Preliminaries: Models, theories, consequence relations

Marek Sergot
Department of Computing
Imperial College, London

January 2005 v1.0h

Notation

$\mathcal{L}$ is some (logical) language, usually propositional or (a fragment of) first-order predicate logic. Unless stated otherwise, $\mathcal{L}$ is closed under truth-functional operations. (Thus, if $\alpha \in \mathcal{L}$ then $\neg\alpha \in \mathcal{L}$, and if $\alpha \in \mathcal{L}$ and $\beta \in \mathcal{L}$ then $\alpha \vee \beta \in \mathcal{L}$, $\alpha \wedge \beta \in \mathcal{L}$, $\alpha \rightarrow \beta \in \mathcal{L}$, etc.)
Lower-case Greek letters $\alpha, \beta, \ldots$ range over formulas, lower-case Latin letters, $p, q, r, \ldots$ represent atomic formulas.
Upper-case letters $A, B, C, \ldots, X, Y, \ldots$ represent sets of formulas.
(Sometimes I say 'sentence' instead of 'formula'.)

$\mathcal{M}$ is a model* for $\mathcal{L}$.

$\mathcal{M} \models \alpha$	means that formula α evaluates to true in model $\mathcal{M}$; $\mathcal{M}$ is a model of α.
$\models \alpha$	means that formula α evaluates to true in all models for $\mathcal{L}$.
$A \models \alpha$	means that α is true in all models of A, i.e. $\models (\bigwedge A \rightarrow \alpha)$ when A is finite. ($\bigwedge A$ denotes the conjunction of all formulas of A.)

$\mathcal{M}$ is a model of a set of formulas A when $\mathcal{M} \models \alpha$ for every formula α in A. I will write $\mathcal{M} \models A$. (So: $A \models \alpha$ means $\mathcal{M} \models \alpha$ for every $\mathcal{M}$ such that $\mathcal{M} \models A$.)
For convenience (laziness) I will sometimes write $A \models B$ when B is a set of formulas. $A \models B$ means $A \models \beta$ for all $\beta \in B$, i.e. $A \models \bigwedge B$.

$\mathrm{Th}(A)$ stands for the set of all classical truth-functional consequences of A, i.e.

$$\mathrm{Th}(A) \stackrel{\text{def}}{=} \{\alpha \in \mathcal{L} \mid A \models \alpha\}$$

*Excuse my terminology. By '$\mathcal{M}$ is a model for a $\mathcal{L}$' I mean $\mathcal{M}$ is a structure in which formulas of $\mathcal{L}$ can be evaluated — so it's an interpretation of formulas of $\mathcal{L}$.

Consequence operators

Th is an example of a *consequence operator*. It can be regarded as an operator mapping sets of formulas of $\mathcal{L}$ to sets of formulas of $\mathcal{L}$.
We will be looking at many other kinds of consequence operators presently.
In general, I will write $\mathrm{Cn}(A)$ for the set of all consequences of A. (We will look at various examples of Cn.)
$A \vdash \alpha$ means the same as $\alpha \in \mathrm{Cn}(A)$. $\vdash$ is then a *consequence relation.*

For convenience (laziness): When B is a set of formulas, $A \vdash B$ means $A \vdash \beta$ for all $\beta \in B$, i.e., $B \subseteq \mathrm{Cn}(A)$.
It is convenient to use both notations. For example, transitivity of Cn (a property of some consequence relations but not all) is easier to see when written

- if $A \vdash B$ and $B \vdash C$ then $A \vdash C$

than when written in the form

- if $B \subseteq \mathrm{Cn}(A)$ and $C \subseteq \mathrm{Cn}(B)$ then $C \subseteq \mathrm{Cn}(A)$

which is also equivalent to

- if $B \subseteq \mathrm{Cn}(A)$ then $\mathrm{Cn}(B) \subseteq \mathrm{Cn}(A)$

I will sometimes write $A \vdash_{PL} \alpha$ for $\alpha \in \mathrm{Th}(A)$ by analogy with $\vdash$ and Cn.

Note: many authors use the notation Cn where I use Th, and then something like C for a consequence operator in general. However, many of the papers in AI have employed Th and for that reason I have adopted it here. That leaves Cn as a natural choice for consequence operators generally.

Some properties of Th

It is easy to check that Th has the following properties (among others). I leave the checking as a simple exercise. (The only one that isn't obvious is 'compactness'.) Some of the properties have names. (It is not necessary to memorize them.)

- $A \subseteq \mathrm{Th}(A)$ (inclusion)
  $A \vdash_{PL} A$ (reflexivity)
- $\mathrm{Th}(\mathrm{Th}(A)) = \mathrm{Th}(A)$ (idempotence)
- if $A \subseteq B$ then $\mathrm{Th}(A) \subseteq \mathrm{Th}(B)$ (monotony)
  if $A \vdash_{PL} \alpha$ then $A \cup X \vdash_{PL} \alpha$
- if $B \subseteq \mathrm{Th}(A)$ then $\mathrm{Th}(B) \subseteq \mathrm{Th}(A)$ (transitivity/syllogism)
  if $A \vdash_{PL} B$ and $B \vdash_{PL} C$ then $A \vdash_{PL} C$

- if $A \subseteq B \subseteq \text{Th}(A)$ then $\text{Th}(B) \subseteq \text{Th}(A)$ ('cut')
  if $A \cup B \vdash_{PL} \alpha$ and $A \vdash_{PL} B$ then $A \vdash_{PL} \alpha$ ('cumulative transitivity')
- $\beta \in \text{Th}(A \cup \{\alpha\})$ iff $(\alpha \to \beta) \in \text{Th}(A)$ (deduction)
  $A \cup \{\alpha\} \vdash_{PL} \beta$ iff $A \vdash_{PL} (\alpha \to \beta)$
- if $\alpha \in \text{Th}(A)$ then $\alpha \in \text{Th}(A')$ for some finite $A' \subseteq A$ (compactness)
- $\text{Th}(\emptyset) \neq \emptyset$ (because e.g. $p \vee \neg p \in \text{Th}(\emptyset)$)
  $\text{Th}(\emptyset)$ is the set of all tautologies.
- $\text{Th}(\{p, \neg p\}) = \mathcal{L}$
  $\{p, \neg p\} \vdash_{PL} \alpha$ any α
- $\{\alpha, \beta\} \subseteq \text{Th}(A)$ iff $(\alpha \wedge \beta) \in \text{Th}(A)$
  $A \vdash_{PL} \alpha$ and $A \vdash_{PL} \beta$ iff $A \vdash_{PL} \alpha \wedge \beta$
- if $\{\alpha, \alpha \to \beta\} \subseteq \text{Th}(A)$ then $\beta \in \text{Th}(A)$
- if $\emptyset \vdash_{PL} \alpha \leftrightarrow \beta$ then $A \vdash_{PL} \alpha$ iff $A \vdash_{PL} \beta$
- $\text{Th}(\emptyset) \subseteq \text{Th}(A)$ (every $\text{Th}(A)$ contains all tautologies)
- $\alpha \vee \beta \in \text{Th}(\{\alpha\})$
- 'disjunction in the premises' ('OR')
  if $A \cup \{\alpha\} \vdash_{PL} \gamma$ and $A \cup \{\beta\} \vdash_{PL} \gamma$ then $A \cup \{\alpha \vee \beta\} \vdash_{PL} \gamma$
  $\text{Th}(A \cup \{\alpha\}) \cap \text{Th}(A \cup \{\beta\}) \subseteq \text{Th}(A \cup \{\alpha \vee \beta\})$

(And many others)

Classical consequence operators (Tarski)

A consequence operator Cn is **'classical'** if it satisfies the following three properties.

- $A \subseteq \text{Cn}(A)$ (inclusion)
- $\text{Cn}(\text{Cn}(A)) \subseteq \text{Cn}(A)$ (closure)
- if $A \subseteq B$ then $\text{Cn}(A) \subseteq \text{Cn}(B)$ (monotony)

Note that inclusion and closure together imply

- $\text{Cn}(\text{Cn}(A)) = \text{Cn}(A)$ (idempotence)

Expressed in terms of the corresponding consequence relation $\vdash$ the three defining conditions are:

- if $\alpha \in A$ then $A \vdash \alpha$ (reflexivity)
- if $A \vdash B$ and $A \cup B \vdash \alpha$ then $A \vdash \alpha$ ('cut')
- if $A \subseteq B$ then $A \vdash \alpha$ implies $B \vdash \alpha$ (monotony)
  (Or: if $A \vdash \alpha$ then $A \cup X \vdash \alpha$, any X)

'cut' and 'closure/idempotence'

Expressed in terms of Cn, 'cut' is

- if $B \subseteq \text{Cn}(A)$ then $\text{Cn}(A \cup B) \subseteq \text{Cn}(A)$

or equivalently (assuming $A \subseteq \text{Cn}(A)$)

- if $A \subseteq B \subseteq \text{Cn}(A)$ then $\text{Cn}(B) \subseteq \text{Cn}(A)$

Easy to check that 'closure' is a special case of 'cut':

- $A \subseteq \text{Cn}(A) \subseteq \text{Cn}(A)$ ('inclusion');
- 'cut' gives $\text{Cn}(\text{Cn}(A)) \subseteq \text{Cn}(A)$ ('closure').

(It is in the Tutorial Exercise sheet.)

Side remark: In fact, inclusion with monotony make cut equivalent to closure.
(Show 'closure' implies 'cut': Suppose $A \subseteq B \subseteq \text{Cn}(A)$. Then by monotony, $\text{Cn}(B) \subseteq \text{Cn}(\text{Cn}(A))$. But $\text{Cn}(\text{Cn}(A)) \subseteq \text{Cn}(A)$ ('closure'), so we get $\text{Cn}(B) \subseteq \text{Cn}(A)$ as required.)

In summary: 'cut' is more general than closure, though for monotonic consequence relations (with inclusion/reflexivity) they are equivalent.
When we look at non-classical consequence relations later in the course, 'cut' will be a more useful property than 'closure'.

Examples

- identity $Id(A) = A$ is a classical consequence operator.
- absurdity $Abs(A) = \mathcal{L}$ is a classical consequence operator.

Check these claims. (Easy exercise)

For identity $Id(A) = A$, three things to check (all trivial). For all A:

- 'inclusion': $A \subseteq A$.
- 'closure': $Id(Id(A)) \subseteq Id(A)$ is just $A \subseteq A$.
- 'monotony': if $A \subseteq B$ then $A \subseteq B$ (trivial).

For absurdity $Abs(A) = \mathcal{L}$, three things to check (all trivial). For all A:

- 'inclusion': $A \subseteq \mathcal{L}$.
- 'closure': $Abs(Abs(A)) \subseteq Abs(A)$ is just $\mathcal{L} \subseteq \mathcal{L}$.
- 'monotony': if $A \subseteq B$ then $\mathcal{L} \subseteq \mathcal{L}$ (trivial).

(Thanks to David Tuckey, MSc 2017, for spotting a typo.)

Classical truth-functional consequence Th

Classical truth-functional consequence Th is a classical consequence operator. Check:

- $A \subseteq \text{Th}(A)$
- $\text{Th}(\text{Th}(A)) \subseteq \text{Th}(A)$
- if $A \subseteq B$ then $\text{Th}(A) \subseteq \text{Th}(B)$

In addition, Th has the following properties:

- $\beta \in \text{Th}(A \cup \{\alpha\})$ iff $(\alpha \to \beta) \in \text{Th}(A)$ (deduction)
- if $\alpha \in \text{Th}(A)$ then $\alpha \in \text{Th}(A')$ for some finite $A' \subseteq A$ (compactness)

Some consequence operators Cn have the additional properties of deduction and compactness, some do not.

A consequence relation Cn which includes classical truth-functional consequence

$$\text{Th}(A) \subseteq \text{Cn}(A)$$

is called 'supraclassical'. Most of the consequence relations that come up later are supraclassical, but some are not.

Non-monotonic consequence relations

Many consequence relations used in knowledge representation fail monotony: they are non-monotonic.

$$\{\textit{coffee}\} \vdash \textit{tastes-nice} \qquad \text{BUT} \qquad \{\textit{coffee}, \text{diesel-oil}\} \nvdash \textit{tastes-nice}$$

We will look at a variety of non-monotonic consequence relations later in the course.

It is nevertheless possible to identify some general properties that (non-)monotonic consequence relations can have. We will look at some after we have looked at some specific non-monotonic reasoning systems and logics. If you are curious in the meantime:

Further reading:
David Makinson, General Patterns in Nonmonotonic Reasoning.
Handbook of Logic in Artificial Intelligence and Logic Programming (Vol 3), Gabbay, Hogger, Robinson (eds). Oxford University Press, 1994.