Permission & Constrained I/O Logic

by Xavier Parent

October 30, 2013

Permission

- It is permitted to drive at a speed of 95 km/h on a motorway
- Anyone over 18 can buy booze legally

Permission

- It is permitted to drive at a speed of 95 km/h on a motorway
- Anyone over 18 can buy booze legally

Negative permission

Let N be a set of generators, and let *out* be an input/output logic.

$$(a, x) \in negperm(N) \text{ iff } (a, \neg x) \notin out(N)$$

Positive permission - static

Let N and P be two sets of generators, where P stands for permissive norms, and let *out* be an input/output logic.

$$(a, x) \in statperm(P, N) \text{ iff } (a, x) \in out(N \cup Q) \quad \text{for some singleton or empty } Q \subseteq P$$

(a, x) is generated either by the obligations in N alone, or by the norms in N together with

Permission

Positive permission - dynamic

$$(a, x) \in \mathit{dynperm}(P, N) \text{ iff } (c, \neg z) \in \mathit{out}(N \cup \{(a, \neg x)\}) \\ \text{for some } (c, z) \in \mathit{statperm}(P, N) \text{ with} \\ c \text{ consistent}$$

Forbidding x under condition a would prevent the agent from making use of some explicit (static) permission (c, z).
(a, x) protected by the code.

Permission

Positive permission - dynamic

Danish's caricatures of the prophet Mohammed

freedom of speech

banning

$$(a, x) \in dynperm(P, N) \text{ iff } (c, \neg z) \in out(N \cup \{(a, \neg x)\})$$

$$\text{for some } (c, z) \in statperm(P, N) \text{ with}$$

$$c \text{ consistent}$$

publishing caricatures

freedom speech

Forbidding x under condition a would prevent the agent from making use of some explicit (static) permission (c, z).
(a, x) protected by the code.

Proof theory

Subverse of a rule: obtained by downgrading to permission status one of the premises, and also the conclusion of the rule.
Ex: subverse of AND is

$$\frac{(a,x)^o \quad (a,y)^p}{(a,x \wedge y)^p}$$

Theorem

If *out* satisfies a rule R, then the corresponding *statperm* satisfies its subverse.

Constrained I/O Logic (cIOL)

Problem with uIOL: excess output

- conflict

  $N_1 = \{(a, b), (a, \neg b)\}$ input: a - output: $Cn(b, \neg b) = L$ Explosion!!

Constrained I/O Logic (cIOL)

Problem with uIOL: excess output

- conflict

  $N_1 = \{(a, b), (a, \neg b)\}$ input: a - output: $Cn(b, \neg b) = L$ Explosion!!

- norm violation
  CTD problem:

  $N_2 = \{(\top, \neg a), (\neg a, \neg b), (a, b)\}$ input: a - output: L for $out_{3,4}$ Explosion!!

- permission as exception

  $N_3 = \{(\top, \neg k)\}$ $P = \{(s, k)\}$ input: s - statpermission: L

Constrained I/O Logic (cIOL)

Problem with uIOL: excess output

- conflict

  $N_1 = \{(a, b), (a, \neg b)\}$ input: a - output: $Cn(b, \neg b) = L$ Explosion!!

- norm violation
  CTD problem:

  $N_2 = \{(\top, \neg a), (\neg a, \neg b), (a, b)\}$ input: a - output: L for $out_{3,4}$ Explosion!!

- permission as exception

  $N_3 = \{(\top, \neg k)\}$ $P = \{(s, k)\}$ input: s - statpermission: L

Threshold idea

- Cut back the set of generators in N to just below the threshold of yielding excess

Norm violation

- C: a set of additional formulae called 'constraints'. The output must be consistent with it.
- For CTDs, $C = A$.
  - Cf. Hansson's setteldness interpretation of circumstances

Maxfamily

- $maxfamily(N, A, C)$ is the set of $\subseteq$-maximal subsets N' of N such that $out(N', A)$ is consistent with C.
- $outfamily(N, A, C) = \{out(N', A) \mid N' \in maxfamily(N, A, C)\}$.
- $out_c(N, A) = out_{\cup/\cap}(N, A) = \cup / \cap \, outfamily(N, A, C)$

Terminology:

- $out_c(N, A)$: constrained output.
- $out_\cap(N, A)$: full meet constrained output (skeptical)
- $out_\cup(N, A)$: full join constrained output (credulous)

Chisholm example

h: help; t: tell

$$N = \{\underbrace{(\top, h)}_{❶}, \underbrace{(h, t)}_{❷}, \underbrace{(\neg h, \neg t)}_{❸}\}$$

$A = \{\neg h\} = C$

$out = out_{3,4}$

Chisholm example

h: help; t: tell

$$N = \{\underbrace{(\top, h)}_{❶}, \underbrace{(h, t)}_{❷}, \underbrace{(\neg h, \neg t)}_{❸}\}$$

$A = \{\neg h\} = C$ $\qquad$ $out = out_{3,4}$

$$maxfamily(N, A, A) = \{\{❷, ❸\}\} \text{ and } out_{\cup/\cap}(N, A) = Cn(\neg t)$$

Chisholm example

h: help; t: tell

$$N = \{\underbrace{(\top, h)}_{❶}, \underbrace{(h, t)}_{❷}, \underbrace{(\neg h, \neg t)}_{❸}\}$$

$A = \{\neg h\} = C$ $\qquad$ $out = out_{3,4}$

$\quad maxfamily(N, A, A) = \{\{❷, ❸\}\}$ *and* $out_{\cup/\cap}(N, A) = Cn(\neg t)$

$A = \{h\}$

$\quad maxfamily(N, A, A) = \{\{❶, ❷, ❸\}\}$ *and* $out_{\cup/\cap}(N, A) = Cn(h, t)$

Multiple level of violations

k: keep promise; a: apologies; s: ashamed

$$N = \{\underbrace{(\top, k)}_{❶}, \underbrace{(\neg k, a)}_{❷}, \underbrace{(\neg k \wedge \neg a, s)}_{❸}\}$$

Multiple level of violations

k: keep promise; a: apologies; s: ashamed

$$N = \{\underbrace{(\top, k)}_{❶}, \underbrace{(\neg k, a)}_{❷}, \underbrace{(\neg k \wedge \neg a, s)}_{❸}\}$$

$A = \{\neg k\}$

$maxfamily(N, A, A) =$
$out_{\cup/\cap}(N, A) =$

$A = \{\neg k, \neg a\}$

$maxfamily(N, A, A) =$
$out_{\cup/\cap}(N, A) =$

Multiple level of violations

k: keep promise; a: apologies; s: ashamed

$$N = \{\underbrace{(\top, k)}_{❶}, \underbrace{(\neg k, a)}_{❷}, \underbrace{(\neg k \wedge \neg a, s)}_{❸}\}$$

$A = \{\neg k\}$

$maxfamily(N, A, A) = \{\{❷, ❸\}\}$
$out_{\cup/\cap}(N, A) = Cn(a)$

$A = \{\neg k, \neg a\}$

$maxfamily(N, A, A) =$
$out_{\cup/\cap}(N, A) =$

Multiple level of violations

k: keep promise; a: apologies; s: ashamed

$$N = \{\underbrace{(\top, k)}_{❶}, \underbrace{(\neg k, a)}_{❷}, \underbrace{(\neg k \wedge \neg a, s)}_{❸}\}$$

$A = \{\neg k\}$

$$maxfamily(N, A, A) = \{\{❷, ❸\}\}$$
$$out_{\cup/\cap}(N, A) = Cn(a)$$

$A = \{\neg k, \neg a\}$

$$maxfamily(N, A, A) = \{\{❸\}\}$$
$$out_{\cup/\cap}(N, A) = Cn(s)$$

Accommodating conflicts

Enough to take $out_{\cap}(N, A)$.

Accommodating conflicts

Unary conflict:

$$\underbrace{(a, b)}_{❶}, \underbrace{(a, \neg b)}_{❷} \qquad\qquad C = \emptyset$$

$A = \{a\}$

$$maxfamily(N, A, C) = \{\{❶\}, \{❷\}\}$$
$$outfamily(N, A, C) = \{\{Cn(b), Cn(\neg b)\}$$
$$out_{\cap}(N, A) = Cn(b \vee \neg b)$$

Accommodating conflicts

Binary conflict:

$$\underbrace{(a, b)}_{❶}, \underbrace{(a, c)}_{❷} \qquad C = \{b \to \neg c\}$$

$A = \{a\}$

$$maxfamily(N, A, C) = \{\{❶\}, \{❷\}\}$$
$$outfamily(N, A, C) = \{\{Cn(b), Cn(c)\}$$
$$out_{\cap}(N, A) = Cn(b \vee c)$$

Accommodating conflicts

What, if no pairs in N generate "trouble"? That is, when

$$out(N, A) \text{ is consistent with } C?$$

Accommodating conflicts

What, if no pairs in N generate "trouble"? That is, when

$$out(N, A) \text{ is consistent with } C?$$

Maxfamily

- $maxfamily(N, A, C)$ is $\{N\}$
- $outfamily(N, A, C) = \{out(N, A)\}$.
- $out_c(N, A) = out(N, A)$

The final output is as in the unconstrained case!

Accommodating conflicts

What, if no pairs in N generate "trouble"? That is, when

$$out(N, A) \text{ is consistent with } C?$$

Remember the lecture on conflicts, and the three requirements any logic should meet.

Desiderata

Desideratum 1 (conflicts are logically consistent)

$\neg \diamond (A \wedge B), \bigcirc A, \bigcirc B \nvdash \bot \qquad (\bot : \text{contradiction})$

Desideratum 2 (Avoid deontic explosion)

The logic should not contain (DEX), or anything like it.

DEX) $\quad \neg \diamond (A \wedge B), \bigcirc A, \bigcirc B \vdash \bigcirc C$

Desideratum 3

The logic should explain in a plausible way the apparent validity of several paradigm arguments:

Desiderata

Desideratum 1 (conflicts are logically consistent)

$\neg\Diamond(A \wedge B), \bigcirc A, \bigcirc B \nvdash \perp$ $\qquad$ ($\perp$: contradiction)

Desideratum 2 (Avoid deontic explosion)

The logic should not contain (DEX), or anything like it.

DEX) $\quad \neg\Diamond(A \wedge B), \bigcirc A, \bigcirc B \vdash \bigcirc C$

Desideratum 3

The logic should explain in a plausible way the apparent validity of several paradigm arguments:

Smith argument (Horty)

i) Smith ought to fight in the army or perform alternative national service. — $\bigcirc(f \vee s)$

ii) Smith ought not to fight in the army. — $\bigcirc\neg f$

$\therefore$ iii) Smith ought to perform alternative national service. — $\bigcirc s$

Desiderata

Desideratum 1 (conflicts are logically consistent)

$\neg\Diamond(A \wedge B), \bigcirc A, \bigcirc B \nvdash \bot$ $\qquad$ ($\bot$: contradiction)

Desideratum 2 (Avoid deontic explosion)

The logic should not contain (DEX), or anything like it.

DEX) $\quad \neg\Diamond(A \wedge B), \bigcirc A, \bigcirc B \vdash \bigcirc C$

Desideratum 3

The logic should explain in a plausible way the apparent validity of several paradigm arguments:

All three are met!

Desiderata

Desideratum 1 (conflicts are logically consistent)

$\neg\Diamond(A \wedge B), \bigcirc A, \bigcirc B \nvdash \bot \qquad (\bot : \text{contradiction})$

Desideratum 2 (Avoid deontic explosion)

The logic should not contain (DEX), or anything like it.

DEX) $\quad \neg\Diamond(A \wedge B), \bigcirc A, \bigcirc B \vdash \bigcirc C$

Desideratum 3

The logic should explain in a plausible way the apparent validity of several paradigm arguments:

$$N = \{\underbrace{(\top, f \vee s)}_{❶}, \underbrace{(\top, \neg f)}_{❷}\} \qquad C = \emptyset$$

For $A = \top$, $out(N, A) = Cn(s)$ is consistent.

Conflict resolution

Basic idea

Start with a priority relation $\geq$ amongs rules in N

- $(a, x) \geq (b, y)$: (a, x) is at least as strong as (b, y)
- $(a, x) > (b, y)$: (a, x) is strictly stronger than (b, y)
  - $\alpha > \beta = \alpha \geq \beta \& \beta \not\geq \alpha$

Lift it to a relation $\geq^s$ amongs sets of rules.
Use $\geq^s$ to select a "preferred" element in the maxfamily.
Restrict the output to this preferred element.

Lifting

$\forall\forall$ definition

$N_1 \geq^s N_2$ iff $\forall (a, x) \in N_1 \; \forall (b, y) \in N_2 \; (a, x) \geq (b, y)$

Lifting

∀∀ definition

$N_1 \geq^s N_2$ iff $\forall (a, x) \in N_1 \ \forall (b, y) \in N_2 \ (a, x) \geq (b, y)$

Brass

$N_1 \geq^s N_2$ iff $\forall (a, x) \in N_2 - N_1 \ \exists (b, y) \in N_1 - N_2 \ (b, y) \geq (a, x)$

Lifting

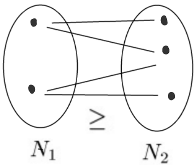
∀∀ definition

$N_1 \geq^s N_2$ iff $\forall (a, x) \in N_1\ \forall (b, y) \in N_2\ (a, x) \geq (b, y)$

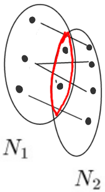
Brass

$N_1 \geq^s N_2$ iff $\forall (a, x) \in N_2 - N_1\ \exists (b, y) \in N_1 - N_2\ (b, y) \geq (a, x)$

N_1 and N_2 do not overlap



N_1 and N_2 overlap



$\boxed{\bullet}$ is put aside

Lifting

$\forall\forall$ definition

$N_1 \geq^s N_2$ iff $\forall (a,x) \in N_1 \; \forall (b,y) \in N_2 \; (a,x) \geq (b,y)$

Brass

$N_1 \geq^s N_2$ iff $\forall (a,x) \in N_2 - N_1 \; \exists (b,y) \in N_1 - N_2 \; (b,y) \geq (a,x)$

Mission exampe (Goble)

$(\top, a) > (\top, b) > (\top, c) \qquad C = \{a \to (b \to \neg c), a \to \neg b\}$

maxfamily

$\underbrace{\{(\top, b), (\top, c)\}}_{N'} \; \underbrace{\{(\top, a), (\top, c)\}}_{N''}$

Brass: $N'' >^s N'$ – witness: $(\top, a) > (\top, b)$
$\forall\forall$: N'' and N' not comparable – witness: $(\top, c)$

Lifting

$\forall\forall$ definition

$N_1 \geq^s N_2$ iff $\forall (a,x) \in N_1 \; \forall (b,y) \in N_2 \; (a,x) \geq (b,y)$

Brass

$N_1 \geq^s N_2$ iff $\forall (a,x) \in N_2 - N_1 \; \exists (b,y) \in N_1 - N_2 \; (b,y) \geq (a,x)$

Mission exampe (Goble)

$\$10 > \$8 > \$6 \qquad \$16 > \$14$

maxfamily

$\underbrace{\{(\top,b),(\top,c)\}}_{N'} \; \underbrace{\{(\top,a),(\top,c)\}}_{N''}$

Brass: $N'' >^s N'$ – witness: $(\top,a) > (\top,b)$

$\forall\forall$: N'' and N' not comparable – witness: $(\top,c)$

Worst + best > Worst + 2nd best. No?

Calculating the output

Full construction

- *maxfamily*(N, A, C) is the set of $\subseteq$-maximal subsets N' of N such that $out(N', A)$ is consistent with C
- *filterfamily*(N, A, C) is the set of $N' \in maxfamily(N, A, C)$ that "maximize" the ouput, i.e., that are such that $out(N', A) \subset out(N'', A)$ for no $N'' \in maxfamily(N, A, C)$
- *preffamily*(N, A, C) is the set of $\geq^s$-maximal elements of *filterfamily*(N, A, C)
- $preffamily_d(N, A, C)$ is the set of elements N' of *preffamily*(N, A, C) stripped of all the pairs (a, x) that are "inactive" in N', in the sense that $out(N', A) = out(N' - \{(a, x)\}, A)$.
  - $=$ the set of 'binding' (good, etc) reasons

“Preferred” output

Light version

- $maxfamily(N, A, C)$ is the set of $\subseteq$-maximal subsets N' of N such that $out(N', A)$ is consistent with C
- $preffamily(N, A, C)$ is the set of $\geq^s$-maximal elements of $maxfamily(N, A, C)$

  go through $N_1, \ldots N_n$
  pick up the best (strongest.,..) one(s) under $\geq^s$

  - $=$ the set of 'binding' (good, etc) reasons
- $out_P(N, A) = out(N', A)$, where $N' \in preffamily(N, A, C)$

out_P: preferred output

Cancer example

❶ is $(a, \neg b)$
❷ is (a, b)
❸ is (b, c)

❸ $>$ ❷ $>$ ❶

a: set of data
b: chemo
c: keep WBC count high enough (drug)

Cancer example

❶ is $(a, \neg b)$
❷ is (a, b)
❸ is (b, c)

❸ $>$ ❷ $>$ ❶

a: set of data
b: chemo
c: keep WBC count high enough (drug)

$A = \{a\} = C$

$out{=}out_3$ or out_4

- Maxfamifly= {{❶ ,❸ },{❷ ,❸ }}
- Preffamily = {{❷ ,❸ }}

So

$$out_P = Cn(b, c)$$

Cancer example

❶ is $(a, \neg b)$
❷ is (a, b)
❸ is (b, c)

❸ $>$ ❷ $>$ ❶

a: set of data
b: chemo
c: keep WBC count high enough (drug)

$A = \{a, \neg c\} = C$ (c out of reach)

- Maxfamifly= {{❶ ,❸ },{❷ }}
- Preffamily = {{❶ ,❸ }}

So

$$out_P = Cn(\neg b)$$

This tallies with our intuitions: usually physicians postpone chemo.
Most approaches from literature output b.

Non-triggerred high-ranking obligations

Horty

$$N = \{❶, ❷, ❸\} \qquad A = \{\top\} \qquad ❸ > ❷ > ❶$$

Priest(❶)	$(\top, a)$	
Bishop(❷)	$(\top, b)$	a: heating on
Cardinal(❸)	$(a, \neg b)$	b: window open

Use out_3 and/or out_4. Require output be consistent, viz $C = \emptyset$.

Non-triggerred high-ranking obligations

Horty

$$N = \{❶, ❷, ❸\} \qquad A = \{\top\} \qquad ❸ > ❷ > ❶$$

Priest(❶)	$(\top, a)$	
Bishop(❷)	$(\top, b)$	a: heating on
Cardinal(❸)	$(a, \neg b)$	b: window open

Use out_3 and/or out_4. Require output be consistent, viz $C = \emptyset$.

maxfamily

$$\underbrace{\{❶, ❷\}}_{N_1} \quad \underbrace{\{❶, ❸\}}_{N_2} \quad \underbrace{\{❷, ❸\}}_{N_3} \qquad N_3 >^s N_2 >^s N_1$$

preffamily

$$\underbrace{\{❷, ❸\}}_{N_3}$$ final output: $Cn(b)$