Equational Reasoning

Logic Reasoning where judgements are equations

$\vdash \quad \subseteq \quad \models$

deductive entailment — semantic entailment

computational

G. Smolka

10-2

June 13, 2005

Validity / Satisfaction

$\mathcal{D}, \sigma \models s \doteq t \;\overset{\text{def}}{\Longleftrightarrow}\; \mathcal{D}\sigma s = \mathcal{D}\sigma t$

σ satisfies $s \doteq t$ in $\mathcal{D}$

σ is a solution of $s \doteq t$ in $\mathcal{D}$

$\mathcal{D} \models e \;\overset{\text{def}}{\Longleftrightarrow}\; \forall \sigma.\ \mathcal{D}, \sigma \models e$

$\mathcal{D}$ satisfies e

e valid in $\mathcal{D}$

$\mathcal{D}, \sigma \models E \;\overset{\text{def}}{\Longleftrightarrow}\; \forall e \in E.\ \mathcal{D}, \sigma \models e$

$\mathcal{D} \models E \;\overset{\text{def}}{\Longleftrightarrow}\; \forall e \in E.\ \mathcal{D} \models e$

Semantic Entailment

$A \models e \;\overset{\text{def}}{\Longleftrightarrow}\; \forall \mathcal{D}.\ \mathcal{D} \models A \Rightarrow \mathcal{D} \models e$ — every interpretation satisfying A satisfies e

$E \models_0 e \;\overset{\text{def}}{\Longleftrightarrow}\; \forall \mathcal{D}\, \forall \sigma.\ \mathcal{D}, \sigma \models E \Rightarrow \mathcal{D}, \sigma \models e$ — every solution of E is a solution of e

$E \models_0^A e \;\overset{\text{def}}{\Longleftrightarrow}\; \forall \mathcal{D}\, \forall \sigma.\ \mathcal{D} \models A \wedge \mathcal{D}, \sigma \models E \Rightarrow \mathcal{D}, \sigma \models e$

In every interpretation satisfying A, every solution of E is a solution of e

$\models$ semantic entailment

$\models_0$ conservative semantic entailment

$\models_0^A$ conservative semantic entailment wrt A

$\models, \models_0$ are special cases of $\models_0^A$

$\models_0 = \models_0^{\emptyset}$

Basic Properties of $\models, \models_0, \models_0^A$

1) $\models, \models_0, \models_0^A$ are entailment relations on Eq

2) $\models_0 \subseteq \models, \models_0^A$

3) $A' \models A \Rightarrow \models_0^A \subseteq \models_0^{A'}$

4) $A \models e \Leftrightarrow\ \models_0^A e$

5) $E \models_0^A e \Rightarrow E \cup A \models e$

6) E closed: $E \models_0^A e \Leftrightarrow E \cup A \models e$

Deductive Properties of $\vdash_0 \subseteq \models, \models^A_0$

Refl	$\models_0 s = s$	Equivalence
Sym	$s = t \models_0 t = s$	
Trans	$s = s',\ s' = t \models_0 s = t$	
CL	$s = s' \models_0 st = s't$	Congruence
CR	$t = t' \models_0 st = st'$	
β	$\models_0 (\lambda x.s)t = s[x:=t]$	
η	$\models_0 \lambda x.sx = s$ if $x \notin FVs$	

Deductive Properties of $\models$

$\models \;\hat{=}\; \models_0$ plus universal quantification of free variables

ξ $\quad s = s' \models \lambda x.s = \lambda x.s'$

Subst $\quad s = s' \models \varphi_\theta s = \varphi_\theta s'$

Subst can be derived from ξ, β, CL, Sym, Trans

since $\models \varphi_\theta s = (\lambda x_1 \dots x_n.\, s)(\theta x_1) \cdots (\theta x_n)$ $\quad \beta$

if $FVs = \{x_1, \dots, x_n\}$

Stability under Substitution

$\mathrel{|\!\sim}$ entailment relation on Eq

$\mathrel{|\!\sim}$ stable under substitution (sus) if

$\forall E \subseteq Eq\ \forall e \in E\ \forall$ subst. $\theta.\quad E \mathrel{|\!\sim} e \Rightarrow \varphi_\theta E \mathrel{|\!\sim} \varphi_\theta e$

$\boxed{\vdash_0 \text{ and } \models^A_0 \text{ are sus}}$

$\boxed{\mathrel{|\!\sim} \text{ sus} \iff \forall E \subseteq Eq\ \forall \text{ subst. } \theta .\ \varphi_\theta(\mathcal{C}(\mathrel{|\!\sim})E) \subseteq \mathcal{C}(\mathrel{|\!\sim})(\varphi_\theta E)}$

$\boxed{\text{If } f, g \text{ closure operators on } Eq \text{, then: } f, g \text{ sus} \Rightarrow f \circ g \text{ sus}}$

Goal: Deductive Entailment Relations

$\vdash \;\leq\; \models \qquad \vdash_0 \;\leq\; \models_0 \qquad \vdash^A_0 \;\leq\; \models^A_0$

- Must be computational
- Should be as complete as possible wrt semantic counterpart
- Modular construction with simple closure operators

Symmetry Closure

$\overleftrightarrow{E} \stackrel{def}{=} E \cup \{(t,s) \mid (s,t) \in E\}$

$\boxed{\leftrightarrow \text{ is CCO, sound for } \models_0, \text{ sus}}$

Reflexive Transitive Closure

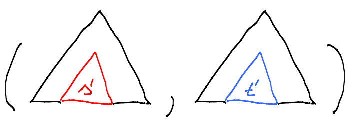
$E^* \stackrel{def}{\Leftrightarrow} \{(s,s) \mid s \in Ter\} \cup E^+$

$$\boxed{\begin{array}{l}(s,t) \in E^* \Leftrightarrow \exists s_1,\ldots,s_n \in Ter.\ s_1 = s \wedge s_n = t \\ \qquad \wedge\ \forall i \in \{1,\ldots,n-1\}.\ (s_i, s_{i+1}) \in E\end{array}}$$

$\boxed{* \text{ is CCO, sound for } \models_0, \text{ sus}}$

Replacement Closure

Rep $E \stackrel{def}{=} \{(s,t) \mid \exists (s',t') \in E$ and that t obtainable from s by replacing exactly one occurrence of s' with $t'\}$



where $(s',t') \in E$

$\boxed{\text{Rep is CCO, sound for } \models}$

CCO : Computational Closure Operator on Eq

Conservative Replacement Closure

Rep_0 $E \stackrel{def}{=} \{(s,t) \mid \exists (s',t') \in E$ and that t obtainable from s by replacing exactly one occurrence of s' with t' and that no free variable of (s',t') is captured$\}$

$\boxed{\text{Rep}_0 \text{ is CCO, sound for } \models_0, \text{ sus}}$

Proof of soundness:

$$\begin{aligned} s &= (\lambda x.s'')s' && \beta \\ &= (\lambda x.s'')t' && CR, s' = t' \\ &= t && \beta \quad \square \end{aligned}$$

$\boxed{\text{Rep}_0 \leq \text{Rep}}$ $\boxed{E \text{ closed}: \text{Rep}_0 E = \text{Rep}\, E}$

Substitution Closure

$\mathrm{Sub}\,E \overset{\text{def}}{=} \{(\theta s, \theta t) \mid (s,t) \in E,\ \theta \text{ substitution}\}$

$\boxed{\mathrm{Sub} \text{ is CCO, sound for } \models}$

E-Conversion

$\mathrm{Con}_0\,E \overset{\text{def}}{=} \mathrm{Rep}_0\,\overleftrightarrow{E}$ conservative E-conversion

$\mathrm{Con}\,E \overset{\text{def}}{=} \mathrm{Rep}(\mathrm{Sub}\,\overleftrightarrow{E})$ E-conversion

$\boxed{\mathrm{Con}_0 \text{ is CCO, sound for } \models_0, \text{ SUS}}$

$\boxed{\mathrm{Con} \text{ is CCO, sound for } \models}$

$\boxed{\mathrm{Con}_0 \leq \mathrm{Con}}$ $\boxed{E \text{ closed: } \mathrm{Con}_0 E = \mathrm{Con}\,E}$

λ-Conversion

$\mathrm{Lam} \overset{\text{def}}{=} \{((\lambda x.s)t,\ s[x:=t]) \mid (\lambda x.s)t \in \mathrm{Ter}\}$ β

$\cup\ \{(\lambda x.sx,\ s) \mid \lambda x.sx \in \mathrm{Ter},\ x \notin \mathrm{FV}\,s\}$ η

$\boxed{\models \mathrm{Lam}}$

$\boxed{\mathrm{Lam} = \mathrm{Sub}\,\mathrm{Lam}}$

$\boxed{\mathrm{Con}\,\mathrm{Lam} = \mathrm{Rep}(\overleftrightarrow{\mathrm{Lam}})}$

Deductive Entailment Relations

$A \vdash e \overset{\text{def}}{\iff} e \in (\mathrm{Con}\,\mathrm{Lam} \cup \mathrm{Con}\,A)^*$ $\quad\boxed{\vdash\ \subseteq\ \models}$

$E \vdash_0 e \overset{\text{def}}{\iff} e \in (\mathrm{Con}\,\mathrm{Lam} \cup \mathrm{Con}_0 E)^*$ $\quad\boxed{\vdash_0\ \subseteq\ \models_0}$

$E \overset{A}{\vdash}_0 e \overset{\text{def}}{\iff} e \in (\mathrm{Con}\,\mathrm{Lam} \cup \mathrm{Con}_0 E \cup \mathrm{Con}\,A)^*$ $\quad\boxed{\overset{A}{\vdash}_0\ \subseteq\ \overset{A}{\models}_0}$

$\boxed{\vdash,\ \vdash_0,\ \overset{A}{\vdash}_0 \text{ are computational entailment relations}}$ idempotence!

$\boxed{\vdash,\ \vdash_0 \text{ are special cases of } \overset{A}{\vdash}_0}$

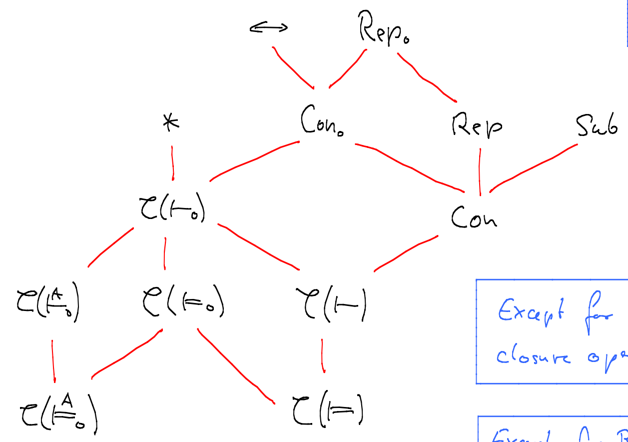
Properties of $\vdash$, $\vdash_0$, $\vdash^A_0$

You have seen these properties before for $\models$, $\models_0$, $\models^A_0$

1) $\vdash_0 \subseteq \vdash, \vdash^A_0$
2) $A' \vdash A \Rightarrow \vdash^A_0 \subseteq \vdash^{A'}_0$
3) $A \vdash e \Leftrightarrow \vdash^A_0 e$
4) $E \vdash^A_0 e \Rightarrow E \cup A \vdash e$
5) E closed: $E \vdash^A_0 e \Leftrightarrow E \cup A \vdash e$
6) $\vdash_0$ and $\vdash^A_0$ are sus

Soundness Hierarchy of Closure Operators

f above $g \Rightarrow f$ sound for g



Except for $\models^A_0$, $\models_0$, $\models$ all shown closure operators are computational

Except for Rep, Sub, Con, $\vdash$, $\models$ all shown closure operators are sus

Closedness Hierarchy

$\leftrightarrow$
|
Subst
|
Rep_0
|
Rep
|
$*$

f above $g \Rightarrow$

$\forall E.\ fE \subseteq E \Rightarrow f(gE) \subseteq gE$

First-order / Higher-order Distinction

- A value constant is called *first-order* if it has a type $C_1 \to \dots \to C_n$ where $C_1, \dots, C_n$ are type constants.
- A variable is called *first-order* if its type is a type constant.
- A term is called *first-order* if it is combinatoric and contains only first-order constants and variables.
- A set of equations is called *first-order* if it contains only first-order terms.
- Something is *higher-order* if it is not first-order.

Completeness Results

No value constants. Then:
$\forall e \in Eq.\quad \models e \Leftrightarrow \vdash e$

Harvey Friedman 1975

No higher-order value constants, A first-order. Then:
$\forall e \in Eq: \quad A \models e \Leftrightarrow A \vdash e$

See M. Kaminski's Bachelor Thesis 2005

Undecidability Results

$\exists A$ finite and first-order:
$\{e \mid A \vdash e,\ e \text{ first-order}\}$ undecidable.

Turing 1936
universal TM

Given the constants $0, 1, +, \cdot, \leq, \wedge, \neg, \forall$:
$\{e \mid \mathbb{N} \models e\}$ not semi-decidable.

Gödel 1937

$\exists$ finite $A \subseteq Eq$: $\{e \mid A \models e\}$ not semi-decidable

λ-Reduction

Church, Rosser, Turing

$Red_\lambda \overset{\text{def}}{=} Rep\,(Lam)$

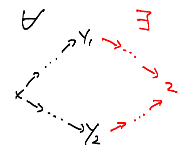
Red_λ is terminating and confluent

$\forall (s,t) \in Eq.\quad \vdash s = t \Leftrightarrow s, t$ have the same Red_λ-NF

Termination and Confluence

R binary relation

- R terminating if $\neg\exists$ infinite path
- R confluent if $\forall x, y_1, y_2$.
  $\left.\begin{array}{l}\exists \text{ path from } x \text{ to } y_1 \\ \exists \text{ path from } x \text{ to } y_2\end{array}\right\} \exists z.\ \begin{array}{l}\exists \text{ path from } y_1 \text{ to } z \\ \wedge\ \exists \text{ path from } y_2 \text{ to } z\end{array}$



- y R-normal form of x if y terminal and reachable from x

R terminating $\Rightarrow$ every x has R-NF

R confluent $\Rightarrow$ no x has more than one R-NF

Minimal Equational Deduction System $\mathcal{M}$

CR $\dfrac{t = t'}{s\,t = s\,t'}$ ξ $\dfrac{s = s'}{\lambda x.s = \lambda x.s'}$

β $\dfrac{}{(\lambda x.s)t = s[x:=t]}$ η $\dfrac{}{\lambda x.sx = s}\ x \notin FVs$

ST $\dfrac{s = t \quad s = t'}{t = t'}$

$\boxed{\forall A\,e.\quad A \vdash e \iff A \vdash^{\mathcal{M}} e}$

More Notation

$E \vdash^{A} e \overset{\text{def}}{\iff} E \cup A \vdash e$

$E \models^{A} e \overset{\text{def}}{\iff} E \cup A \models e$

$\boxed{\vdash^{A}\ \text{cco, sound for}\ \models^{A}}$

$\boxed{A' \vdash A \implies \vdash^{A} \subseteq \vdash^{A'}}$

$\boxed{\models^{A}\ \text{closure operator}}$

$\boxed{A' \models A \implies \models^{A} \subseteq \models^{A'}}$