11.

Defaults

Strictness of FOL

To reason from $P(a)$ to $Q(a)$, need either

- facts about a itself
- universals, e.g. $\forall x(P(x) \supset Q(x))$
  - something that applies to all instances
  - all or nothing!

But most of what we learn about the world is in terms of <u>generics</u>

e.g., encyclopedia entries for ferris wheels, violins, turtles, wildflowers

Properties are not strict for all instances, because

- genetic / manufacturing varieties
  - early ferris wheels
- cases in exceptional circumstances
  - dried wildflowers
- borderline cases
  - toy violins
- imagined cases
  - flying turtles

etc.

Generics vs. universals

✓ Violins have four strings.

vs.

× All violins have four strings.

vs.

? All violins that are not E_1 or E_2 or ... have four strings.

(exceptions usually cannot be enumerated)

Similarly, for general properties of individuals

- Alexander the great: ruthlessness
- Ecuador: exports
- pneumonia: treatment

Goal: be able to say a *P* is a *Q* <u>in general</u>, but not necessarily

It is reasonable to conclude *Q(a)* given *P(a),*
unless there is a good reason not to

Here: qualitative version (no numbers)

Varieties of defaults (I)

General statements

- prototypical: The prototypical *P* is a *Q*.

  Owls hunt at night.

- normal: Under typical circumstances, *P*'s are *Q*'s.

  People work close to where they live.

- statistical: Most *P*'s are *Q*'s.

  The people in the waiting room are growing impatient.

Lack of information to the contrary

- group confidence: All known *P*'s are *Q*'s.

  Natural languages are easy for children to learn.

- familiarity: If a *P* was not a *Q,* you would know it.
  - an older brother
  - very unusual individual, situation or event

Varieties of defaults (II)

Conventional

- conversational: Unless I tell you otherwise, a P is a Q

  "There is a gas station two blocks east."
  the default: the gas station is open.

- representational: Unless otherwise indicated, a P is a Q

  the speed limit in a city

Persistence

- inertia: A P is a Q if it used to be a Q.
  - colours of objects
  - locations of parked cars (for a while!)

Here: we will use "Birds fly" as a typical default.

Closed-world assumption

Reiter's observation:

There are usually many more -ve facts than +ve facts!

Example: airline flight guide provides

DirectConnect(cleveland,toronto) DirectConnect(toronto,northBay)
DirectConnect(toronto,winnipeg) ...

but not: ¬DirectConnect(cleveland,northBay)

Conversational default, called CWA:

only +ve facts will be given, relative to some vocabulary

But note: KB $\not\models$ -ve facts (would have to answer: "I don't know")

Proposal: a new version of entailment: KB $\models_c \alpha$ iff KB $\cup$ *Negs* $\models \alpha$

where *Negs* = $\{\neg p \mid p$ atomic and KB $\not\models p\}$

Note: relation to negation as failure

a common pattern:
KB´ = KB $\cup$ Δ

Gives: KB $\models_c$ +ve facts and -ve facts

Properties of CWA

For every α (without quantifiers), KB $\models_c \alpha$ or KB $\models_c \neg\alpha$

Why? Inductive argument:

- immediately true for atomic sentences
- push $\neg$ in, e.g. KB $\models \neg\neg\alpha$ iff KB $\models \alpha$
- KB $\models (\alpha \wedge \beta)$ iff KB $\models \alpha$ and KB $\models \beta$
- Say KB $\not\models_c (\alpha \vee \beta)$. Then KB $\not\models_c \alpha$ and KB $\not\models_c \beta$.
  So by induction, KB $\models_c \neg\alpha$ and KB $\models_c \neg\beta$. Thus, KB $\models_c \neg(\alpha \vee \beta)$.

CWA is an assumption about <u>complete</u> knowledge

never any unknowns, relative to vocabulary

In general, a KB has <u>incomplete</u> knowledge,

e.g. Let KB be $(p \vee q)$. Then KB $\models (p \vee q)$,
but KB $\not\models p$, KB $\not\models \neg p$, KB $\not\models q$, KB $\not\models \neg q$

With CWA, have: If KB $\models_c (\alpha \vee \beta)$, then KB $\models_c \alpha$ or KB $\models_c \beta$.

similar argument to above

Query evaluation

With CWA can reduce queries (without quantifiers) to the atomic case:

KB $\models_c (\alpha \wedge \beta)$ iff KB $\models_c \alpha$ and KB $\models_c \beta$

KB $\models_c (\alpha \vee \beta)$ iff KB $\models_c \alpha$ or KB $\models_c \beta$

KB $\models_c \neg(\alpha \wedge \beta)$ iff KB $\models_c \neg\alpha$ or KB $\models_c \neg\beta$

KB $\models_c \neg(\alpha \vee \beta)$ iff KB $\models_c \neg\alpha$ and KB $\models_c \neg\beta$

KB $\models_c \neg\neg\alpha$ iff KB $\models_c \alpha$

reduces to: KB $\models_c \rho$, where ρ is a literal

If KB $\cup$ *Negs* is consistent, get KB $\models_c \neg\alpha$ iff KB $\not\models_c \alpha$

reduces to: KB $\models_c p$, where p is atomic

If atoms stored as a table, deciding if KB $\models_c \alpha$ is like DB-retrieval:

- reduce query to set of atomic queries
- solve atomic queries by table lookup

Different from ordinary logic reasoning (e.g. no reasoning by cases)

Consistency of CWA

If KB is a set of atoms, then KB $\cup$ *Negs* is always consistent

Also works if KB has conjunctions and if KB has only negative disjunctions

If KB contains $(\neg p \vee \neg q)$. Add both $\neg p$, $\neg q$.

Problem when KB $\models (\alpha \vee \beta)$, but KB $\not\models \alpha$ and KB $\not\models \beta$

e.g. KB = $(p \vee q)$ *Negs* = $\{\neg p, \neg q\}$

KB $\cup$ *Negs* is inconsistent and so for every α, KB $\models_c \alpha$!

Solution: only apply CWA to atoms that are "uncontroversial"

One approach: GCWA

Negs $= \{\neg p \mid$ If KB $\models (p \vee q_1 \vee ... \vee q_n)$ then KB $\models (q_1 \vee ... \vee q_n)$ $\}$

When KB is consistent, get:

- KB $\cup$ *Negs* consistent
- everything derivable is also derivable by CWA

Quantifiers and equality

So far, results do not extend to wffs with quantifiers

can have KB $\not\models_c \forall x.\alpha$ and KB $\not\models_c \neg\forall x.\alpha$

e.g. just because for every t, we have KB $\models_c \neg$DirectConnect(myHome, t) does not mean that KB $\models_c \forall x[\neg$DirectConnect(myHome, x)]

But may want to treat KB as providing complete information about what individuals exist

Define: KB $\models_{cd} \alpha$ iff KB $\cup$ *Negs* $\cup$ *Dc* $\models \alpha$ — where the c_i are all the constants appearing in KB (assumed finite)

where *Dc* is domain closure: $\forall x[x=c_1 \vee ... \vee x=c_n]$,

Get: KB $\models_{cd} \exists x.\alpha$ iff KB $\models_{cd} \alpha[x/c]$, for some c appearing in the KB
KB $\models_{cd} \forall x.\alpha$ iff KB $\models_{cd} \alpha[x/c]$, for all c appearing in the KB

Then add: *Un* is unique names: $(c_i \neq c_j)$, for $i \neq j$

Get: KB $\models_{cdu} (c = d)$ iff c and d are the same constant

→ full recursive query evaluation

Non-monotonicity

Ordinary entailment is monotonic

If KB $\models \alpha$, then KB* $\models \alpha$, for any KB $\subseteq$ KB*

But CWA entailment is *not* monotonic

Can have KB $\models_c \alpha$, KB $\subseteq$ KB', but KB' $\not\models_c \alpha$

e.g. $\{p\} \models_c \neg q$, but $\{p, q\} \not\models_c \neg q$

Suggests study of <u>non-monotonic reasoning</u>

- start with explicit beliefs
- generate implicit beliefs non-monotonically, taking *defaults* into account
- implicit beliefs may not be uniquely determined (vs. monotonic case)

Will consider three approaches:

- minimal entailment: interpretations that minimize abnormality
- default logic: KB as facts + default rules of inference
- autoepistemic logic: facts that refer to what is/is not believed

Minimizing abnormality

CWA makes the extension of all predicates as small as possible

by adding negated literals

Generalize: do this only for selected predicates

Ab predicates used to talk about typical cases

Example KB: Bird(chilly), ¬Flies(chilly),
Bird(tweety), (chilly ≠ tweety),
$\forall x[\text{Bird}(x) \wedge \neg\text{Ab}(x) \supset \text{Flies}(x)]$ ← All birds that are normal fly

Would like to conclude by default Flies(tweety), but KB $\not\models$ Flies(tweety)

because there is an interpretation $\mathfrak{I}$ where I[tweety] $\in$ I [Ab]

Solution: consider only interpretations where I[Ab] is as small as possible, relative to KB

for example: KB requires that I[chilly] $\in$ I[Ab]

this is sometimes called "circumscription" since we circumscribe the Ab predicate

Generalizes to many Ab_i predicates

Minimal entailment

Given two interps over the same domain, $\mathfrak{I}_1$ and $\mathfrak{I}_2$

$\mathfrak{I}_1 \leq \mathfrak{I}_2$ iff $I_1[\mathrm{Ab}] \subseteq I_2[\mathrm{Ab}]$ for every Ab predicate

$\mathfrak{I}_1 < \mathfrak{I}_2$ iff $\mathfrak{I}_1 \leq \mathfrak{I}_2$ but not $\mathfrak{I}_2 \leq \mathfrak{I}_1$ read: $\mathfrak{I}_1$ is more normal than $\mathfrak{I}_2$

Define a new version of entailment, $|=_\leq$ by

KB $|=_\leq \alpha$ iff for every $\mathfrak{I}$, if $\mathfrak{I}$ |= KB and no $\mathfrak{I}^* < \mathfrak{I}$ s.t. $\mathfrak{I}^*$ |= KB then $\mathfrak{I}$ |= α.

So α must be true in all interps satisfying KB that are *minimal* in abnormalities

Get: KB $|=_\leq$ Flies(tweety)

because if interp satisfies KB and is minimal, only *I[chilly]* will be in *I[Ab]*

Minimization need not produce a *unique* interpretation:

Bird(a), Bird(b), $[\neg\mathrm{Flies}(a) \vee \neg\mathrm{Flies}(b)]$ yields two minimal interpretations

KB $|\neq_\leq$ Flies(a), KB $|\neq_\leq$ Flies(b), KB $|=_\leq$ Flies(a) $\vee$ Flies(b)

Different from the CWA: no inconsistency!

But stronger than GCWA: conclude a or b flies

Fixed and variable predicates

Imagine KB as before + $\forall x[\mathrm{Penguin}(x) \supset \mathrm{Bird}(x) \wedge \neg\mathrm{Flies}(x)]$

Get: KB |= $\forall x[\mathrm{Penguin}(x) \supset \mathrm{Ab}(x)]$

So minimizing Ab also minimizes penguins: KB $|=_\leq \forall x \neg\mathrm{Penguin}(x)$

McCarthy's definition: Let **P** and **Q** be sets of predicates

$\mathfrak{I}_1 \leq \mathfrak{I}_2$ iff same domain and

1. $I_1[P] \subseteq I_2[P]$, for every $P \in$ **P** Ab predicates
2. $I_1[Q] = I_2[Q]$, for every $Q \notin$ **Q** fixed predicates

so only predicates in **Q** are allowed to vary

Get definition of $|=_\leq$ that is parameterized by what is minimized *and* what is allowed to vary

Previous example: minimize Ab, but allow only Flies to vary.

Problems:
- need to decide what to allow to vary
- cannot conclude ¬Penguin(tweety) by default!

only get default (¬Penguin(tweety) ⊃ Flies(tweety))

Default logic

Beliefs as deductive theory

explicit beliefs = axioms

implicit beliefs = theorems = least set closed under inference rules

e.g. If we can prove α and $(\alpha \supset \beta)$, then infer β

Would like to generalize to default rules:

If can prove $\mathrm{Bird}(x)$, but *cannot* prove $\neg\mathrm{Flies}(x)$, then infer $\mathrm{Flies}(x)$.

Problem: how to characterize theorems

cannot write a derivation, since do not know when to apply default rules

no guarantee of unique set of theorems

If cannot infer p, infer q + If cannot infer q, infer p ??

Solution: default logic

no notion of theorem

instead, have <u>extensions</u>: sets of sentences that are "reasonable" beliefs, given explicit facts and default rules

Extensions

Default logic KB uses two components: KB = $\langle F, D\rangle$

- F is a set of sentences (facts)
- D is a set of <u>default rules</u>: triples $\langle \alpha : \beta / \gamma \rangle$ read as

If you can infer α, and β is *consistent*, then infer γ

α: the prerequisite, β: the justification, γ: the conclusion

e.g. $\langle \mathrm{Bird(tweety)} : \mathrm{Flies(tweety)} \ / \ \mathrm{Flies(tweety)} \rangle$

treat $\langle \mathrm{Bird}(x) : \mathrm{Flies}(x) / \mathrm{Flies}(x) \rangle$ as <u>set</u> of rules

Default rules where $\beta = \gamma$ are called <u>normal</u> and write as $\langle \alpha \Rightarrow \beta \rangle$

will see later a reason for wanting non-normal ones

A set of sentences E is an <u>extension</u> of $\langle F, D\rangle$ iff for every sentence π, E satisfies the following:

$$\pi \in E \text{ iff } F \cup \Delta \models \pi, \quad \text{where } \Delta = \{\gamma \mid \langle \alpha : \beta / \gamma \rangle \in D, \ \alpha \in E, \neg\beta \notin E\}$$

So, an extension E is the set of entailments of $F \cup \{\gamma\}$, where the γ are assumptions from D.

to check if E is an extension, guess at Δ and show that it satisfies the above constraint

Example

Suppose KB has

F = Bird(chilly), ¬Flies(chilly), Bird(tweety)
D = ‹Bird(x) ⇒ Flies(x)›

then there is a unique extension, where Δ = Flies(tweety)

- This is an extension since tweety is the only t for this Δ such that Bird(t) ∈ E and ¬Flies(t) ∉ E.
- No other extension, since this applies no matter what Flies(t) assumptions are in Δ.

But in general can have multiple extensions:

F = {Republican(dick), Quaker(dick)} D = { ‹Republican(x) ⇒ ¬Pacifist(x)›,
‹Quaker(x) ⇒ Pacifist(x)› }

Two extensions: E_1 has Δ = ¬Pacifist(dick); E_2 has Δ = Pacifist(dick)

Which to believe?

<u>credulous</u>: choose an extension arbitrarily
<u>skeptical</u>: believe what is common to all extensions

Can sometimes use non-normal defaults to avoid conflicts in defaults

‹ Quaker(x) : Pacifist(x) ∧¬Republican(x) / Pacifist(x) ›

but then need to consider all possible interactions in defaults!

Unsupported conclusions

Extension tries to eliminate facts that do not result from either F or D.

e.g., we do not want Yellow(tweety) and its entailments in the extension

But the definition has a problem:

Suppose F = {} and D = ‹ p : True / p ›.

Then E = entailments of {p} is an extension

since $p \in E$ and ¬True ∉ E, for above default

However, no good reason to believe p!

Only support for p is default rule, which requires p itself as a prerequisite
So default should have no effect. Want one extension: E = entailments of {}

Reiter's definition:

For any set S, let Γ(S) be the least set containing F, closed under entailment, and satisfying

if ‹α : β / γ› ∈ D, α ∈ Γ(S), and ¬β ∉ S, then γ ∈ Γ(S).

note: not Γ(S)

A set E is an <u>extension</u> of ‹F, D› iff E = Γ(E).

called a <u>fixed point</u> of the Γ operator

Autoepistemic logic

One disadvantage of default logic is that rules cannot be combined or reasoned about

$\langle\alpha : \beta / \gamma\rangle \twoheadrightarrow \langle\alpha : \beta / (\gamma \vee \delta)\rangle$

Solution: express defaults as *sentences* in an extended language that talks about <u>belief</u> explicitly

for any sentence α, we have another sentence **B**α

Bα says "I believe α": autoepistemic logic

e.g. $\forall x[\text{Bird}(x) \wedge \neg\mathbf{B}\neg\text{Flies}(x) \supset \text{Flies}(x)]$

All birds fly except those that I believe to not fly =

Any bird not believed to be flightless flies.

No longer expressing defaults using formulas of FOL.

Semantics of belief

These are not sentences of FOL, so what semantics and entailment?

- modal logic of belief provide semantics
- for here: treat **B**α as if it were an new atomic wff
- still get entailment: $\forall x[\text{Bird}(x) \wedge \neg\mathbf{B}\neg\text{Flies}(x) \supset \text{Flies}(x) \vee \text{Run}(x)]$

Main property for set of implicit beliefs, E:

1. If $E \models \alpha$ then $\alpha \in E$. (closed under entailment)
2. If $\alpha \in E$ then $\mathbf{B}\alpha \in E$. (positive introspection)
3. If $\alpha \notin E$ then $\neg\mathbf{B}\alpha \in E$. (negative introspection)

Any such set of sentences is called <u>stable</u>

Note: if E contains p but does not contain q, it will contain $\mathbf{B}p$, $\mathbf{BB}p$, $\mathbf{BBB}p$, $\neg\mathbf{B}q$, $\mathbf{B}\neg\mathbf{B}q$, $\mathbf{B}(\mathbf{B}p \wedge \neg\mathbf{B}q)$, *etc*.

Stable expansions

Given KB, possibly containing **B** operators, our implicit beliefs should be a stable set that is minimal.

Moore's definition: A set of sentences E is called a stable expansion of KB iff it satisfies the following:

$\pi \in E$ iff KB $\cup\, \Delta \models \pi$, where $\Delta = \{\mathbf{B}\alpha \mid \alpha \in E\} \cup \{\neg\mathbf{B}\alpha \mid \alpha \notin E\}$

fixed point of another operator

analogous to the extensions of default logic

Example: for KB = { Bird(chilly), ¬Flies(chilly), Bird(tweety), $\forall x[\text{Bird}(x) \wedge \neg\mathbf{B}\neg\text{Flies}(x) \supset \text{Flies}(x)]$ }

get a unique stable expansion containing Flies(tweety)

As in default logic, stable expansions are not uniquely determined

KB = $\{(\neg\mathbf{B}p \supset q), (\neg\mathbf{B}q \supset p)\}$

2 stable expansions
(one with p, one with q)

KB = $\{(\neg\mathbf{B}p \supset p)\}$ (self-defeating default)

no stable expansions!
so what to believe?

Enumerating stable expansions

Define: A wff is objective if it has no **B** operators

When a KB is propositional, and **B** operators only dominate objective wffs, we can enumerate all stable expansions using the following:

1. Suppose $\mathbf{B}\alpha_1, \mathbf{B}\alpha_2, \ldots \mathbf{B}\alpha_n$ are all the **B** wffs in KB.
2. Replace some of these by True and the rest by ¬True in KB and simplify. Call the result KB° (it's objective).

   at most 2^n possible replacements
3. Check that for each α_i,
   - if $\mathbf{B}\alpha_i$ was replaced by True, then KB° $\models \alpha_i$
   - if $\mathbf{B}\alpha_i$ was replaced by ¬True, then KB° $\not\models \alpha_i$
4. If yes, then KB° determines a stable expansion.

   entailments of KB° are the objective part

Example enumeration

For KB = { Bird(chilly), ¬Flies(chilly), Bird(tweety),
[Bird(tweety) ∧ ¬**B**¬Flies(tweety) ⊃ Flies(tweety)],
[Bird(chilly) ∧ ¬**B**¬Flies(chilly) ⊃ Flies(chilly)] }

Two **B** wffs: **B**¬Flies(tweety) and **B**¬Flies(chilly),
so four replacements to try.

Only one satisfies the required constraint:
B¬Flies(tweety) → ¬True,
B¬Flies(chilly) → True

Resulting KB° has
(Bird(tweety) ⊃ Flies(tweety))
and so entails
Flies(tweety)
as desired.

More ungroundedness

Definition of stable expansion may not be strong enough

KB = {(**B**p ⊃ p)} has 2 stable expansions:

- one without p and with ¬**B**p
  corresponds to KB° = {}
- one with p and **B**p.
  corresponds to KB° = {p}

But why should p be believed?

only justification for having p is having **B**p!

similar to problem with default logic extension

Konolige's definition:

A <u>grounded</u> stable expansion is a stable expansion that is minimal wrt to the set of sentences without **B** operators.

rules out second stable expansion

Other examples suggest that an even stronger definition is required!

can get an equivalence with Reiter's definition of extension in default logic