Unconstrained I/O Logic

by Xavier Parent

October 30, 2013

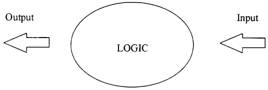
Introduction

What is input/output logic?

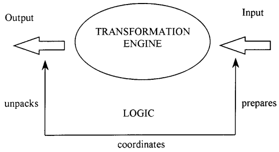
- A general framework for reasoning about norms
- Needn't assume norms bear a truth-value
- On top of a base logic
- Comes in two levels
  - Unconstrained: obligation, permission, constitutive norm
  - Constrained: CTDs, priorities, permission as exception
- Both a semantics and a proof theory

Logic in the I/O business

I/O logic: a "way of using logic"



(a) Inference motor



(b) Secretarial assistant

Role of logic

- Any base logic may be used as a secretarial assistant
- In the literature: classical logic

Language

- A conditional obligation is just a pair (a, x), called a generator
- a and x are propositional formulae
- (a, x) is a rule; a is called the body and x is called the head
- A normative system N is a set of such pairs

Exercise

- Give the normative system consisting of two norms stating that the community has to give a house with low rent (house) to low income agents (poor) and to provide free health insurance (healthins) to elderly agents (old).

Exercise

- Give the normative system consisting of two norms stating that the community has to give a house with low rent (house) to low income agents (poor) and to provide free health insurance (healthins) to elderly agents (old).
- $N = \{(poor, house), (old, healthins)\}$

Semantics based on detachment

The semantics is ‘operational’

$$\boxed{x \in out(N, A)}$$

- Calculates whether according to normative system N and in context A, a formula x is obligatory
- A is the input (a set of wffs); x (a wff) is the output
- Detachment as a core mechanism

Modus-ponens

(i)	If a, then x
(ii)	a
(iii)	So, x

Boghossian:
this is constitutive of the notion of conditional

Detachment
- the only assumption made in IOL
- can hardly be challenged

DSDL
- extra assumptions
- potentially prone to criticisms
  - maximizing
  - trichotomy of value relations

Detachment

(1) Factual detachment

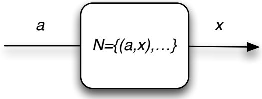
(i) If a is the case, then x is obligatory
(ii) a is the case
(iii) So, x is obligatory

In the I/O notation:

$$\text{If } (a, x) \in N \text{ then } x \in out(N, a)$$

N: a set of pairs of propositional wffs (conditional norms)
$out(N, a)$: output of a under N.

Factual detachment



Iteration of successive detachments

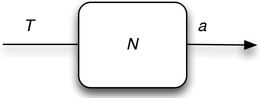
(2) Deontic detachment

- (i) If a is obligatory, then x is obligatory
- (ii) a is obligatory
- (iii) So, x is obligatory

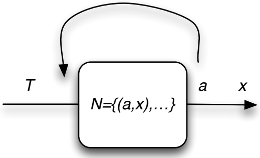
In the I/O notation:

$$\text{If } a \in out(N, \top) \text{ and } (a, x) \in N \text{ then } x \in out(N, \top)$$

Iteration of successive detachments



(1) a detached



(2) x detached

Exercise

- $N = \{(\mathrm{poor}, \mathrm{house}), (\mathrm{old}, \mathrm{healthins})\}$
- Represent that the community has to provide a house to someone with no income if no-income implies poor

Exercise

- $N = \{(\mathrm{poor}, \mathrm{house}), (\mathrm{old}, \mathrm{healthins})\}$
- Represent that the community has to provide a house to someone with no income if no-income implies poor
- $\mathrm{house} \in out(N, \neg\mathrm{income} \wedge (\neg\mathrm{income} \to \mathrm{poor}))$

Building blocks Nr. 1

Consequence operation

$Cn(A)$: set of logical consequences of A in classical logic.
$Cn(A) = \{x : A \vdash x\}$ (for $\vdash$, read 'proves').

Notation: Curly brackets $\{\}$ omitted when convenient.
$Cn(\emptyset)$: set of all tautologies.

Tarskian properties

Inclusion: $A \subseteq Cn(A)$
Monotony: $A \subseteq B$ implies $Cn(A) \subseteq Cn(B)$
Idempotence: $Cn(A) = CnCn(A)$

Compactness

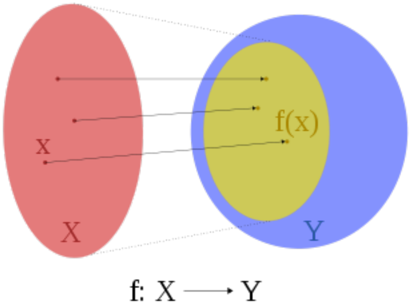
If $x \in Cn(A)$, then there is a finite $A' \subseteq A$ such that $x \in Cn(A')$.

Building block Nr.2

Detachment

Image

In mathematics, an image is the subset of a function's codomain which is the output of the function on a subset of its domain.



Building block Nr.2

Detachment

$N = \{(a, x), (b, y)...\}$: a set of pairs
$X = \{a, b, ...\}$: a set of formulae (input set)

Image of X under N

The image of X under N is the output of N on X:

$$G(X) =_{df} \{x : (a, x) \in N \text{ for some } a \in X\}$$

For $G(X)$, read 'the G of X'

Building block Nr.2

Detachment

$N = \{(a, x), (b, y)...\}$: a set of pairs
$X = \{a, b, ...\}$: a set of formulae (input set)

Image of X under N

The image of X under N is the output of N on X:

$$G(X) =_{df} \{x : (a, x) \in N \text{ for some } a \in X\}$$

For $G(X)$, read 'the G of X'

N	X	$G(X)$
$\{(a_1, x_1), (a_2, x_2)\}$	$\{a_1\}$	?
$\{(a_1, x_1), (a_2, x_2)\}$	$\{a_1, x_2\}$	?
$\{(a_1, x_1), (a_2, x_2)\}$	$\{a_1, a_2\}$	?
$\{(a_1, x_1), (a_2, x_2)\}$	$\emptyset$	?

Building block Nr.2

Detachment

$N = \{(a, x), (b, y)...\}$: a set of pairs
$X = \{a, b, ...\}$: a set of formulae (input set)

Image of X under N

The image of X under N is the output of N on X:

$$G(X) =_{df} \{x : (a, x) \in N \text{ for some } a \in X\}$$

For $G(X)$, read 'the G of X'

Useful property

If $X \subseteq Y$, then $G(X) \subseteq G(Y)$

Building block Nr.2

Detachment

$N = \{(a, x), (b, y)...\}$: a set of pairs
$X = \{a, b, ...\}$: a set of formulae (input set)

Image of X under N

The image of X under N is the output of N on X:

$$G(X) =_{df} \{x : (a, x) \in N \text{ for some } a \in X\}$$

For $G(X)$, read 'the G of X'

Useful property

If $X \subseteq Y$, then $G(X) \subseteq G(Y)$

This is called monotony w.r.t input.
The bigger the input, the more can be detached.

Simple-minded output operation out_1

Simple-minded output, out_1

$$out_1(N, A) =_{df} Cn(G(Cn(A)))$$

Simple-minded output operation out_1

Simple-minded output, out_1

$$out_1(N, A) =_{df} Cn(G(Cn(A))$$

?-$out_1(N, A)$

- ?-$Cn(A)$
- ?-$G(Cn(A))$
- ?-$Cn(G(Cn(A)))$

Simple-minded output operation out_1

Simple-minded output, out_1

$$out_1(N, A) =_{df} Cn(G(Cn(A))$$

?-$out_1(N, A)$

- ?-$Cn(A)$
- ?-$G(Cn(A))$
- ?-$Cn(G(Cn(A)))$

Examples

$N = \{(a, x)\}$. $out_1(N, a) =?$
$N = \{(a \vee b, x)\}$. $out_1(N, a) =?$
$N = \{(a, x), (a, y)\}$. $out_1(N, a) =?$

Simple-minded output operation out_1

Simple-minded output, out_1

$$out_1(N, A) =_{df} Cn(G(Cn(A))$$

?-$out_1(N, A)$

- ?-$Cn(A)$
- ?-$G(Cn(A))$
- ?-$Cn(G(Cn(A)))$

Examples

$N = \{(a, x)\}$. $out_1(N, a) =?$
$N = \{(a \vee b, x)\}$. $out_1(N, a) =?$
$N = \{(a, x), (a, y)\}$. $out_1(N, a) =?$

A	$Cn(A)$	$G(Cn(A))$	$Cn(G(Cn(A))$
a	a	x	$Cn(x)$
a	$a \vee b$	x	$Cn(x)$

Remarks

- 1-argument notation:

$$out_1(N) = \{(a, x) \mid x \in out_1(N, a)\}$$

- Unpacking out_1: $x \in Cn(G(Cn(A))$ means

$$\begin{array}{l} A \vdash a_1 \wedge ... \wedge a_n \\ \quad (a_1, x_1), ..., (a_n, x_n) \text{ in } N \\ \qquad \text{and } x_1 \wedge ... \wedge x_n \vdash x \end{array}$$

Exercise

- $N = \{(\text{poor}, \text{house}), (\text{old}, \text{healthins})\}$
- What are the obligations of the community for low income ederly agents?

Exercise

- $N = \{(\text{poor}, \text{house}), (\text{old}, \text{healthins})\}$
- What are the obligations of the community for low income ederly agents?
- All logical consequences of giving a house with low rent and providing a free health insurance
- $out_1(N, \text{poor} \wedge \text{old}) = Cn(\text{house}, \text{healthins})$

Exercise (2)

Question

What are:

- $out_1(\{(a, x), (b, y)\}, \top)$,
- $out_1(\{(a, x), (b, y)\}, a \vee b)$,
- $out_1(\{(a, x), (b, y)\}, a \wedge b)$,
- $out_1(\{(a, x), (b, y)\}, a \wedge b \wedge c)$,
- $out_1(\{(a, x), (x, y)\}, a)$, *and*
- $out_1(\{(a, x), (a \wedge b, \neg x)\}, a \wedge b)$?

Exercise (2)

Question

What are:

- $out_1(\{(a,x),(b,y)\},\top)$,
- $out_1(\{(a,x),(b,y)\},a \vee b)$,
- $out_1(\{(a,x),(b,y)\},a \wedge b)$,
- $out_1(\{(a,x),(b,y)\},a \wedge b \wedge c)$,
- $out_1(\{(a,x),(x,y)\},a)$, *and*
- $out_1(\{(a,x),(a \wedge b,\neg x)\},a \wedge b)$*?*

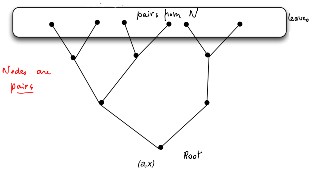
They are respectively:

- $Cn(\emptyset)$,
- $Cn(\emptyset)$,
- $Cn(x,y)$,
- $Cn(x,y)$,
- $Cn(x)$,
- $Cn(x,\neg x)$.

Proof-theory

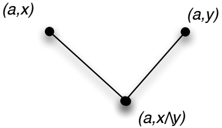
Trick: $(a, x) \in out(N)$ in place of $x \in out(N, a)$.

Derivation



A derivation of (a, x) from N using rule set R is a tree

- nodes are pairs of the form (b, y)
  - root node labelled with (a, x) (conclusion, at the bottom)
  - leaf nodes labelled with elements in N (assumptions, at the top)
- a fork shows an application of a rule from R



$$\frac{(a,x) \qquad (a,y)}{(a,x \wedge y)}\ AND$$

For ease of exposition

- do not draw the bullet points and the edges
- draw a horizontal line labelled with the name of the rule
- allow for e.g. $a \vdash b$ to appear as nodes.

Rules for out_1

$$(\mathrm{SI})\quad \frac{(a,x) \qquad b \vdash a}{(b,x)} \qquad\qquad (\mathrm{WO})\quad \frac{(a,x) \qquad x \vdash y}{(a,y)}$$

$$(\mathrm{AND})\quad \frac{(a,x) \qquad (a,y)}{(a, x \wedge y)}$$

Simplest IOL system

Output operation	Rules
Simple-minded (out_1)	{SI, WO, AND}

Completeness theorem

(a,x) derivable from N using {SI, WO, AND} $\Leftrightarrow (a,x) \in out_1(N)$

(equivalently, $x \in out_1(N,a)$)

Soundness

Soundness theorem

(a, x) derivable from N using $\{$SI, WO, AND$\} \Rightarrow x \in out_1(N, a)$

Proof

A spin-off of: out_1 validates the rules (SI), (WO) and (AND)

Soundness

Soundness theorem

(a, x) derivable from N using $\{\text{SI, WO, AND}\} \Rightarrow x \in out_1(N, a)$

Proof

A spin-off of: out_1 validates the rules (SI), (WO) and (AND)

Soundness

Soundness theorem

(a, x) derivable from N using $\{$SI, WO, AND$\} \Rightarrow x \in out_1(N, a)$

Proof

A spin-off of: out_1 validates the rules (SI), (WO) and (AND)

WARNING

To show that an I/O operation, here out_1, validates a given rule, it is a mistake to start by assuming that a pair appearing as a premisse is in N.

For (SI), the correct hypothesis is: $x \in out_1(N, a)$.
$(a, x) \in N$ isn't correct.
To show: $x \in out_1(N, b)$.

$$(\mathrm{SI}) \quad \frac{(a, x) \qquad\qquad b \vdash a}{(b, x)}$$

Soundness

Soundness theorem

(a, x) derivable from N using $\{\text{SI, WO, AND}\} \Rightarrow x \in out_1(N, a)$

Proof

A spin-off of: out_1 validates the rules (SI), (WO) and (AND)

For (SI). Assume $x \in out_1(N, a)$ and $b \vdash a$.
From $b \vdash a$, we get

$$
\begin{aligned}
& a \in Cn(b) \\
& \{a\} \subseteq Cn(b) \\
& Cn(a) \subseteq CnCn(b) \\
& Cn(a) \subseteq Cn(b) \\
& G(Cn(a)) \subseteq G(Cn(b)) \\
& \underbrace{Cn(G(Cn(a)))}_{out_1(N,a)} \subseteq \underbrace{Cn(G(Cn(b)))}_{out_1(N,b)}
\end{aligned}
$$

So, $x \in out_1(N, b)$

Soundness

Soundness theorem

(a, x) derivable from N using $\{\text{SI, WO, AND}\} \Rightarrow x \in out_1(N, a)$

Proof

A spin-off of: out_1 validates the rules (SI), (WO) and (AND)

Alternative argument for SI:

$x \in out_1(N, a)$ means

$$a \vdash a_1 \wedge \ldots \wedge a_n,$$
$$(a_1, x_1), \ldots, (a_n, x_n) \text{ in } N \text{ and } x_1 \wedge \ldots \wedge x_n \vdash x$$

If $b \vdash a$ then

$$b \vdash a_1 \wedge \ldots \wedge a_n,$$
$$(a_1, x_1), \ldots, (a_n, x_n) \text{ in } N \text{ and } x_1 \wedge \ldots \wedge x_n \vdash x$$

So $x \in out_1(N, b)$

Completeness

Completeness theorem

$x \in out_1(N, a) \Rightarrow (a, x)$ derivable from N using {SI, WO, AND}

Proof

Assume $x \in out_1(N, a)$. So

$a \vdash a_1 \wedge ... \wedge a_n$,

$(a_1, x_1), ..., (a_n, x_n)$ in N and $x_1 \wedge ... \wedge x_n \vdash x$

Below: a derivation of (a, x) from N.

$$
\dfrac{\dfrac{\dfrac{(a_1, x_1)}{(a, x_1)}\,\text{SI} \qquad \cdots\cdots \qquad \dfrac{(a_n, x_n)}{(a, x_n)}\,\text{SI}}{(a, x_1 \wedge \ldots \wedge x_n)}\,\text{AND}}{(a, x)}\,\text{WO}
$$

Other I/O operations

- out_2: basic output
- out_3: reusable simple-minded
- out_4: reusable basic

$$\text{(SI)}\quad \frac{(a,x)\qquad b\vdash a}{(b,x)} \qquad\qquad \text{(WO)}\quad \frac{(a,x)\qquad x\vdash y}{(a,y)}$$

$$\text{(AND)}\quad \frac{(a,x)\qquad (a,y)}{(a,x\wedge y)} \qquad\qquad \text{(OR)}\quad \frac{(a,x)\qquad (b,x)}{(a\vee b,x)}$$

$$\text{(CT)}\quad \frac{(a,x)\qquad (a\wedge x,y)}{(a,y)}$$

IOL systems

Output operation	Rules
Simple-minded (out_1)	{SI, WO, AND}
Basic (out_2)	{SI, WO, AND}+{OR}
Reusable simple-minded (out_3)	{SI, WO, AND}+{CT}
Reusable basic (out_4)	{SI, WO, AND}+{OR,CT}

Basic output, out_2

Building block Nr 3

Call a set V of wffs complete iff: $V = L$ (the set of all wffs) or V is maximal consistent.

Maximal consistency

MCS: short for "maximal consistent set" (of wffs).
V is a MCS if

- V is consistent, $V \nvdash \bot$,and
- none of its proper extension is consistent: $\forall V' \supset V, V' \vdash \bot$

Properties

Given a MCS V,

- for all b, either $b \in B$ or $\neg b \in V$ ($\neg$-completeness)
- $a \wedge b \in V$ iff $a, b \in V$ (conj. property)
- $a \vee b \in V$ iff $a \in V$ or $b \in V$ (disj. property)
- $a \rightarrow b \in V$ iff $b \in V$ whenever $a \in V$

Apply, *mutatis mutandis*, to complete sets.

Basic output, out_2

Basic output

$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \text{ complete}\}$

Basic output, out_2

Basic output

$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \text{ complete}\}$

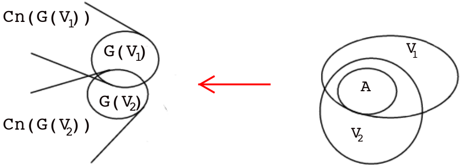
$\cap$ (meet) $=$ $\forall$ (for all)

$x \in out_2(N, A)$, iff for all complete $V \supseteq A$, $x \in Cn(G(V))$

Basic output, out_2

Basic output

$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \text{ complete}\}$



V	$G(V)$	$Cn(G(V))$
V_1	$G(V_1)$	$Cn(G(V_1))$
V_2	$G(V_2)$	$Cn(G(V_2))$
$\vdots$	$\vdots$	$\vdots$
V_i	$G(V_i)$	$Cn(G(V_i))$

One of these V_i is L.
The others are maximal consistent extension (MCE) of A

Example

$$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \text{ complete}\}$$

Reasoning by cases.

Question

What are:

- $out_1(\{(a, x), (b, x)\}, a \vee b)$, *and*
- $out_2(\{(a, x), (b, x)\}, a \wedge b)$*?*

Answer

$out_1(N, a \vee b) = Cn(\emptyset)$.
But $out_2(N, a \vee b) = Cn(x)$.

V	$G(V)$	$Cn(G(V))$
L	{x}	$Cn(x)$
MCE of $\{a \vee b\}$	{x}	$Cn(x)$

Disj. property: $a \vee b$ in a MCS iff one of the disjuncts is in it.

Example

$$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \text{ complete}\}$$

Question

What is $out_2(\{(a \wedge b, x), (a \wedge \neg b, x)\}, a\}$*?*

Example

$$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \ complete\}$$

Question

What is $out_2(\{(a \wedge b, x), (a \wedge \neg b, x)\}, a\}$*?*

Answer

$out_2(\{(a \wedge b, x), (a \wedge \neg b, x)\}, a\} = Cn(x)$

V	$G(V)$	$Cn(G(V))$
L	$\{\mathrm{x}\}$	$Cn(x)$
MCE of $\{a, b\}$	$\{\mathrm{x}\}$	$Cn(x)$
MCE of $\{a, \neg b\}$	$\{\mathrm{x}\}$	$Cn(x)$

$\neg$-completeness: for all b, either b or $\neg b$ is in a MCS

out_2

$$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \textit{ complete}\}$$

Completeness theorem

(a, x) derivable from N using {SI, WO, AND,OR} $\Leftrightarrow (a, x) \in out_2(N)$

(equivalently, $x \in out_2(N, a)$)

out_2

$$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \textit{ complete}\}$$

Soundness

out_2 validates {SI, WO, AND, OR}

out_2

$$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \text{ } complete\}$$

Soundness

out_2 validates {SI, WO, AND, OR}

OR example:

Assume

(1) $x \in out_2(N, a)$

(2) $x \in out_2(N, b)$

To show:

(3) $x \in out_2(N, a \vee b)$

out_2

$$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \text{ complete}\}$$

Soundness

out_2 validates {SI, WO, AND, OR}

OR example:

Assume

(1) $x \in out_2(N, a)$

(2) $x \in out_2(N, b)$

To show:

(3) $x \in out_2(N, a \vee b)$

Let V be a complete set such that $a \vee b \in V$. To show: $x \in Cn(G(V))$.

out_2

$$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \text{ complete}\}$$

Soundness

out_2 validates {SI, WO, AND, OR}

OR example:

Assume

(1) $x \in out_2(N, a)$

(2) $x \in out_2(N, b)$

To show:

(3) $x \in out_2(N, a \vee b)$

Let V be a complete set such that $a \vee b \in V$. To show: $x \in Cn(G(V))$.
Either $a \in V$ or $b \in V$ (disj. property)

out_2

$$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \ complete\}$$

Soundness

out_2 validates {SI, WO, AND, OR}

OR example:

Assume

(1) $x \in out_2(N, a)$

(2) $x \in out_2(N, b)$

To show:

(3) $x \in out_2(N, a \vee b)$

Let V be a complete set such that $a \vee b \in V$. To show: $x \in Cn(G(V))$.
Either $a \in V$ or $b \in V$ (disj. property)

Case 1: $a \in V$

Case 2: $b \in V$

out_2

$$out_2(N, A) = \cap\{Cn(G(V)) : A \subseteq V, V \textit{ complete}\}$$

Soundness

out_2 validates {SI, WO, AND, OR}

OR example:

Assume

(1) $x \in out_2(N, a)$

(2) $x \in out_2(N, b)$

To show:

(3) $x \in out_2(N, a \vee b)$

Let V be a complete set such that $a \vee b \in V$. To show: $x \in Cn(G(V))$.

Either $a \in V$ or $b \in V$ (disj. property)

Case 1: $a \in V$

By (1), $x \in Cn(G(V))$

Case 2: $b \in V$

By (2), $x \in Cn(G(V))$

Either way, $x \in Cn(G(V))$ as required.

Reusable simple-minded output, out_3

Reusable simple-minded output

$out_3(N, A) = \cap\{Cn(G(B)) : A \subseteq B = Cn(B) \supseteq G(B)\}$

Reusable simple-minded output, out_3

Reusable simple-minded output
$out_3(N, A) = \cap\{Cn(G(B)) : A \subseteq B = Cn(B) \supseteq G(B)\}$

- $B \subseteq Cn(B)$ is just inclusion (slide 12)
- $B \supseteq Cn(B)$ glossed as "B is closed under Cn"
  - If $x \in Cn(B)$, then $x \in B$
- $B \supseteq G(B)$ glossed as "B is closed under G"

Reusable simple-minded output, out_3

Reusable simple-minded output

$out_3(N, A) = \cap\{Cn(G(B)) : A \subseteq B = Cn(B) \supseteq G(B)\}$

Alternative definition:

$$out_3(N, A) = Cn(G(B^\star))$$

where $B^\star$ is the smallest superset of A that is closed under Cn and G.

$B^* \supseteq G(B^*)$

SE

for input t,

N is	(t, s)	$(t \wedge s, u)$	(t, s) $(t \wedge s, u)$
$B^\star$ is	$Cn(t, s)$	$Cn(t)$	$Cn(t, s, u)$
out_3 is	$Cn(s)$	$Cn(\emptyset)$	$Cn(s, u)$

Reusable simple-minded output, out_3

Reusable simple-minded output

$out_3(N, A) = \cap\{Cn(G(B)) : A \subseteq B = Cn(B) \supseteq G(B)\}$

Question

What are:

- $out_3(\{(a, x), (b, y)\}, a \vee b)$,
- $out_3(\{(a, x), (b, y)\}, a \wedge b)$,
- $out_3(\{(a, x), (x, y)\}, a)$, *and*
- $out_3(\{(a, x), (x, y), (y, z)\}, a)$*?*

Reusable simple-minded output, out_3

Reusable simple-minded output

$out_3(N, A) = \cap\{Cn(G(B)) : A \subseteq B = Cn(B) \supseteq G(B)\}$

Question

What are:

- $out_3(\{(a, x), (b, y)\}, a \vee b)$,
- $out_3(\{(a, x), (b, y)\}, a \wedge b)$,
- $out_3(\{(a, x), (x, y)\}, a)$, *and*
- $out_3(\{(a, x), (x, y), (y, z)\}, a)$?

$Cn(\emptyset)$, $Cn(x, y)$, $Cn(x, y)$, $Cn(x, y, z)$.

Reusable simple-minded output, out_3

Reusable simple-minded output

$out_3(N, A) = \cap\{Cn(G(B)) : A \subseteq B = Cn(B) \supseteq G(B)\}$

Reusable means: output may be recycled as input

Distinctive rule of out_3:

$$(\mathrm{CT}) \quad \frac{(a, x) \qquad (a \wedge x, y)}{(a, y)} \qquad \frac{x \in out_3(N, a) \qquad y \in out_3(N, a \wedge x)}{y \in out_3(N, a)}$$

Reusable simple-minded output, out_3

Reusable simple-minded output

$out_3(N, A) = \cap\{Cn(G(B)) : A \subseteq B = Cn(B) \supseteq G(B)\}$

Assume

(1) $x \in out_3(N, a)$

(2) $y \in out_3(N, a \wedge x)$

To show:

(3) $y \in out_3(N, a)$

Reusable simple-minded output, out_3

Reusable simple-minded output

$out_3(N, A) = \cap\{Cn(G(B)) : A \subseteq B = Cn(B) \supseteq G(B)\}$

Assume
(1) $x \in out_3(N, a)$
(2) $y \in out_3(N, a \wedge x)$

To show:
(3) $y \in out_3(N, a)$

Let B be that $a \in B = Cn(B) \supseteq G(B)$. To show: $y \in Cn(G(B))$.
By (1), $x \in Cn(G(B))$.
So $x \in Cn(B)$ (cos $B \supseteq G(B)$, and Cn is monotonic, slide 12)
So $a \wedge x \in B$ (cos $B = Cn(B)$)
By (2), $y \in Cn(G(B))$. CQFD.

Reusable simple-minded output, out_3

Reusable simple-minded output

$out_3(N, A) = \cap\{Cn(G(B)) : A \subseteq B = Cn(B) \supseteq G(B)\}$

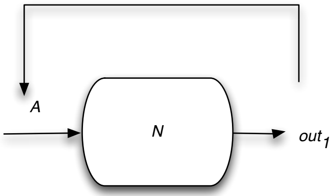
Bulk increment

$out_3^b(N, A) = \cup_{i=0}^{\omega} A_i$ where

- $A_0 = out_1(N, A)$
- $A_{n+1} = Cn(A_n \cup out_1(N, A_n \cup A))$

Result

$out_3^b(N, A) = out_3(N, A)$



out_1 made iterative

Reusable basic output, out_4

Reusable basic

$out_4(N, A) = \cap\{Cn(G(V)) : A \subseteq V \supseteq G(V), V \text{ complete }\}$

Rules added to out_1 : OR and CT.