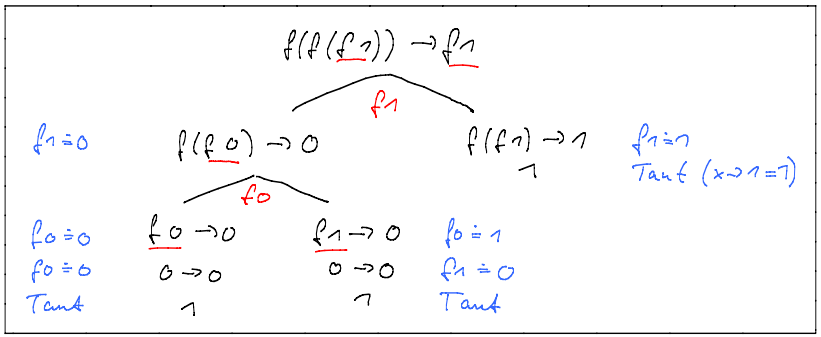
subgoals

- simplify goals with BRep and Taut-Conversion
- results in a conversion proof

Example

Claim $PL \vdash f(f(fx)) \to fx = x$

Proof



red underline $\_$ $\hat{=}$ BRep-conversion

Deductivity

- Tool for proving implications, proof transformer
- Relates external $\vdash$ to internal $\to$
- External version of BRep

$$s \text{ closed} \wedge A, s \doteq x \vdash^{PL} t = x \Rightarrow A \vdash^{PL} s \to t = x$$

Proof. Let s closed, $A, s \doteq x \vdash^{PL} t = x$.

∃ conversion proof of $t = x$ from $A, s = x, PL$

∃ conversion proof of $s \doteq x \to t = s \doteq x \to x$ from A, PL

- s closed $\Rightarrow$ s is the only instance of s
- Replacement $s \leadsto x$ justified by BRep

- $BA \vdash s \doteq x \to t = s \to t$
- $BA \vdash s \doteq x \to x = x$ □

Consistency and Satisfiability

A consistent $:\Leftrightarrow \neg \exists x, y: A \vdash x = y \wedge x \neq y$

A satisfiable $:\Leftrightarrow A$ has a proper model

$$A \text{ satisfiable} \Rightarrow A \text{ consistent}$$

If s closed, then

$$A \vdash^{PL} s = x \iff A, \bar{s} = x \vdash^{PL} 0 = x$$
$$\iff PL \cup A \cup \{\bar{s} = x\} \text{ inconsistent}$$

Follows with Deductivity

Variables versus parameters

$A \vdash \ell$, $A \models \ell$

- Can always assume A closed
- Can always assume ℓ closed*

$\ell = (x \doteq y \to fx = x \doteq y \to fy)$

$\Theta = \{x := a, y := b\}$

$A \vdash \Theta \ell$ by Generativity

$\psi = \{a := x, b := y\}$

$\psi A \vdash \psi(\Theta \ell)$ by Stability

$A \vdash \ell$ since $\psi A = A$ and $\psi(\Theta \ell) = \ell$

* provided A frugal, i.e., for every type there are infinitely many constants not used in A

Propositional Completeness

Plain terms, equations: like pure but parameters in place of variables: $r = a \mid \neg r \mid r \circ r$ where $a : B$

$$A, \ell \text{ plain} \Rightarrow \left(A \models^{PL} \ell \Leftrightarrow A \vdash^{PL} \ell \right)$$

Suffices to show

$$A \text{ plain} \wedge PL \cup A \text{ consistent} \Rightarrow PL \cup A \text{ satisfiable}$$

Proof can be based on ACCs, see [Fitting], [Andrews]
For finite A, the above follows from previous results.

Propositional Compactness

If A plain, then

$PL \cup A$ satisfiable $\Leftrightarrow \forall A' \subseteq A$ finite: $PL \cup A'$ satisfiable

straightforward consequence of Prop. Completeness

Higher-Order Predicate Logic

- PL + axiomatization of quantifiers
- Focus: Deduction techniques for quantifiers

2006-7-3+5

Specification QL

Extends	PL	
Constants	$\forall_T, \exists_T : (T \to B) \to B$	
Axioms	$\forall(\lambda x.r) = r$	$\forall\eta$
	$\forall f \to fx = r$	$\forall I$
	$\exists(\lambda x.0) = 0$	$\exists 0$
	$fx \to \exists f = r$	$\exists I$

polymorphic (Predicate Logic: T is a fixed sort)

Instantiation

axioms are schematic, x, f are variables

Notation: $\forall x.r \rightsquigarrow \forall(\lambda x.r)$
$\exists x.r \rightsquigarrow \exists(\lambda x.r)$

Duality

$\delta 0 = 1$	$\delta 1 = 0$
$\delta\wedge = \vee$	$\delta\vee = \wedge$
$\delta\to = \leftarrow$	$\delta\leftarrow = \to$
$\delta\leftrightarrow = \leftrightarrow$	$\delta\not\leftrightarrow = \leftrightarrow$
$\delta\forall_T = \exists_T$	$\delta\exists_T = \forall_T$

$\delta(\delta r) = r$

$QL \vdash \delta(QL)$

$QL \vdash e \Leftrightarrow QL \vdash \delta e$

Alternative: Axiomatize $\exists$ as defined constant: $\exists f = \overline{\forall x.\overline{fx}}$

Golden Rule

$r \to t = 1 \quad \overset{BA}{\vdash\!\dashv} \quad r = r \wedge t \quad \overset{BA}{\vdash\!\dashv} \quad t = r \vee t$ GR

Proof. Follows from:

1) $BA \vdash x \to y = x \Leftrightarrow x \wedge y$
2) $BA \vdash x \to y = y \Leftrightarrow x \vee y$
3) $r = t \overset{BA}{\vdash\!\dashv} r \Leftrightarrow t = 1$ □

Examples

$QL \vdash \forall f = \forall f \wedge fx$ by $\forall I$, GR

$QL \vdash \exists f = \exists f \vee fx$ by $\exists I$, GR

Drop Laws

can eliminate / introduce quantifiers

$\varphi[x:=t] = \top \overset{QL}{\vdash\!\dashv} \exists x.\varphi = \top$ — Gen∃

$\varphi = \top \overset{QL}{\vdash\!\dashv} \forall x.\varphi = \top$ — Gen∀ — Generalisation

$QL \vdash p = \exists x.p$ — ∃E — Elimination

$QL \vdash p = \forall x.p$ — ∀E

Proof of Gen ∀

Claim $\forall x.\varphi = \top \overset{QL}{\vdash\!\dashv} \varphi = \top$

Proof "⊣"

$\forall x.\varphi = \forall x.\top$ — $\varphi = \top$, ∀⊤

$= \top$

"⊢"

$\varphi = \top \wedge \varphi$ — Taut

$= (\forall x.\varphi) \wedge \varphi$ — $\forall x.\varphi = \varphi$

$= (\forall x.\varphi) \wedge (\lambda x.\varphi)x$ — β

$= \forall x.\varphi$ — ∀I, GR

$= \top$ — $\forall x.\varphi = \varphi$ □

Pull Laws (∧,∨)

The following equations are derivable in QL

∀∧	$\forall f \wedge q = \forall x.fx \wedge q$	∃∨	$\exists f \vee q = \exists x.fx \vee q$
∀∨	$\forall f \vee q = \forall x\, fx \vee q$	∃∧	$\exists f \wedge q = \exists x.fx \wedge q$
∀∧'	$\forall f \wedge \forall g = \forall x.fx \wedge gx$	∃∨'	$\exists f \vee \exists g = \exists x.fx \vee gx$

Proof of ∀∧

Claim $QL \vdash \forall x.fx \wedge q = \forall f \wedge q$

Proof $(\forall x.fx \wedge q) \leftrightarrow \forall f \wedge q$ — Eq, BCA, q

$(\forall x.fx \wedge \varphi) \leftrightarrow \forall f \wedge \varphi$ | $(\forall x.fx \wedge \top) \leftrightarrow \forall f \wedge \top$

Taut: $(\forall x.\varphi) \leftrightarrow \varphi$ | $\forall x.fx \leftrightarrow \forall f$ — Taut

∀I, GR: $(\forall x.\varphi) \wedge (\lambda x.\varphi)x \leftrightarrow \varphi$ | $\forall f \leftrightarrow \forall f$ — η

β: $(\forall x.\varphi) \wedge \varphi \leftrightarrow \varphi$ | $\top$ — Taut

Taut: $\top$

□

De Morgan Laws (dM)

$$\overline{\forall x.\, fx} = \exists x.\, \overline{fx} \qquad \overline{\exists x.\, fx} = \forall x.\, \overline{fx}$$

Proof with MoC and Duality

Proof of dM

MoC $\quad \bar{n} = t \overset{BA}{\mapsto} n \wedge t = 0,\ n \vee t = 1$

$\overline{\exists x.\, fx} = \forall x.\, \overline{fx}$ — MoC

	$(\exists x.\, fx) \wedge \forall x.\, \overline{fx}$	$(\exists x.\, fx) \vee \forall x.\, \overline{fx}$
$\exists\wedge$	$\exists x.\, fx \wedge \forall x.\, \overline{fx}$	$(\exists x.\, fx) \vee \overline{fx}$
$\forall I, GR$	$\exists x.\, fx \wedge (\forall x.\, \overline{fx}) \wedge \overline{fx}$	$\exists f \vee \overline{fx}$
Taut	$\exists x.\, 0$	$fx \to \exists f$
$\exists 0$	0	1
		$\forall\vee$, Gen $\forall$
		η
		Taut
		$\exists I$

Pull Laws ($\to$)

The following equations are decidable in QL

$$q \to \forall f = \forall x.\, q \to fx$$
$$q \to \exists f = \exists x.\, q \to fx$$
$$\forall f \to q = \exists x.\, fx \to q$$
$$\exists f \to q = \forall x.\, fx \to q$$
$$\forall f \to \exists g = \exists x.\, fx \to gx$$

Proof. Use pull laws for $\wedge, \vee$ and dM.

Turing's Law

$$QL \vdash \overline{\exists x \forall y.\, fxy \leftrightarrow \overline{fyy}} = 1$$

$f : T \to T \to B$

1) There is no barber who shaves everyone who doesn't shave himself
2) There is no TM that halts on the rep of a TM y if and only if y doesn't halt on its own rep
3) There is no set that contains all sets that don't contain themselves

Turing's Law

$$\mathrm{QL} \vdash \overline{\exists x\, \forall y.\ fxy \leftrightarrow \overline{fyy}} = \top$$

$f : T \to T \to B$

Proof — backward proof!

$\overline{\exists x\, \forall y.\ fxy \leftrightarrow \overline{fyy}}$

$\forall x\, \exists y.\ \overline{fxy \leftrightarrow \overline{fyy}}$ dM

$\exists y.\ fxy \leftrightarrow fyy$ Gen ∀, Taut

$fxx \leftrightarrow fxx$ Gen ∃

$\top$ Taut

Cantor's Law

Let X be a set. Then there exists no surjective function $X \to \mathcal{P} X$

$X \rightsquigarrow$ type T

$\mathcal{P} X \rightsquigarrow$ type $T \to B$

$\exists f\, \forall g\, \exists x.\ fx \doteq g$

$T \to (T \to B)$ $\quad T \to B$ $\quad T$

Cantor's Law (for $X = B$)

$$\mathrm{PL} \vdash \overline{\exists f\, \forall g\, \exists x\, \forall y.\ fxy \leftrightarrow gy} = \top$$

$f : B \to B \to B$
$g : B \to B$
$x, y : B$

Proof

$\overline{\exists f\, \forall g\, \exists x\, \forall y.\ fxy \leftrightarrow gy}$

$\forall f\, \exists g\, \forall x\, \exists y.\ fxy \leftrightarrow \overline{gy}$ dM, Taut

$\forall x\, \exists y.\ fxy \leftrightarrow \overline{(\lambda y.\ \overline{fyy})\, y}$ Gen ∀, Gen ∃ $(g := \lambda y.\ \overline{fyy})$

$\exists y.\ fxy \leftrightarrow fyy$ Gen ∀, β, Taut

$fxx \leftrightarrow fxx$ Gen ∃

$\top$ Taut

Predicate Logic with Choice

- PL + axiomatization of choice operator
- $CL \vdash QL \vdash PL \vdash BA$
- Skolem quantifier elimination

2006-7-10

Specification CL

Extends	PL		
Constants	$C_\tau : (\tau \to \beta) \to \tau$		choice
Axioms	$fx \to f(Cf) = \top$	CI	
Derived Constants	$\overline{C}f = C(\lambda x.\, \overline{fx})$	D$\overline{C}$	dual choice
	$\exists f = f(Cf)$	D$\exists$	
	$\forall f = f(\overline{C}f)$	D$\forall$	

C not uniquely determined, 4 possibilities for B

$CL \vdash QL$ Can reuse proof techniques for QL

Claim $CL \vdash \forall I$

Proof
$$\begin{aligned} \forall f \to fx &= \overline{fx} \to \overline{\forall f} && \text{Taut} \\ &= \overline{fx} \to \overline{f(C(\lambda x.\, \overline{fx}))} && D\forall, D\overline{C} \\ &= \wedge x \to \wedge(C\wedge) && \wedge = \lambda x.\, \overline{fx},\ \beta \\ &= \top && CI \end{aligned}$$
□

Exercise Prove $QL \vdash \exists 0, \exists I, \forall \top$.

Duality

$\delta 0 = \top$	$\delta \top = 0$
$\delta \wedge = \vee$	$\delta \vee = \wedge$
$\delta \to = \not\leftarrow$	$\delta \not\leftarrow = \to$
$\delta \leftrightarrow = \not\leftrightarrow$	$\delta \not\leftrightarrow = \leftrightarrow$
$\delta C = \overline{C}$	$\delta \overline{C} = C$
$\delta \forall_\tau = \exists_\tau$	$\delta \exists_\tau = \forall_\tau$

$\delta(\delta\alpha) = \alpha$

$CL \vdash \delta(CL)$

$CL \vdash \varepsilon \Leftrightarrow CL \vdash \delta\varepsilon$

Claim $CL \vdash fx \leftrightarrow f(\bar{C} f) = 0$ dual of CI

Proof
$$\begin{aligned} & fx \leftrightarrow f(\bar{C} f) \\ = \; & \overline{\overline{fx} \to \overline{f(\bar{C} f)}} && D\leftrightarrow, \text{Taut} \\ = \; & \overline{\neg x \to \neg(C \neg)} && \neg = \lambda x.\, \overline{fx},\ D\bar{C},\ \beta \\ = \; & \overline{\top} && CI \\ = \; & 0 && \text{Taut} \end{aligned}$$
□

Skolem's Law

$$CL \vdash \forall x\, \exists y.\ fxy = \exists g\, \forall x.\ fx(gx) \qquad \text{Sko}$$

$CL \models \text{Sko}$
$CL \not\vdash \text{Sko}$ (difficult to show)

Proof of Skolem's Law

$\exists q$, $D\leftrightarrow$, And

$(\underbrace{\forall x \exists y.\ fxy}_{\exists(fx)}) \to \exists g\, \forall x.\ fx(gx)$
$\exists g.\ (\forall x.\ fx(C(fx))) \to \forall x.\ fx(gx)$ — η; $\forall\exists$, Pull$\to$; Gen $\exists$: $g := \lambda x.\, C(fx)$; β; Taut
$\top$

$(\exists g\, \forall x.\ fx(gx)) \to \forall x \exists y.\ fxy$
$\exists z\, \exists y.\ fz(gz) \to fxy$ — Pull$\to$, Gen $\forall$
$fx(gx) \to fx(gx)$ — Gen $\exists$: $z := x$, $y := gx$; Taut
$\top$

Prenex Form

$\underbrace{Q_1 x_1 \dots Q_n x_n}_{\text{quantifier prefix}}.\ \underbrace{\wedge}_{\text{body}}$ where $n \geq 0$ and $\exists, \forall, C \notin \mathcal{V}(\wedge)$ and $\wedge : \beta$

Skolem Form

$\exists x_1 \dots \exists x_m \forall y_1 \dots \forall y_n.\ \wedge$ where $m, n \geq 0$ and $\exists, \forall, C \notin \mathcal{V}(\wedge)$ and $\wedge : \beta$

Equivalent Skolem form can be obtained from prenex form with Skolem's law

Prenex form often can be obtained with pull laws and DM

Skolem Quantifier Elimination (Skolemization) [1928]

Let A be in prenex form and frugal.
Then there exists for each Λ an A' such that

1) $\forall, \exists, C \notin \mathcal{V}(A')$

2) $A \vdash^{CL} \Lambda = \tau \iff A' \vdash^{CL} \Lambda = \tau$

A first-order $\rightsquigarrow$ A' algebraic

Lemma

If $\exists x.\Lambda$ closed and a does not appear in A, Λ and $t = \tau$, then

$A, \exists x.\Lambda = \tau \vdash^{QL} t = \tau \iff A, \Lambda[x := a] = \tau \vdash^{QL} t = \tau$

Outermost existential quantifiers can be eliminated by introducing so-called Skolem constants

$A, \exists x.\Lambda = \tau \vdash^{QL} t = \tau$

$\iff A \vdash^{QL} (\exists x.\Lambda) \to t = \tau$ Deduktivity, $\exists x.\Lambda$ closed

$\iff A \vdash^{QL} \forall x.\, \Lambda \to t = \tau$ Pull $\to$

$\iff A \vdash^{QL} \Lambda \to t = \tau$ Gen $\forall$

$\iff A \vdash^{QL} \Lambda[x := a] \to t = \tau$ Stability, $a \notin \mathcal{V}(A, \Lambda, t, \tau)$, $x \notin \mathcal{V} t$

$\iff A, \Lambda[x := a] = \tau \vdash t = \tau$ Deduktivity, $\Lambda[x := a]$ closed

Example

$A = \{\, \forall x.\, fx \to \exists f \wedge gx = \tau \,\}$ f, g constants

$\forall x.\, fx \to \exists f \wedge gx$

$= \forall x\, \exists y.\ fx \to fy \wedge gx$ Pull

$= \exists z\, \forall x.\ fx \to f(zx) \wedge gx$ Skolem

$A' = \{\, fx \to f(ax) \wedge gx = \tau \,\}$ a new constant

Predicate Logic with Identity

- $QL' = QL + \{\doteq, \neq\}$ as derived constants
- $QL'' = QL' +$ Extensionality

2006-7-12

$QL' = QL + \{\doteq, \neq\}$

$$x \doteq y = \forall f.\ fx \to fy \qquad D\doteq \qquad \text{Leibniz}$$
$$x \neq y = \neg(x \doteq y) \qquad D\neq$$

Duality of QL preserved with

$S(\doteq) = (\neq) \qquad S(\neq) = (\doteq)$

$$QL \vdash \forall f.\ fx \to fy = \forall f.\ fx \leftrightarrow fy$$

Basic Identity Laws

The followings equations are deducible in QL'

Ref	$x \doteq x = \top$	
Sym	$x \doteq y = y \doteq x$	
Trans	$x \doteq y \to y \doteq z \to x \doteq z = \top$	
CR	$x \doteq y \to gx \doteq gy = \top$	
CL	$g \doteq h \to gx \doteq hx = \top$	consequence of CR
Rep	$x \doteq y \to fx = x \doteq y \to fy$	generalizes BRep
$D\doteq'$	$x \doteq y = \forall f.\ fx \leftrightarrow fy$	

Claim $QL' \vdash x \doteq y = y \doteq x$

Proof Eq, D⇔, And

$$\begin{aligned}
& x \doteq y \to y \doteq x \\
= \quad & (\forall f.\ fx \to fy) \to \forall g.\ gy \to gx && D\doteq \\
\vdash \quad & \exists f.\ (fx \to fy) \to gy \to gx && \text{Pull, Gen}\forall \\
= \quad & \exists f.\ (fx \to fy) \to \overline{gx} \to \overline{gy} && \text{Taut (Contraposition)} \\
\vdash \quad & (\overline{gx} \to \overline{gy}) \to \overline{gx} \to \overline{gy} && \text{Gen}\exists: f = \lambda x.\, gx,\ \beta \\
= \quad & \top && \text{Taut} \qquad \square
\end{aligned}$$

Rep provides for capture-free replacement

$$QL' \vdash s_1 \doteq s_2 \to t[x := s_1] = s_1 \doteq s_2 \to t[x := s_2]$$

$(\lambda x.\, t)\, s_1$ $\qquad$ $(\lambda x.\, t)\, s_2$

Extensionality

Ext $\quad (\forall x.\, fx \doteq gx) \to f \doteq g = \top$

$QL' \vdash Ext$ — obvious

$QL' \nvdash Ext$ — difficult (uses non-standard interpretations)

$CL' \nvdash Ext$ — conjecture

$QL'' := QL' \cup Ext$ $\qquad$ $CL'' := CL' \cup Ext$

Equivalent variants of Ext

$$\forall x.\, fx \doteq gx = f \doteq g \quad \overset{QL'}{\vdash\!\dashv} \quad Ext$$

$$\forall x.\, s \doteq t = (\lambda x.s) \doteq (\lambda x.t) \quad \overset{QL'}{\vdash\!\dashv} \quad Ext$$

Ext provides for replacement with legal capture

$$QL'' \vdash (\forall x.\, s \doteq t) \to f(\lambda x.\, gs) \doteq f(\lambda x.\, gt) = \top$$

Ext / CR / β

$(\lambda x.s) \doteq (\lambda x.t)$ $\qquad$ $f(\lambda x.\, g((\lambda x.s)\, x))$

$(\lambda h.\, f(\lambda x.\, g(h\, x)))\, (\lambda x.s)$

Relationship to ξ $\quad \dfrac{s = t}{\lambda x.s = \lambda x.t}$ at meta level

First-Order Predicate Logic (FOPL)

- Fragment of predicate logic with complete deduction
- Thoroughly studied
- Most textbook present first-order predicate logic only

2006-7-17+19

First-Order Formulas (fofs)

α atomic and algebraic formula

$\sigma = \alpha \mid \neg\sigma \mid \sigma\wedge\sigma \mid \sigma\vee\sigma \mid \forall x.\sigma \mid \exists x.\sigma$

where x non-functional

normal fof: closed fof where $\neg$ is only applied to atomic formulas

Formulas

Formula: term of type B

Atomic formula: formula such that no proper subterm is a formula

S set of formulas

$S \vdash^{A} \sigma :\Leftrightarrow \{\sigma=\tau \mid \tau\in S\} \vdash^{A} \sigma=\tau$

$S \not\vdash^{A} \sigma :\Leftrightarrow \{\sigma=\tau \mid \tau\in S\} \not\vdash^{A} \sigma=\tau$

S A-consistent $:\Leftrightarrow S \not\vdash^{A} 0$

S A-satisfiable $:\Leftrightarrow \{\sigma=\tau \mid \sigma\in S\}\cup A$ has a model $\mathcal{D}$ s.t. $\mathcal{D}0 \neq \mathcal{D}\tau$

Important Results for FOPL

- If S,σ first-order, then
  $S \vdash^{QL'} \sigma \Leftrightarrow S \models^{QL'} \sigma$
  Completeness
  Gödel 1929

- $\{\sigma \mid \emptyset \vdash^{NL} \sigma,\ \sigma$ fof licensed by NL$\}$
  is not semi-decidable where
  NL is axiomatization of $B,\mathbb{N}$
  with $\sigma\tau, \neg, \forall_{\mathbb{N}}, 0, 1, +, \cdot, \doteq_{\mathbb{N}}, \leq$
  Incompleteness
  Gödel 1931

- $\{s \mid \emptyset \models^{QL'} \sigma,\ \sigma$ fof$\}$ undecidable
  Undecidability
  Church 1936

Model Existence Theorem (MET)

If S set of normal fofs, then S QL'-consistent $\Rightarrow$ S QL'-satisfiable

Corollary (Completeness)

If S, A are first-order, then $S \models^{QL'} A \Leftrightarrow S \vdash^{QL'} A$

Follows with deductivity
$S \vdash^{QL'} A \Leftrightarrow S, \bar{A} \vdash^{QL'} \bot$

- First MET shown by Kurt Gödel, 1929, in his Doctoral Thesis, University of Vienna, "Über die Vollständigkeit des Logikkalkuls"
- Modern proofs
  - Henkin 1949
  - Smullyan 1963
- Textbooks
  - Andrews 2002
  - Fitting 1996

Proof of MET, Ideas

Idea: if S unsat, a conflict will appear after finitely many consequences are added

- Suffices to consider only normal fofs
- Unsatisfiability can be shown with finitely many steps, i.e., a refutation
- Leads to a proof method known as semantic tableau

1) Identify obvious conflicts such that
   S contains a conflict $\Rightarrow$ S unsat
2) Add consequences to S such that
   a) S unsat $\Leftrightarrow$ $S \cup \{A\}$ unsat — linear consequence
   b) S unsat $\Leftrightarrow$ $S \cup \{A_1\}$ unsat $\wedge$ $S \cup \{A_2\}$ unsat — binary consequence, leads to branching
3) Unsatisfiability of S can be made explicit (i.e., through conflict) by adding finitely many consequences

Refutation of $(a \vee a) \wedge (\bar{a} \vee \bar{a})$

partial refutations are called tableaus

1	$(a \vee a) \wedge (\bar{a} \vee \bar{a})$	
2	$a \vee a$	(1)
3	$\bar{a} \vee \bar{a}$	(1)
4	a $\quad$ a	(2)
5	$\bar{a}$ $\bar{a}$ $\quad$ $\bar{a}$ $\bar{a}$	(3)

$\{(a \vee a) \wedge (\bar{a} \vee \bar{a}),\ a \vee a,\ \bar{a} \vee \bar{a},\ a,\ \bar{a}\}$

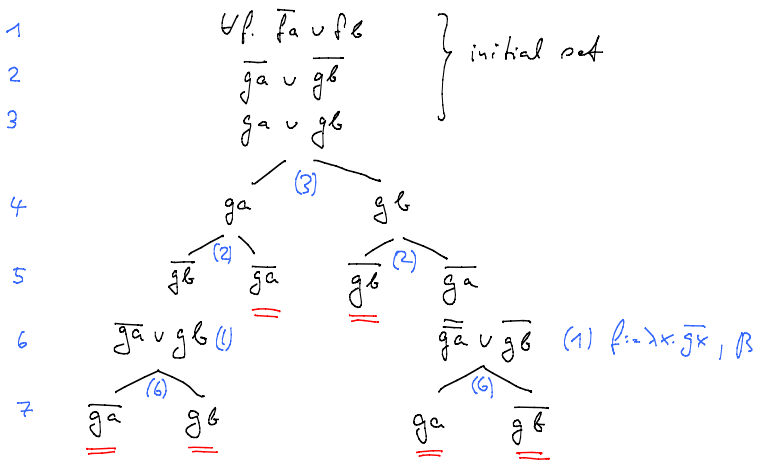
Refutation of $\forall x.\ \overline{fxa} \wedge fbx$

1	$\forall x.\ \overline{fxa} \wedge fbx$	
2	$\overline{fba} \wedge fbb$	(1)
3	$\overline{faa} \wedge fba$	(1)
4	$\overline{fba}$	(2)
5	fba	(3)

Dual: $\exists x.\ fxa \rightarrow fbx$

$QL' \vdash A = \bot \Leftrightarrow QL' \vdash \bar{A} = \top$

Refutation of a higher-order set



Trivial Sets

S is trivial if it contains one of the following conflicts:

1) $\bot \in S$
2) $\overline{\top} \in S$
3) $\exists \alpha:\ \alpha \in S \wedge \overline{\alpha} \in S$
4) $\exists \epsilon:\ \overline{\epsilon \doteq \epsilon} \in S$

S trivial $\Rightarrow$ S unsatisfiable

Consistency Classes (CCs)

- A CC is a set of non-trivial sets of normal fofs satisfying certain conditions
- Idea: S has a refutation $\Leftrightarrow$ no CC contains S
- CCs are easier to formalize than refutations
- MET: Every member of a CC is QL'-satisfiable

Definition of Consistency Classes

A set Ψ of non-trivial sets of normal first-order formulas is a consistency class if

1) $\alpha \wedge t \in S \in \Psi \Rightarrow S \cup \{\alpha, t\} \in \Psi$
2) $\alpha \vee t \in S \in \Psi \Rightarrow S \cup \{\alpha\} \in \Psi \vee S \cup \{t\} \in \Psi$

3) $\forall x.\alpha \in S \in \Psi \wedge \alpha[x := t]$ first-order $\Rightarrow S \cup \{\alpha[x := t]\} \in \Psi$
4) $\exists x.\alpha \in S \in \Psi \wedge \alpha[x := a]$ first-order $\wedge\ a \notin \mathcal{N}S \cup B \Rightarrow S \cup \{\alpha[x := a]\} \in \Psi$

5) $S \in \Psi \wedge \alpha \doteq \alpha$ first-order $\Rightarrow S \cup \{\alpha \doteq \alpha\} \in \Psi$
6) $\{\alpha_1 \doteq \alpha_2,\ t[x := \alpha_1]\} \subseteq S \in \Psi \Rightarrow S \cup \{t[x := \alpha_2]\} \in \Psi$

Smullyan's Model Existence Theorem [1963]

Every frugal member of a CC is satisfiable

S frugal if for every type there are infinitely many constants not occurring in S

First-Order Compactness

Let S be set of first-order formulas. Then:
Every finite subset of S satisfiable $\Rightarrow$ S satisfiable

Proof.

1) $\{S \mid S$ normal and first-order and every finite subset of S is satisfiable$\}$ is a consistency class
2) If S frugal, claim follows with Smullyan's MET
3) If S not frugal, we apply bijective constant renaming Θ such that ΘS frugal and exploit stability. □

First-Order Completeness

$\{S \mid S$ normal first-order $\wedge$ S QL'-consistent$\}$ is a consistency class

Proof requires arguments like the following

- $S, \alpha\vee\beta \vdash^{QL'} \sigma \iff S,\alpha \vdash^{QL'} \sigma \wedge S,\beta \vdash^{QL'} \sigma$
  Deducticity + tautology $(\chi\to\zeta)\wedge(\gamma\to\zeta) = \chi\vee\gamma\to\zeta$
- $S, \exists x.\alpha \vdash^{QL'} \sigma \iff S, \alpha[x:=a] \vdash^{QL'} \sigma$
  if $a \notin \mathcal{V}(S,\alpha)\cup\mathcal{B}$
  see Skolemization

Saturated Sets (S set of normal Pofs)

S saturated if the following conditions are satisfied:

1) $\alpha\wedge\beta \in S \Rightarrow \alpha\in S \wedge \beta\in S$
2) $\alpha\vee\beta \in S \Rightarrow \alpha\in S \vee \beta\in S$
3) $\exists x.\alpha\in S \Rightarrow \exists t : \alpha[x:=t]\in S$
4) $\forall x.\alpha\in S \wedge \alpha[x:=t]$ first-order $\Rightarrow \alpha[x:=t]\in S$
5) $\alpha\doteq\alpha$ first-order $\Rightarrow \alpha\doteq\alpha\in S$
6) $\alpha_1\doteq\alpha_2\in S \wedge \beta[x:=\alpha_1]\in S \Rightarrow \beta[x:=\alpha_2]\in S$

Hintikkas Model Existence Theorem [1955]

S saturated and non-trivial $\Rightarrow$ S satisfiable

term models are due to Herbrand and Birkhoff

Proof. Construct term model as follows

$[A] = \{t \mid A \doteq t \in S\}$ equivalence class

$\mathcal{C}_C = \{[A] \mid A \doteq B \in S \wedge B : C\}$ if $C \neq \mathcal{B}$

$\mathcal{C}_c\,[A_1] \ldots [A_n]$ where $c : C_1 \to \ldots \to C_n \to C$

$= [c A_1 \ldots A_n]$ if $C \neq \mathcal{B}$

$= 1$ if $C = \mathcal{B}$ and $c A_1 \ldots A_n \in S$

$= 0$ if $C = \mathcal{B}$ and $c A_1 \ldots A_n \notin S$

Extension Lemma

Every finsat member S of a consistency class can be extended to a saturated and non-trivial set of normal first-order terms

Smullyan's MET follows from Extension Lemma and Hintikka's MET

Proof of Extension Lemma

A cc $\mathcal{Y}$ is compact if $\forall S: \; S \in \mathcal{Y} \Leftrightarrow \forall S' \subseteq_{fin} S : S' \in \mathcal{Y}$

Compact ccs are closed under chain limits

$\mathcal{Y}$ compact
$S_0 \subseteq S_1 \subseteq S_2 \subseteq \ldots$
$\forall i : S_i \in \mathcal{Y}$
$\Big\}$ $\bigcup_{i \in \mathbb{N}} S_i \in \mathcal{Y}$

For every cc $\mathcal{Y}$ exists compact cc $\mathcal{Y}''$ such that $\mathcal{Y} \subseteq \mathcal{Y}''$

$\mathcal{Y}' := \{S \mid \exists S' \in \mathcal{Y} : S \subseteq S'\}$ cc, $\mathcal{Y} \subseteq \mathcal{Y}'$

$\mathcal{Y}'' := \{S \mid \forall S' \subseteq_{fin} S : S' \in \mathcal{Y}'\}$ cc, $\mathcal{Y}' \subseteq \mathcal{Y}''$

$\mathcal{Y}$ compact $\wedge$ $S \in \mathcal{Y}$ finsat $\Rightarrow$ $\exists S' \in \mathcal{Y} : S \subseteq S' \wedge S'$ saturated

Construct chain $S = S_0 \subseteq S_1 \subseteq S_2 \subseteq \ldots \subseteq \bigcup_{i \in \mathbb{N}} S_i = S'$ as follows:

1) Choose enumeration $A_1, A_2, A_3, \ldots$ of all fofs

2) $S_{n+1} = \begin{cases} S_n & \text{if } S_n \cup \{A_{n+1}\} \notin \mathcal{Y} \\ S_n \cup \{A_{n+1}\} & \text{otherwise if } A_{n+1} \neq \exists \ldots \\ S_n \cup \{\exists x A, A[x := a]\} & \text{otherwise if } A_{n+1} = \exists x . A \\ & \text{and } a \notin \mathcal{N}(S_n, A) \cup \mathcal{B} \end{cases}$